Blockchain-Based Administration of Access in Smart Home IoT

Mehrnoosh Shakarami, James Benson, Ravi Sandhu
mehrnoosh.shakarami@my.utsa.edu,james.benson@utsa.edu,ravi.sandhu@utsa.edu
Institute for Cyber Security (ICS) & NSF Center for Security and Privacy Enhanced Cloud Computing (C-SPECC)
Department of Computer Science, University of Texas at San Antonio

Abstract

There is a rising concern about authorization in IoT environments to be appropriately designed and applied, due to smart things surge to be part of people's daily lives on one hand, and the amount of personal/private information they utilize, on the other hand. Different access control systems have been proposed for different IoT environments, many are remaining only at a conceptual level. In this paper, we propose a decentralized, ledger-based, publish-subscribe based architecture for the administration of access in a smart home IoT environment to preside at the assignments of underlying operational authorizations. Proposed architecture is endorsed by a proof-of-concept implementation, which utilizes smart contracts to ensure the integrity of administration supplemented by intrinsic benefits of blockchain to be distributed and transparent. Despite the rising hype around the blockchain technology that stokes its utilization in different domains, utilizing it for access control purposes is not yet promising. Our implementation results assure using blockchain for administrative access control is propitious, while is not yet appropriate for operational access control, which have been mainly the focus of previously proposed blockchain-based access control works.

CCS Concepts

• **Security and privacy** → **Access control**; **Authorization**;

Keywords

smart home IoT, decentralized access control, blockchain-based access control, access administration

ACM Reference Format:
Mehrnoosh Shakarami, James Benson, Ravi Sandhu. 2022. Blockchain-Based Administration of Access in Smart Home IoT. In *Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems (SaT-CPS '22), April 27, 2022, Baltimore, MD, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3510547.3517921

1 Introduction

The inevitable advent of integration of IoT with people's everyday lives, calls for potent security mechanisms specifically crafted for smart things and each environment in which they are utilized. Authorization issues have been widely explored in the smart home environment, reflecting on its significant role in personal lives. Smart home access control is required to be granular, context-aware, easy to use (as we do not expect smart home users to be IT experts) and being able to manage dynamicity and complexity of the smart home environment. As one of the most important areas of IoT application, smart home has a unique combination of challenges to be dealt with during administration of access [38].

Intrinsic benefits of Role-Based Access Control (RBAC) such as its policy neutrality, adherence to least privilege principle and its built-in support for Static and Dynamic Separation of Duty (SSoD and DSoD), made it a preferred choice in prior research works in order to establish an access control model for mediating access to users in a smart home, using the concept of a role [17, 19, 58, 60]. Besides, when the operational model is established, it is also possible to use RBAC for administration of RBAC. Different users could be assigned to administrative roles to handle administrative tasks in the smart home such as establishing new roles and managing assignments in the underlying operational model.

Although there are many access control models proposed for IoT environments, many of the proposals have remained at the conceptual level. Even so, different deployment mechanisms in some frameworks relying on existing technologies, including cloud [20, 31], Open Authorization (OAuth) [56] and blockchain [40, 44, 46, 47, 49], among them blockchain has been widely used in recognition of its transparency which benefits auditability. Moreover, blockchain's distributed nature removes the need to trust the third parties, which is of advantage to privacy protection. Nonetheless, using blockchain for access control is still controversial [36]. The performance of blockchain-based systems is still not competitive with current centralized access control systems. In time-sensitive applications using blockchain for access control would negatively affect users' experience [42, 53]. On the other hand, communication with blockchain demands higher amounts of computation power and space that is available to resource/energy constrained IoT devices.

In this paper, we propose using blockchain for administration of access, unlike most of previously presented approaches which have been applied in operational environments. We recognize blockchain could bring its intrinsic advantages of distribution, transparency, and scalability to the administration of access while it is not yet practical to be used for operational access control. Using blockchain for enforcing the administrative model would equip the access management with improved security and posteriori auditing [55] as discussed in Section 2, as well as the potential of generalization of the proposed approach to environments with similar dynamics relying on scalable nature of the blockchain. Since administrative access control tasks are less frequent, blockchain's monetary costs and time burdens are bearable, for instance it is reasonable to wait for seconds for an administrative change to take effect.

SaT-CPS '22, April 27, 2022, Baltimore, MD, USA.

ACM ISBN 978-1-4503-9229-7/22/04…$15.00
https://doi.org/10.1145/3510547.3517921

We discuss our enforcement architecture based on blockchain for access administration in the smart home IoT, while Greengrass [2] has been utilized to mediate device-cloud connections.It also handles required access control tasks in the local environment. Greengrass in the corresponding operational model serves as the smart hub and policy engine. So, the blockchain burdens of time, computational power and storage would not be imposed at the operational level. There is no need to store the ledger information or even communication wallets on resource-constrained IoT devices, as we do not use blockchain at the operational level for access control.

To build our enforcement architecture, we adopt the administrative model presented in [57] which relies EGRBAC (Extended Generalized RBAC) [19] as the underlying operational model. Our proposed architecture provides interoperability of administrative and operational levels of access. Besides proposing an enforcement architecture, different interactions to it are presented in a sequence diagram and it is also backed up by a proof-of-concept implementation. The rest of the paper proceeds as follows. Section 2 provides motivation and articulates the problem. Section 3 discusses utilization of blockchain in access control and justifies our proposal of using it at an administrative level. Adopted administrative policy and operational model have been discussed in Section 4. The enforcement architecture alongside a sequence diagram based on a provided use case, are discussed in Section 5. Section 6 describes the experimental setup and results. Properties of the proposed approach as well as important security considerations are discussed in Section 7. Section 8 concludes the paper.

2 Problem Statement and Motivation

Security of any management system is of utmost importance, so would be administration of access in a smart home. The proposed architecture in this paper is intended for administration of access for user to device interactions. It relies on blockchain's intrinsic characteristics of being immutable, tamper resistant and transparent for security provision.

Threat Model. In our proposed architecture IoT devices are not part of the blockchain network for obvious performance benefits elaborated in the next section (See 3.2). Thus, IoT devices would rely on access authorization rules made by access administrators. A malicious insider or an attacker could target the smart home's security by spoofing (impersonating as access manager), tampering (modifying the access control policy towards his/her desired intent), privilege escalation (trying to elevate the available privileges or repudiation (denial of performing an action).

Motivating Example. An example of an insider security threat could be a dishonest babysitter trying to tamper with access control rules so that s/he would have access to the house when s/he is not meant to. As another example an attacker can fake an administrator account enforcing an IoT device to maliciously deny the access to a subject even though the policy would have granted it.

