symmetry

MDPI

Review

Blockchain Platforms and Access Control Classification for IoT Systems

Adam Ibrahim Abdi [1,*], Fathy Elbouraey Eassa [1], Kamal Jambi [1], Khalid Almarhabi [2] and Abdullah Saad AL-Malaise AL-Ghamdi [3]

[1] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University (KAU), Jeddah 21589, Saudi Arabia; feassa@kau.edu.sa (F.E.E.); kjambi@kau.edu.sa (K.J.)

[2] Department of Computer Science, College of Computing at Alqunfudah Umm Al-Qura University, Makkah 21514, Saudi Arabia; kamarhabi@uqu.edu.sa

[3] Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University (KAU), Jeddah 21589, Saudi Arabia; aalmalaise@kau.edu.sa

* Correspondence: aabdi0002@stu.kau.edu.sa

Received: 6 September 2020; Accepted: 3 October 2020; Published: 12 October 2020

check for updates

Abstract: The Internet of Things paradigm is growing rapidly. In fact, controlling this massive growth of IoT globally raises new security and privacy issues. The traditional access control mechanisms provide security to IoT systems such as DAC (discretionary access control) and mandatory access control (MAC). However, these mechanisms are based on central authority management, which raises some issues such as absence of scalability, single point of failure, and lack of privacy. Recently, the decentralized and immutable nature of blockchain technology integrated with access control can help to overcome privacy and security issues in the IoT. This paper presents a review of different access control mechanisms in IoT systems. We present a comparison table of reviewed access control mechanisms. The mechanisms' scalability, distribution, security, user-centric, privacy and policy enforcing are compared. In addition, we provide access control classifications. Finally, we highlight challenges and future research directions in developing decentralized access control mechanisms for IoT systems.

Keywords: access control (AC); blockchain; Internet of Things (IoT); AC classification; security; privacy

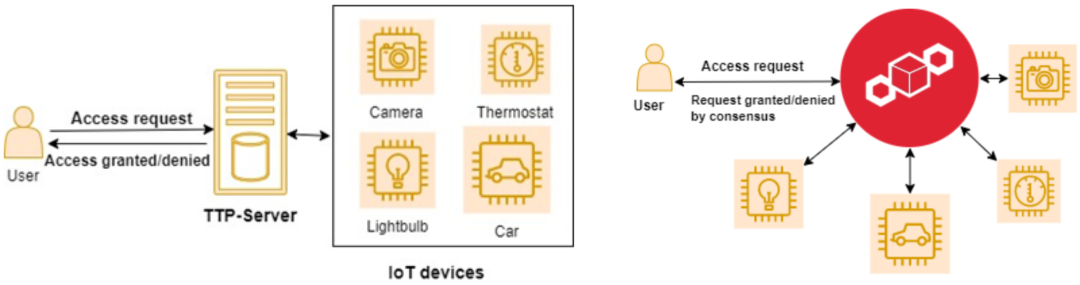
1. Introduction

The Internet of Things (IoT) refers to digital objects connected to the Internet. The IoT technology takes the main role in enhancing our lifestyle, based on different domains such as health care, smart cities, transportation, and Smart Grids. Those interconnected IoT devices are continuously growing to 20.4 billion IoT connections by the year 2022 [1]. These large numbers of IoT can share information with other entities, as well as generate collected data from surrounding environments. To perform these tasks, the IoT devices must have an access control system that can filter who can access its resources, as well as how it can be accessed (e.g., grant permission). Classical access control systems like discretionary access control (DAC) [2,3], mandatory access control (MAC) [4,5] and RBAC [6,7] have been used to protect resources, but these approaches are under a single authority or centralized management, which pose some challenges, such as single point of failure and privacy issues. However, the centralized access controls cannot provide a suitable efficiency access management to IoT, due to these issues.

- Some IoT scenarios need multiple managers in the supply chain [8], while other IoT scenarios may have various domain controllers as vehicle-to-vehicle.

- IoT devices are constrained resources and cannot handle heavy computational access managements.
- Access control must be scalable to handle exponentially growing IoT devices.
- Centralized IoT architecture can suffer a single point of failure, due to denial of service attacks.
- In the IoT system based on centralized management, service providers can access and analyze user data, which can cause security and privacy issues.

Furthermore, to overcome these challenges, researchers have begun to work with decentralized environments over IoT access management. A decentralized technology ensures reliability, privacy, and scalability, which are more suitable for fast-growing IoT environments. Recently, blockchain technology is the concept of decentralized architecture that is more suitable for the ecosystem for IoT [9], due to immutability, distributed ledgers, and sharing information, so it can handle and manage transactions without relying on third parties. A classical and blockchain-based access control is presented in Figure 1.