How Blockchain Helps with Security. In our proposed architecture administrator account and access management tasks cannot be forged or manipulated, as we utilize a wallet private key to encrypt an administrator account and definition/configuration of access, so the administrator account cannot be faked. The access administration policy is defined by programming a smart contract and is recorded into the ledger via a consensus process which is protected from tampering relying on the irreversible nature of blockchain. Therefore, it is impossible to change the authorization rules in favor of a malicious insider or an attacker, as stated in the above example. Moreover, access administrative requests are submitted via transactions which helps to verify proper implementation and integrity of access control rules. In our system, the operational access control policy is updated accordingly with transaction logs of the blockchain that ensures authorization rules' authenticity.

Furthermore, a blockchain based solution equips the system with transparency and auditability. So, if any unduly granting/denial of access happens, intrinsically immutable logs of transactions on blockchain provides a way for posteriori auditing and verifying the related policy on the chain. In case of maliciously denying access, using blockchain would equip the system with means to verify which policy was enforced, and if the policy is disobeyed by an IoT device, it reveals the device being maliciously controlled.

However, we are assuming users' communications with the edge services are secured over the home local network. Routing attacks which could stop Greengrass from receiving updates from the cloud, and attacks against web3 API compromising credentials are considered out-of-scope of this paper. Security considerations are discussed in further details in Section 7.2.

3 Blockchain For Access Control

In this section we discuss some of the previous works on utilization of blockchain in operational access control as well as its usage for administration of access.

3.1 Blockchain for Operational Access Control

Numerous access control frameworks have been proposed based on blockchain, given the intense hype around this technology. Ethereum [8] is the first blockchain platform to present the smart contracts [22] and provides a built-in Turing-complete programming language, i.e. Solidity, which makes it possible to create arbitrary state-transition functions on blockchain to encode intended logic. Other blockchain platforms like Hyperledger Fabric [12], Ripple [15], bitcoin [3] and EOS [6] have later provided smart contract capability to their chains. However, being the first platform to provide smart contracts along with the maturest code-base and user-base, Ethereum has been used in many access control frameworks to provide distributed IoT access control.

BlendCAC [62] encodes access rights in capability tokens which are deployed as smart contracts along with another type of smart contract intended for delegation. The proposed capability-based model is at operational level in which each transaction required to spend almost $1.02 to be completed given the gas price in the public Ethereum in 2018, which is strongly prohibitive to be used by a normal user of a smart home even if the Ether price did not spike. Authors of [43] proposed to fix some of the BlendCAC issues, using a fine-grained access control model, however no cost or performance metric was discussed. Another blockchain-based, capability-based approach could be found in [34] which is a decentralized user-centric approach based on the publish-subscribe model.

Figure 1: Whether a Blockchain is the Appropriate Technical Solution for Your Problem [61]

Although some researches mentioned RBAC and ABAC as not flexible or scalable enough to handle access control requirements in IoT environments [33, 50], there are still many research works based on these approaches. An attribute-based access control for IoT environments has been proposed in [32]. Authors tested the proposed frameworks not on a local network, but on one of Ethereum test networks called Rinkeby. Rinkeby uses Proof of Authority (PoA) as its consensus mechanism, which is faster than current Ethereum consensus mechanism, i.e., Proof of Work (PoW). So, presented results in [32] could be considered as a lower bound. Yet, those costs are still prohibitive to be applied in a smart home for operational access control. Another approach [42] has been codified and tested on the Ropsten (PoW) testnet, which uses the same consensus algorithm as Ethereum mainnet (main network). However, the estimated space and gas requirements in the paper proves the proposed approach to be nonviable in a smart home environment.

Role-Based access control has been also used to design blockchain-based approaches for mediating access in IoT environments [24, 51]. Cruz et. al. [24] proposed an RBAC-based platform along with a challenge-response protocol to facilitate inter-organizational access control. Since their RBAC-based smart contract only encodes add/remove a user/endorser and change the contract's status, which we argue as administrative-type of tasks, the evaluation results show this platform to be practical in terms of cost, however authors have not provided any time evaluation.

There are some blockchain-based access control frameworks lacking the basis of a formally defined access control models, such as the trust-based layered framework proposed by Dorri. et al. [30] or the RDF-based architecture for IoT access control in smart buildings [21]. ControlChain [49] is another blockchain-based architecture for IoT authorization which does not rely on a specific access control paradigm, instead authors included an encoder using which different access control models could be transformed to their architecture authorizations. Furthermore, some blockchain-based access control frameworks are built upon other blockchain platforms, such as bitcoin [41, 46], Hyperledger Fabric [37, 40] and EOS [51].

3.2 Why Not to Use Blockchain at Operational Level

As one of the application domains of blockchain, access control in the IoT domain gained a lot of attention in the literature. There are a handful of publications which recognize blockchain-based access control would strengthen overall IoT security [29, 39, 52], while others assert blockchains are not yet ready for mass usage in any domain, for which their designs and code bases have to be more mature [28]. Many of the previously proposed researches use blockchain at the operational level for access control, as briefly discussed in Section 3.1. Nonetheless, there are some inherent characteristics of blockchain which make it unsuitable for that purpose. Elaborating on different approaches to use blockchain for operational access control, consider the following options: In the case of device democracy, which has been advocated by IBM as the future of IoT [1], each IoT device takes responsibility for its own access control. However, not every IoT device could be burdened with required storage and computational power, as many IoT devices are currently energy- and resource-constrained (e.g., light sensors or wearable IoT devices).

Another approaches for blockchain-based access control, use the blockchain as the storage for access control policies [45, 46]. So, every time a policy appended to the set of access control policies or retrieved from the blockchain a transaction should be communicated and confirmed. The required duration of confirming a transaction is inappropriate for operational access control purposes in which a user cannot wait ten minutes for a transaction to be completed [46]. Moreover, some actions might be latency-sensitive, for example when a wearable health IoT device should make an emergency call to 911. As one of the most popular blockchains utilized in access control, Ethereum has the average block time (the time it takes for a block to be added to the blockchain) of 13 seconds [9], which is still significant for a home user to get access to the door lock, for example. In recent research [59], authors implemented their operational access control approach based on an alliance chain built on Ethereum, yet the access control time is in the order of seconds and varies based on number of access requests.

Another problem of using blockchain for operational access control is financial, as every transaction needs a fee to be paid in cryptocurrency to be completed. Considering how recurrent the access transactions would be even in a small IoT environment like a smart home, the monetary burden could be prohibitive. The fluctuating price of cryptocurrency aggravates this problem.

3.3 Blockchain for Administration of Access

Authors in [61] presented a flowchart which shows their standing about the necessity of using blockchain for different use cases. We followed the proposed chart to justify using blockchain in this paper for administration of access in the smart home IoT environment which has been depicted in Figure 1, and indicates our position with utilization of blockchain for administration of access.

Blockchain's features of distributed nature, scalability and transparency make it an appealing infrastructure for access control implementation. Moreover, it could equip the system with essential security benefits, which otherwise cannot be provided using common centralized approaches as explained in Section 2. In this paper we suggest utilizing blockchain for *administrative* access control, not at the operational level, for following reasons:

- Administrative access control tasks are infrequent compared to required operational access authorizations [25], so the burden of required processing time for blockchain adoption is few and far between and worth its benefits.
- As blockchain is an immutable ledger, it provides accountability for administrative tasks. So, access control would be coupled with auditing as a posteriori analysis [55], providing a more complete security solution.
- As the adopted administrative model in this paper is decentralized in nature, it could take benefits of blockchain decentralization to be scaled. So, proposed enforcement architecture could be extended to environments with similar dynamics, e.g., smart buildings. Moreover, relying on the distributed nature of blockchain, we can get around privacy concerns which arise when using third parties in other infrastructure, e.g., cloud.
- The need for storing blockchain information or being involved in heavy computations would be eliminated for resource constrained IoT devices, as those would not be engaged in administration of access.

A distinct feature of our research is to follow the PEI model (Policy, Enforcement Architecture and Implementation) as our reference model [54], which would be further described in Section 5.1. Briefly saying, we rely on an RBAC as the policy model (P in PEI) designed for administration of smart home environments [57]. Almost all the previous works, nevertheless, lack the support of a formal model and rely on informally assumed policy objectives to build their access control frameworks. We then propose our enforcement architecture (E in PEI), implemented (I in PEI) on the Ropsten testnet of Ethereum. Our research is one of the very few works [44] in which administration of access has been considered.

4 PEI: Underlying Administrative Policy

Before discussing our blockchain-based architecture for administration, we briefly describe the RBAC administrative model proposed in [57] which is built upon EGRBAC [19] as underlying operational access control model. There have been multiple studies conducted recently to understand the needs and preferences of smart home users. These studies reported smart home users expressed the need for a fine-grained access control system, and RBAC was reportedly the most preferred approach by users for limiting the access to smart home resources [35, 63, 64]. Adopting EGRBAC as the operational model provides a fine-grained RBAC access control which provides on permission level, instead of device level access provision; so, it would be possible to grant partial access to a device by defining the device role (DR) instead of the whole device control. For instance, a babysitter can turn on/off the AC but is not permitted to change its schedule. EGRBAC captures the environmental context by defining the environment roles (ER) which later would be paired by standard user roles to create the role pairs (RP). RP and DR would later be coupled together to establish the access authorization rules.

EGRBAC is chosen not as a de-facto operational model, but because it has the desired properties for a smart home IoT operational access control on one hand and its enforcement architecture relies on AWS Greengrass [2] which can be best integrated with our enforcement of corresponding administrative model [57]. However, the proposed administrative model in [57] and hence the proposed enforcement architecture in this paper could be utilized for any underlying operational model, regardless of if it being RBAC or used any other access control paradigm.

4.1 Administrative Model

Access administration in a smart home environment is a particular problem as home users lack the expertise of a typical system administrator and are unlikely to spend much time learning complex interfaces to assign/revoke access rights or auditing the access logs. The other complication stems in multiple ownership for smart devices in the home which demands for decentralized access management. Moreover, to avoid a single point of failure it is required to have multiple administrators in the house. For example, if one of the home administrators is on a business trip and there is a problem to the house power system, there should be another administrator who can grant access to the electrician to fix the issues [38].

We adopt a role-based administrative model [57] which corresponds to EGRBAC operational model and governs the authorization functionalities in a smart home in a decentralized way. The decentralization is provided through defining the administrative units (AU), each of which is controlled by an administrative role (AR) which could be taken by multiple administrator users. Each administrative unit controls a predefined set of administrative tasks (AT) which represents the scope of administration. Adopted administrative model classifies possible changes in a smart home IoT environment to be add/remove a user, add/remove a device and modifying the current operational assignments, among which adding a new user is done infrequently and could be done in a centralized way; so, is out of the scope. Therefore, the administrative tasks have been defined as management of the assignment relations in the underlying operational model. Although the model is defined in the smart home context, it could be applied to environments with similar dynamics by defining extra administrative units.

5 PEI: Enforcement Architecture

5.1 Blockchain-Based Enforcement Architecture

In this paper, we consider the access control framework to be based on a three-layer PEI as coined in [54]. PEI stands for Policy (Policy Models), Enforcement and Implementation. Policy layer is specified based on any access control paradigm in an ideal context which assumes all relevant information for making access decisions are instantly and securely available. The Enforcement layer manifests the policy model and provides an enforcement architecture which approximates a correspondent of the policy. Implementation layer deals with detailed implementation technologies and mechanisms.

In this paper, the policy model is adopted from [57] which is a RBAC administrative model. The enforcement architecture in this paper is compatible with the Access Control Oriented (ACO) architecture for cloud-enabled AWS IoT [18, 20], which is enclosed in the gray square with dotted border in Figure 2. Our authorization solution is deployed utilizing the AWS Greengrass SDK [2], an edge run-time and cloud service which provides local messaging,

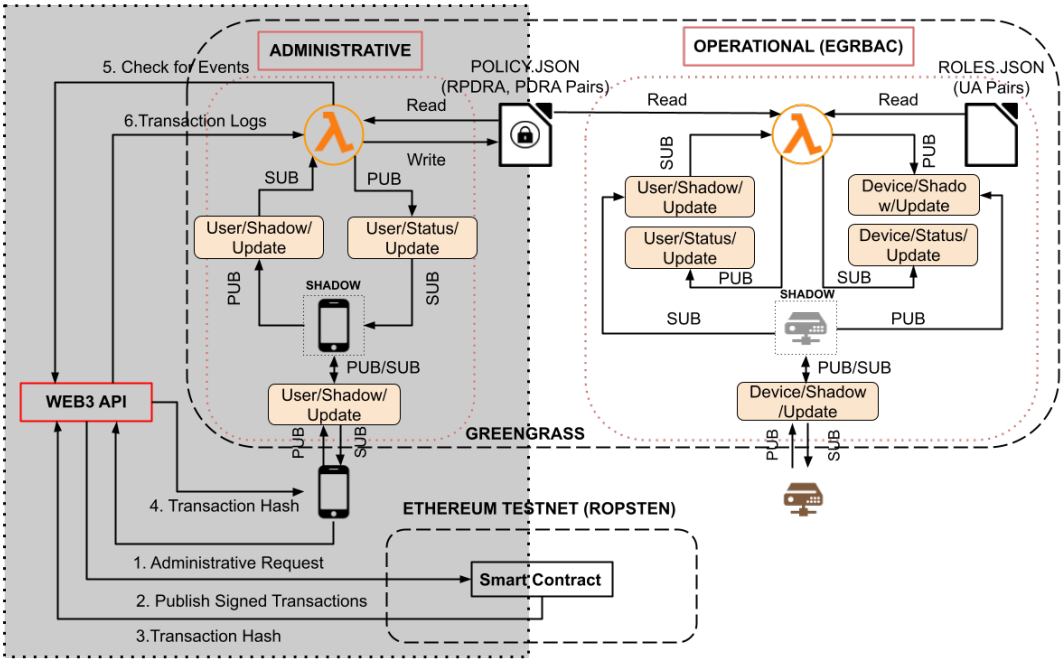

Figure 2: Blockchain-Based Enforcement Architecture for Administration of Access in IoT Smart Home Environment

processing and data management services. We designed our administrative enforcement so as to be interoperable with its underlying operational model [19] which has been shown at the right side of Figure 2. As depicted, the POLICY.JSON file is shared between administrative and operational models; so we chose to protect its integrity with locks against possible concurrent accesses.

Overall, we propose a system which leverages the tight coupling between our authorization design and a publish/subscribe syndication which is specifically useful in the context of smart home IoT environment. Followings are the main components of the Greengrass part of the architecture which runs locally and serves as the smart hub and policy engine in our access management framework:

- Virtual objects (shadows) serve as intermediaries between applications and physical devices and keep the latest known state of the corresponding device. So, the device's state would be available to applications and services even if the device itself is not connected to AWS.
- AWS utilizes a policy-based authorization mechanism. The policies are contained in a JSON file, which includes access control rules for utilizing a resource.
- Lambda functions (λ) are event-driven computational units, sitting and waiting for messages from the topics to which they have been subscribed. As a message is received, the lambda function wakes, does the computation reaching out to POLICY.JSON file, and publishes the results to subscribed topics. We have adopted the *operational* lambda from [19] which executes the operational level policies for user-to-device access requests. We define an *administrative* lambda, which updates the POLICY.JSON file. This file is spelled out based on administrative requests submitted by administrator users, which then would be evaluated based on the administrative policy encoded into smart contracts on the blockchain. Since the policy file is shared between both operational and administrative lambda functions, its integrity should be protected during concurrent accesses by using locks or other concurrency control mechanisms.
- Communications are done through MQTT protocol, which is a lightweight machine-to-machine publish/subscribe messaging protocol, designed for constrained devices. Local MQTT publish/subscribe messaging defines the subscriptions between publishers and subscribers.

As depicted on the left side of the Figure 2, we utilize Ethereum blockchain. Ethereum is a decentralized, public, permissonless, and the most actively used blockchain based on Bloomberg [10]. It is the maturest blockchain in terms of code base, user base and developer community. Ethereum is capable of being configured as both a permission-less and a permissioned blockchain network, as well as the community-based development of the platform. In other words, saying Ethereum is a permissionless blockchain means there is no authority on a network level. The logic deployed on the chain, in the form of a smart contract, does define permissions. In a smart contract, we can define an action that may only be performed by the contract's owner and not by the others.

The whole architecture represented in Figure 2 depicts a scenario of an administrator user at home, in which user can communicate with the Greengrass using the local home network on his/her smartphone. If the user is out of home, however, the administrator user's smartphone would have to communicate with its shadow on the AWS cloud and update it by sending a HTTP request to the AWS IoT Core. Then, the cloud forwards user's request messages to

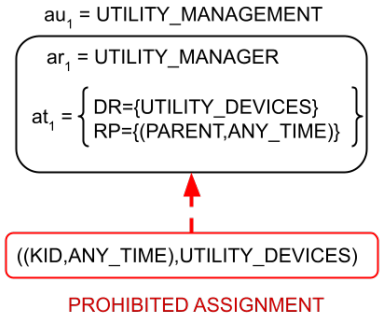

Figure 3: Reference Example

the local Greengrass by publishing to the user's private topic of User/Shadow/Update, which afterwards followed by the same steps as illustrated in Figure 2. At the end, the user's phone shadow on the AWS cloud would update the user's phone with the status of the submitted request.

5.2 Sequence Diagram

For an administrative access request to be handled, a workflow as depicted in the reference scenario in Figure 4 is followed. An administrative user who wants to define/change the assignments of the operational model first submits the request through his/her smartphone. Consider the following example scenario depicted in Figure 3 in which Bob is the parent and the administrator of the smart home. There is an administrative unit (AU) called utility_management with the utility_manager as its administrative role (AR) which has been assigned to Bob. So, Bob is the administrator user who can decide about accesses of different Role Pair (RP) and Device Roles (DR) which are included in the corresponding administrative task (AT). The Device Role, utility_devices, includes the permissions of TurnOn, TurnOff, Reset, Schedule for devices AC,FuseBox,WaterMeter. If Bob, as the utility_administrator, wants to grant the available permissions to a technician for a period of time, he should define the assignment the role pair (RP) (technician,repair_time) to the utility_management device role. It is noteworthy that assigning kids at any time to the utility_devices has been defined as prohibited (refer to Figure 3).

For the above-mentioned example to go through, Bob must determine the desired RP and DR, sign the administrative assignment with his private key (which is securely stored on his personal smartphone) and then submit this administrative request as a transaction to the blockchain. Communication with the blockchain is conducted through a web3 API via HTTP requests. We used Infura [13] as our Web3 API in this paper. After publishing the transaction to the blockchain and getting back the transaction hash, this hash would be returned to the user's phone and also immediately published to the User/Shadow/Update. As λ function has been subscribed to the same topic, it would also be notified with the transaction hash, which would be later used to retrieve the transaction log after being mined by λ. Administrative λ would investigate the transaction log in order to find out if the request has been approved. In either scenario of approve/deny (correspondingly permit/deny), the administrative λ would publish the results to the User/Status/Update topic, so the user would know the results. If permitted, the new assignment would be written to the access control logic, which is the policy.json file. When the technician wants to access the utility_devices, the operational λ would check the access control logic file and grant the required permissions, if defined so.

6 PEI: Implementation

Most of the proposed blockchain-based access control frameworks, which we briefly discussed in Section 3, have left their proposals at the conceptual level [34, 49] or evaluate them on either a locally built blockchain [43, 62, 65] or a testnet which is not using the same protocol as the Ethereum mainnet [32], so none of which provides a reliable and pragmatic assessment of practicality. In this paper, we validated our proposal of using blockchain for administrative access control enforcement by a proof-concept implementation. Further details are provided in the following sections.

6.1 Ethereum Blockchain

Ethereum could be viewed as a state machine in which a transaction would represent a valid transition between states [8]. A transaction is a single cryptographically signed instruction issued by an entity which is tied to an account. There are two types of Ethereum accounts, externally owned accounts (EOA) which belong to an external user and controlled by a private key, and contract accounts which contain and are controlled by the code. Transactions collected into blocks which are chained together via cryptographic hashes to create the blockchain. Each block should be distributed and agreed upon by every node in the network before being added to the chain, using a consensus algorithm. Current version of Ethereum uses proof-of-work, a.k.a PoW, as its consensus algorithm.

It is not a preferable choice to develop and test the smart contracts on the primary public blockchain of Ethereum, a.k.a mainnet, for two reasons. First, because of the immutable nature of blockchain, changing the smart contract code would be a challenging issue as rewriting at transaction level within the blocks is still in its infancy [26]. Second, Ethereum has its own cryptocurrency called *ether* and an internal currency called *gas* to pay the fee of transaction on Ethereum which is proportional to the amount of required computational effort. As any transaction with the mainnet needs gas to be run, buying the real ether (ETH) to provide the gas is prohibitive for testing purposes. Therefore, multiple test networks, a.k.a testnets, have been introduced to test smart contracts before deploying them on the mainnet.

We deployed a smart contract on the Ropsten testnet which is the official Ethereum testnet [11] that uses the same consensus protocol, PoW, as the mainnet. Because many users test their applications on this test network before deployment on the real chain, we recognize it to be a better simulation of a real-world scenario [42]. Therefore, we consider our results to be close enough to the real-world scenario of using the main Ethereum network. Other testnets, such as Rinkeby, Kovan and Görli are using proof-of-authority, a.k.a PoA, which is more time and energy efficient, but different from the mainnet consensus protocol. Therefore, we consider those testnets as nonviable and unreliable to represent the mainnet.

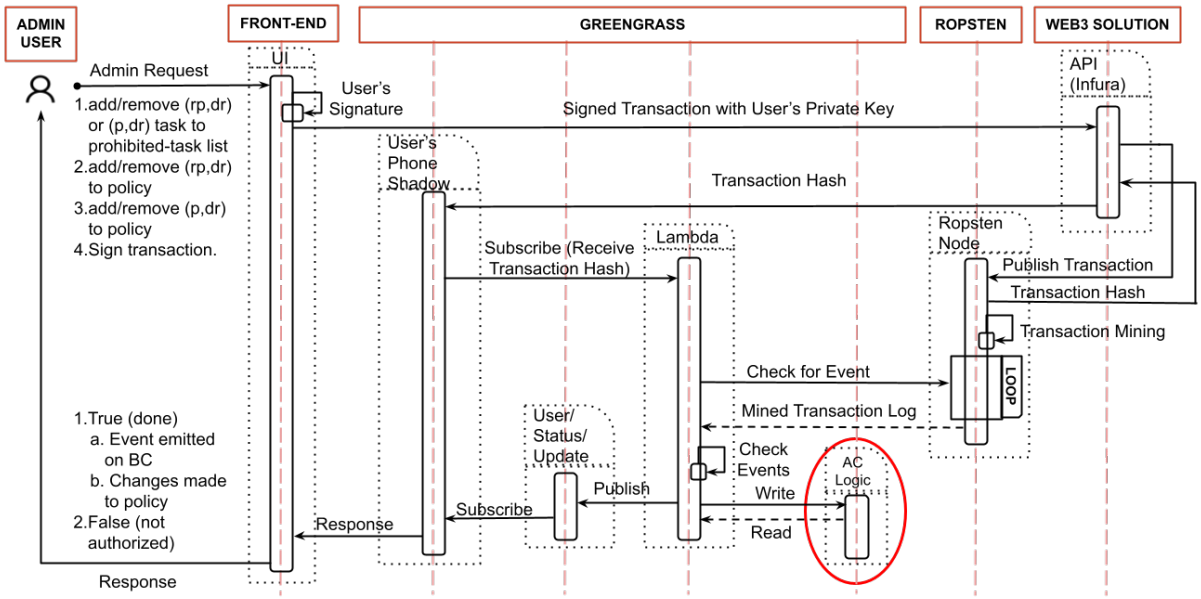

Figure 4: Time-Based Flow of Administration of Access Based On Proposed Architecture

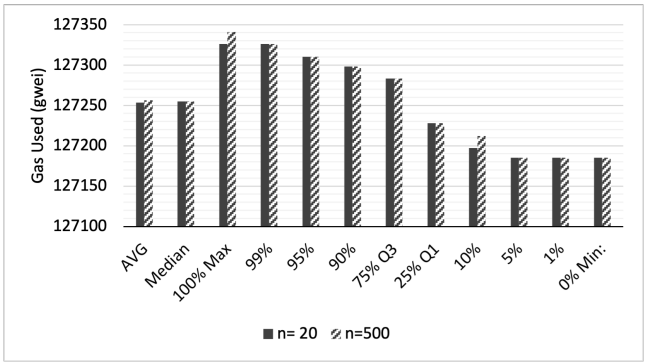

Figure 5: Statistical Summary of Gas Used

6.2 Smart Contract

We implemented administrative access control policy in a single smart contract on the Ropsten blockchain, in which administrative units are predefined and different administrative controls have been coded as functions which would be triggered via transactions. Although the smart contract code is not modifiable after being deployed, it is possible to add/remove data to its stack. So, the administrator can define new tasks to be included in each administrative unit or remove a task from the list of prohibited tasks.

We programmed our smart contract in Solidity and tested it on Remix IDE [14] which is the official browser-based IDE for Ethereum. Administrative units and administrative tasks defined as separate *mapping* data structures in Solidity. To interact with a smart contract, which has been deployed on the Ropsten testnet blockchain, we used a WEB3 API facilitating interaction with the blockchain. We used Infura [13] as the connection point for the web3 API, which hosts some nodes of Ropsten and relays all transactions to the blockchain.

6.3 Experiment Setup

We simulated a use case provided in Figure 3 using AWS IoT Greengrass v1 which runs on a dedicated virtual machine with one virtual CPU, 2 GB of RAM and 20 GB hard drive. The virtual machine's operating system is Ubuntu 20.4.2 LTS and it is connected to a 1 Gbps network. Our AWS lambda code on the Greengrass is written in Python 3.8 and is running in a long-lived isolated runtime environment with limited RAM of 256 MB. Lambda function receives the administrative requests and connects to Infura API to check the transaction status and results, after they have been run on Ropsten testnet. The results would later be reflected on the user's phone via updating its shadow on the Greengrass by lambda function. These results would also be written into the POLICY.JSON file if the administrative request was submitted and approved by smart contract to update the access rules. The POLICY.JSON file would be referred to govern operational accesses in the smart home environment and is shared between operational and administrative Lambda functions and protected by a lock for concurrency control.

6.4 Implementation Results

To evaluate the performance and practicality of our blockchain-based approach for administration of access, we implemented a proof-of-concept under the settings discussed in the previous section. That means each transaction has been sent to Infura and the raw transaction hash is being sent back to the user's phone and the Lambda function. Then, lambda waits for the transaction to be mined and afterwards updates the policy file based on the successful events in the transaction log. The results would also be sent to the user's phone. Experiments are done for a normal distribution with a 99.9% confidence interval. To synchronize timing of the local computer and time servers in case the administrator user wants to make policy changes when away from home, we used Chrony [4].

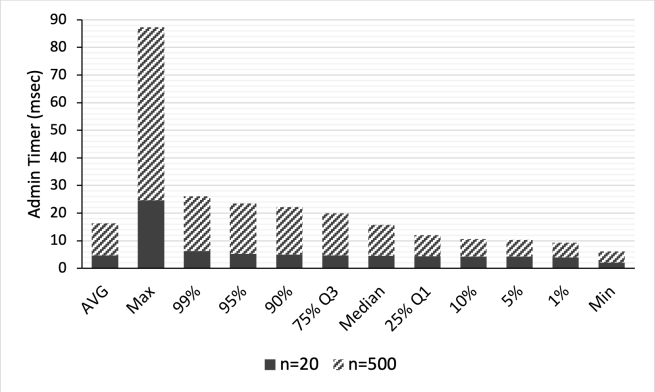

Figure 6: Admin Timer

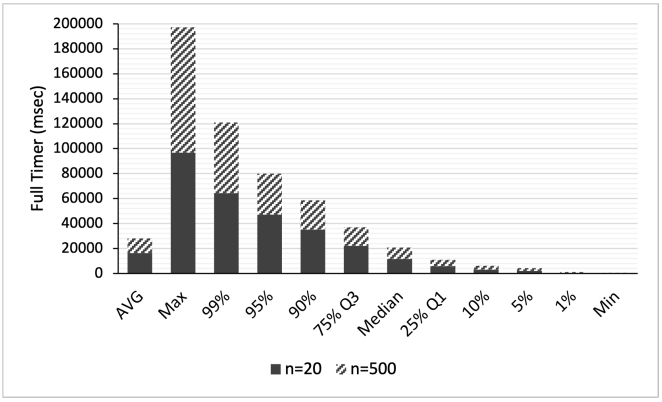

Figure 7: Full Timer

Table 1: Statistical Analysis Results

	Gas Used (gwei)		Admin Timer (ms)		Full Timer (ms)	
	n=20	n=500	n=20	n=500	n=20	n=500
AVG	127254	127256	4.67	11.61	16234.85	12005.34
100% (Max Quantile)	127326	127341	24.75	62.51	96743.43	100602.62
99%	127326	127326	6.29	19.90	64345.05	56523.07
95%	127310	127310	5.22	18.34	47215.98	32847.43
90%	127298	127298	4.95	17.26	34990.34	23773.95
75% (Q3)	127283	127283	4.75	15.19	22015.52	14891.59
Median	127255	127255	4.60	11.20	11667.90	9035.50
25% (Q1)	127228	127228	4.38	7.62	6019.41	4845.63
10%	127197	127212	4.30	6.32	3171.00	2935.90
5%	127185	127185	4.25	6.07	2076.12	2294.55
1%	127185	127185	3.96	5.36	63.50	1097.91
0% (Min Quantile)	127185	127185	2.16	4.04	55.45	68.07

To avoid duplicates, each time a new rule has been administratively requested to be added to the POLICY.JSON file, we check the policy and add the new rule to the policy only if it has no replica in the current policy. We ran our experiments in two settings with the policy sizes of n=20 and n=500. In the first setting, we start with an original policy of size 20 and add one policy in each experiment but keep the maximum size of the policy to be 21. After each experimental run, the original policy with 20 values is reinstated and any changes are dismissed. The second scenario starts with a policy size of 20 but grows incrementally with each policy submission. Both experiments were run for a total of 500 times, resulting in the first case a maximum of 21 policies, and the second case, a policy grew from 20 to the final size of 520.

All the statistical analysis results are provided in Table 1. A visual representation of two important metrics of time and cost are depicted in Figures 5, 6 and 7. Figure 5 shows the required gas for transaction mining on the Ropsten network. The used gas is the actual amount of gas which was used during execution. Gas prices are denoted in GWEI, which equals to 10^{-9} ETH. We calculated the monetary cost of each transaction to be 28 cents, based on the Ether price as of the time of writing this paper.

Based on the results depicted in Figure 6, the difference of average time required for adding a new policy rule (administrative action) would be highly affected by the policy size, which is an indicator of the lambda processing time. After a transaction has been successfully mined, Lambda checks the logs to search out the succeeded transactions. Then, it makes appropriate changes to the POLICY.JSON file and publishes the results to the USER/STATUS/UPDATE to inform the user about his/her administrative request. We call this time *Admin Timer* which is in order of milliseconds.

The Full timer in Figure 7 shows a complete cycle of an administrator submitting a request, to that request being mined, and the lambda function processing the results and updating as necessary. The average total time for an administrative task using blockchain could be estimated as 12.012 seconds. Although this time is unsatisfactory for end users in an operational model, it is quite acceptable for an administrative model, especially compared to the other administrative methods which may take in order of minutes, hours or even days to take effect. On the other hand, with Ethereum moving to Proof of Stake (PoS) as its consensus mechanism, the costs and timing are expected to decrease dramatically [7].

7 Discussion

This section provides properties and limitations of our access administration framework, as well as general security considerations.

7.1 Current Approach Properties and Future Directions

Characteristics and limitations of the chosen underlying policy model [57] have been carried to our framework, added up to its properties/limitations, as discussed below.

Decoupled Assignment and Revocation Any administrator who has been appointed to the corresponding administrative role (AR) of a task, can grant/revoke authorization assignments, no need the grant/revoke to be done by the same administrator.

Symmetric Assignment/Revocation This characteristic enables the AR of an administrative unit to revoke a permission which has been previously conferred by him/her and vice versa.

Generalizability More administrative units could be defined as per users' needs.

Transparency and Auditability Using Ethereum as a public blockchain provides full transparency of transactions as well as access to immutable history logs. Without blockchain, the context of access control decisions would no longer be available; however, blockchain logs provide the posterior auditability. Moreover, the smart contract remains publicly visible on the blockchain even if it would be disabled in the future; in such a case the actual contract remains on the chain but would be marked as not callable.

Privacy The distributed nature of the blockchain eliminates the concern of privacy leakage from a single point of administration. Using blockchain preserves the privacy of the smart home as a whole, because the smart contract is only accessible with users who have their private keys stored on their own devices. Privacy of each user in the smart home withholds through decentralized administration in the form of administrative units (AU); so that each user's privacy zone could be contained in a separate unit while that user has been the only user assigned to corresponding administrative role.

7.2 Security Considerations

Smart Contract Security Benefits of creating decentralized applications (dApp) using smart contracts do not come without costs. As an account-centered model of transactions which is used to identify and communicate with smart contracts on Ethereum, authentication and authorization failures may impose security risks to the system. Ethereum itself is vulnerability-prone, besides the security vulnerabilities which are introduced by unreliability of Solidity [23]. Different verification tools are proposed to analyze the security of deployed smart contracts, a survey of which could be found in [27]. We used Remix IDE [14] for our Solidity smart contract development which performs a static analysis during compilation and reports security vulnerabilities such as implicit typing/visibility, unchecked return values, deprecated constructs, and address checksum and where they occurred in the code. So, we could be sure that our smart contract code is free of vulnerabilities which are checked by Remix. Checking the developed smart contract with other available tools for security vulnerabilities could be considered as a future step, specifically if the contract is going to be generalized for larger environments.

Device-Cloud Communication As the proposed architecture presents cloud-enabled IoT devices, the security of AWS Greengrass and its communication with IoT devices has a great impact on the overall security of our system. IoT devices use X.509 public key infrastructure (PKI) certificates for authentication of devices to Greengrass which are securely tied to AWS IoT policies [5]. We considered best security practices recommended by AWS IoT according which we implemented our architecture in a way that each IoT device has a unique immutable identity stored on it, which would be used to agree on PKI certificates; so, there would be no hard coded credentials in lambda functions [16].

7.3 Restrictions and Future Directions

Proposed framework for smart home IoT environment still needs to be improved to address following restrictions:

Continuous Access Control and Mutability Considering the dynamics of a smart home IoT environment, as a multi-user multi-device environment we need to monitoring the access even after being granted, and sometimes need the immediate change [48]. Moreover, it is required to use access quotas as a consumable non-refundable amount of access to some resource. For instance, the available time for kids to access the PlayStation on a weekend needs to be monitored and access should be revoked immediately (continuous control) as it has been exhausted (mutability).

Conflicts We may not have policy conflicts in that our proposed framework does not include any negative policies, instead to avoid a role from being conferred with specific permissions, we utilize prohibitive assignments. However, it is possible to have administrator interests conflicting. For instance, different homeowners adjust the smart thermostat to different temperature ranges. As users expected the conflicts to be resolved automatically based on the survey of access control needs in a smart home [64], it is highly recommended that a policy resolution algorithm to be incorporated in the access control framework of a smart home [58].

8 Conclusion

This paper presents an architectural enforcement of access administration for a smart home IoT environment. Our architecture is based on Ethereum blockchain and hence is decentralized, auditable, and reliable. Our implementation results are reassuring that although the use of blockchain for operational access control is not promising, an administrative model could successfully utilize the benefits of blockchain. Our proposal properties and limitations and some future directions of research have also been discussed.

Acknowledgment

This work is partially supported by NSF CREST Grant 1736209.

References

[1] 2015. Saving the future of the Internet of Things. https://www.ibm.com/downloads/cas/Y5ONA8EV

[2] 2021. Amazon GreenGrass. https://docs.aws.amazon.com/greengrass/
[3] 2021. Bitcoin. bitcoin.org/en/
[4] 2021. Chrony. https://chrony.tuxfamily.org/
[5] 2021. Device authentication and authorization for AWS IoT Greengrass. https://docs.aws.amazon.com/greengrass/v1/developerguide/device-auth.html
[6] 2021. EOSIO. https://eos.io/
[7] 2021. Ethereum 2.0 Proof of Stake. https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/
[8] 2021. ETHEREUM: A Secure Decentralised Generalised Transaction Ledger, EIP-150 REVISION. https://gavwood.com/paper.pdf
[9] 2021. Ethereum Average Block Time Chart. https://etherscan.io/chart/blocktime
[10] 2021. Ethereum Races Clock to Collect Enough Coins for Big Upgrade. https://www.bloomberg.com/news/articles/2020-11-23/ethereum-races-clock-to-collect-enough-coins-for-huge-upgrade
[11] 2021. EthereumNetworks. https://ethereum.org/en/developers/docs/networks/
[12] 2021. Hyperledger Fabric. https://www.hyperledger.org/use/fabric
[13] 2021. Infura. https://infura.io/product
[14] 2021. Remix IDE. https://remix.ethereum.org/
[15] 2021. Ripple. https://www.ripple.com/
[16] 2021. Security best practices for AWS IoT Greengrass. https://docs.aws.amazon.com/greengrass/v1/developerguide/security-best-practices.html
[17] Gail-Joon Ahn, Hongxin Hu, and Jing Jin. 2008. Towards role-based authorization for osgi service environments. In *Future Trends of Distributed Computing Systems*. IEEE.
[18] Asma Alshehri and Ravi Sandhu. 2016. Access control models for cloud-enabled internet of things: A proposed architecture and research agenda. In *Collaboration and Internet Computing (CIC)*. IEEE.
[19] Safwa Ameer, James Benson, and Ravi Sandhu. 2020. The EGRBAC Model for Smart Home IoT. In *Information Reuse and Integration for Data Science (IRI)*. IEEE.
[20] Smriti Bhatt, Farhan Patwa, and Ravi Sandhu. 2017. Access control model for AWS internet of things. In *International Conference on Network and System Security*. Springer.
[21] Leepakshi Bindra, Changyuan Lin, et al. 2019. Decentralized access control for smart buildings using metadata and smart contracts. In *International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)*. IEEE.
[22] Vitalik Buterin et al. 2014. A next-generation smart contract and decentralized application platform. *white paper* (2014).
[23] Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu. 2020. A survey on ethereum systems security: Vulnerabilities, attacks, and defenses. *ACM Computing Surveys (CSUR)* (2020).
[24] Jason Paul Cruz, Yuichi Kaji, and Naoto Yanai. 2018. RBAC-SC: Role-based access control using smart contract. *IEEE Access* (2018).
[25] MAC Dekker et al. 2008. RBAC administration in distributed systems. In *Symposium on Access Control Models and Technologies*. ACM.
[26] David Derler, Kai Samelin, Daniel Slamanig, and Christoph Striecks. 2019. Fine-Grained and Controlled Rewriting in Blockchains: Chameleon-Hashing Gone Attribute-Based. *IACR Cryptol.* (2019).
[27] Monika Di Angelo and Gernot Salzer. 2019. A survey of tools for analyzing Ethereum smart contracts. In *Decentralized Applications and Infrastructures*. IEEE.
[28] Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, Gang Chen, Beng Chin Ooi, and Ji Wang. 2018. Untangling blockchain: A data processing view of blockchain systems. *IEEE Transactions on Knowledge and Data Engineering* (2018).
[29] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. 2017. Blockchain for IoT security and privacy: The case study of a smart home. In *International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE.
[30] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. 2019. LSB: A Lightweight Scalable Blockchain for IoT security and anonymity. *J. Parallel and Distrib. Comput.* (2019).
[31] Sofia Dutta, Sai Sree Laya Chukkapalli, et al. 2020. Context sensitive access control in smart home environments. In *Big Data Security on Cloud*. IEEE.
[32] Hao Guo, Ehsan Meamari, and Chien-Chung Shen. 2019. Multi-authority attribute-based access control with smart contract. In *International Conference on Blockchain Technology*. ACM.
[33] Sergio Gusmeroli, Salvatore Piccione, and Domenico Rotondi. 2013. A capability-based security approach to manage access control in the internet of things. *Mathematical and Computer Modelling* (2013).
[34] Sayed Hadi Hashemi, Faraz Faghri, Paul Rausch, and Roy H Campbell. 2016. World of empowered IoT users. In *First International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE.
[35] Weijia He, Maximilian Golla, Roshni Padhi, Jordan Ofek, Markus Dürmuth, Earlence Fernandes, and Blase Ur. 2018. Rethinking access control and authentication for the home internet of things (IoT). In *27th {USENIX} Security Symposium*.
[36] Vincent Hu. 2021. *Blockchain for Access Control Systems*. Technical Report. National Institute of Standards and Technology (NIST).
[37] MD Azharul Islam and Sanjay Madria. 2019. A permissioned blockchain based access control system for IOT. In *Blockchain*. IEEE.
[38] Tiffany Hyun-Jin Kim, Lujo Bauer, James Newsome, Adrian Perrig, and Jesse Walker. 2010. Challenges in Access Right Assignment for Secure Home Networks.. In *HotSec*.
[39] Nir Kshetri. 2017. Can blockchain strengthen the internet of things? *IT professional* (2017).
[40] Han Liu, Dezhi Han, and Dun Li. 2020. Fabric-IoT: A blockchain-based access control system in IoT. *Access* (2020).
[41] Damiano Di Francesco Maesa, Paolo Mori, and Laura Ricci. 2017. Blockchain based access control. In *IFIP international conference on distributed applications and interoperable systems*. Springer.
[42] Damiano Di Francesco Maesa, Paolo Mori, and Laura Ricci. 2019. A blockchain based approach for the definition of auditable access control systems. *Computers & Security* (2019).
[43] Yuta Nakamura, Yuanyu Zhang, Masahiro Sasabe, and Shoji Kasahara. 2019. Capability-based access control for the internet of things: an ethereum blockchain-based scheme. In *Global Communications Conference (GLOBECOM)*. IEEE.
[44] Oscar Novo. 2018. Blockchain meets IoT: An architecture for scalable access management in IoT. *Internet of Things Journal* (2018).
[45] Aafaf Ouaddah, Anas Abou El Kalam, and Abdellah Ait Ouahman. 2017. Harnessing the power of blockchain technology to solve IoT security & privacy issues.. In *ICC*.
[46] Aafaf Ouaddah, Anas Abou Elkalam, and Abdellah Ait Ouahman. 2016. FairAccess: a new Blockchain-based access control framework for the Internet of Things. *Security and communication networks* (2016).
[47] Aafaf Ouaddah, Anas Abou Elkalam, and Abdellah Ait Ouahman. 2017. Towards a novel privacy-preserving access control model based on blockchain technology in IoT. In *Europe and MENA cooperation advances in information and communication technologies*. Springer.
[48] Jaehong Park and Ravi Sandhu. 2004. The UCON_ABC usage control model. *Transactions on information and system security (TISSEC)* (2004).
[49] Otto Julio Ahlert Pinno, Andre Ricardo Abed Gregio, and Luis CE De Bona. 2017. Controlchain: Blockchain as a central enabler for access control authorizations in the iot. In *GLOBECOM Global Communications Conference*. IEEE.
[50] Jing Qiu, Zhihong Tian, et al. 2020. A survey on access control in the age of internet of things. *Internet of Things Journal* (2020).
[51] Mohsin Ur Rahman. 2020. Scalable Role-Based Access Control using the EOS Blockchain. *arXiv preprint arXiv:2007.02163* (2020).
[52] Ana Reyna, Cristian Martín, Jaime Chen, Enrique Soler, and Manuel Díaz. 2018. On blockchain and its integration with IoT. Challenges and opportunities. *Future generation computer systems* (2018).
[53] Sara Rouhani and Ralph Deters. 2019. Blockchain based access control systems: State of the art and challenges. In *IEEE/WIC/ACM International Conference on Web Intelligence*.
[54] Ravi Sandhu. 2009. The PEI framework for application-centric security. In *Collaborative Computing: Networking, Applications and Worksharing*. IEEE.
[55] Ravi S Sandhu and Pierangela Samarati. 1994. Access control: principle and practice. *Communications Magazine* (1994).
[56] Savio Sciancalepore, Giuseppe Piro, Daniele Caldarola, Gennaro Boggia, and Giuseppe Bianchi. 2017. OAuth-IoT: An access control framework for the Internet of Things based on open standards. In *Symposium on Computers and Communications (ISCC)*. IEEE.
[57] Mehrnoosh Shakarami and Ravi Sandhu. 2021. Role-Based Administration of Role-Based Smart Home IoT. In *Workshop on Secure and Trustworthy Cyber-Physical Systems*. ACM.
[58] Amit Kumar Sikder, Leonardo Babun, Z Berkay Celik, Abbas Acar, Hidayet Aksu, Patrick McDaniel, Engin Kirda, and A Selcuk Uluagac. 2020. Kratos: multi-user multi-device-aware access control system for the smart home. In *Security and Privacy in Wireless and Mobile Networks*. ACM.
[59] Liang Tan, Na Shi, Keping Yu, Moayad Aloqaily, and Yaser Jararweh. 2021. A Blockchain-empowered Access Control Framework for Smart Devices in Green Internet of Things. *Transactions on Internet Technology (TOIT)* (2021).
[60] Yuan Tian, Nan Zhang, Yueh-Hsun Lin, XiaoFeng Wang, Blase Ur, Xianzheng Guo, and Patrick Tague. 2017. Smartauth: User-centered authorization for the internet of things. In *26th {USENIX} Security Symposium*.
[61] Karl Wüst and Arthur Gervais. 2018. Do you need a blockchain?. In *Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE.
[62] Ronghua Xu, Yu Chen, Erik Blasch, and Genshe Chen. 2018. Blendcac: A blockchain-enabled decentralized capability-based access control for iots. In *Internet of Things (iThings)*. IEEE.
[63] Eric Zeng, Shrirang Mare, and Franziska Roesner. 2017. End user security and privacy concerns with smart homes. In *thirteenth symposium on usable privacy and security ({SOUPS})*.
[64] Eric Zeng and Franziska Roesner. 2019. Understanding and improving security and privacy in multi-user smart homes: a design exploration and in-home user study. In *28th {USENIX} Security Symposium*.
[65] Yuanyu Zhang, Shoji Kasahara, et al. 2018. Smart contract-based access control for the internet of things. *Internet of Things Journal* (2018).