(**a**) Classical AC (**b**) Blockchain Based AC

Figure 1. (**a**,**b**) are classical and blockchain-based IoT access management.

However, the access control approaches for IoT must include these requirements [10]:

- Scalability: the IoT systems need to handle a huge number of devices, users, and access management policies, due to exponential growth of IoT. Therefore, the access control mechanism must be able to deal with large IoT systems.
- Dynamicity: the access control mechanism should be able to support flexible attributes.
- Lightweight: access control mechanisms must have low overhead communication and computation to adapt to constrained devices.

2. Blockchain: Principles and Main Platforms

A blockchain is an immutable ledger that contains a set of linked blocks, which has been validated by the participators (miners) in the peer-to-peer (P2P) network. The concept of blockchain has been rising after the introduction of Bitcoin [11], which is an online cryptocurrency that does not have central authority to manage its transactions, but maintains and validates the blocks of the miners in the decentralized peer-to-peer network. Researchers have been attracted to the ubiquity of Bitcoin and its decentralized concept of block management and have started to focus on blockchain technology in depth. Basically, the blockchain is a decentralized distributed and tamper-proof [12,13] ledger that is shared among every P2P network participant.

2.1. Main Characteristics of the Blockchain

Blockchain technology has potential features [14], such as decentralization, traceability, and being tamper-proof. These features can improve different IoT applications, such as smart cars, smart cities, and healthcare systems, by providing secure data sharing without third parties, as well as traceability and transparency.

- Decentralization: The decentralized nature of blockchain enables all the members of the blockchain network to participate in the process of validating transactions, unlike centralization, which allows only the administrator of the network to perform the authorization and validation processes.

- Traceability: It is easy to audit, because all actors in the blockchain have copies of the transactions in the ledger. So, the actors in the blockchain network can validate data exchange (transaction) for a particular blockchain address. Each record stored in a blockchain is assigned a timestamp, subsequently guaranteeing transaction traceability. In addition to ensuring the privacy of users, the blockchain offers a kind of pseudo-anonymity.
- Tamper-proof: New joining blocks in the blockchain are authorized and validated by all peers in the P2P network through decentralized consensus mechanisms. Therefore, the blockchain is immutable; for example, if an attacker tries to change any record in the blockchain, this would require accommodating the majority of the participants in the network and otherwise would be detected easily.
- Transparency: All the actors in the public blockchain frameworks (ex. Bitcoin and Ethereum) have the same access rights, so they would participate in the process of validating and recording new transactions in the blockchain. Therefore, the recorded data in the ledger would be transparent to all of the actors in the blockchain network.

2.2. Types of Blockchain Systems

Blockchain technology is a decentralized platform that enables data sharing among participants across a peer-to-peer network. Blockchains can be classified [14] based on their architectural characteristics and their quality attributes such as partially decentralized (permissioned blockchain) and fully decentralized (permission-less blockchain). In addition, the blockchain can be categorized [15] into private blockchain, public blockchain, or hybrid blockchain, based on various principles, such as authentication as well as access control mechanisms. Table 1 presents a comparison table of the main blockchain platforms. The platforms are compared in terms of blockchain type, consensus mechanism, smart contract, transaction capacity, forks, lack of permission and lack of fees.

Table 1. Comparison Table of Blockchain Platforms.

	Bitcoin [11,13]	**Ethereum [16]**	**Hyperledger Fabric [17]**	**Hyperledger Burrow [18]**	**Ripple [19]**
Blockchain-Type	Public	Public / Private	Private / Consortium	Private / Consortium	Private
Consensus	PoW	PoW/PoS	PBFT	Tendermint	BFT(RPCA)
Smart contract	No	Yes	Yes	Yes	Yes
Capacity	7 tps	12 tps	Thousands tps	Thousands tps	Thousands tps
Forks	Yes	Yes	No	No	No
Permission-less	Yes	Yes/No	No	No	No
Fee-less	No	No	Yes	Yes	No

2.2.1. Permission-Less Blockchain (Public Blockchain)

In the permission-less or public blockchain, there is no any restriction on participators joining the blockchain network, and they can participate in validating transactions, as well as in the mining process [13]. Each peer in a public blockchain network has full access rights to participate in the process of validating transactions, as well as maintaining block ledgers.

Bitcoin cryptocurrency [11] is one of the most well-known examples of public blockchains. It is designed for huge networks that contain large members of anonymous peers. Therefore, the public blockchain has high computations to validate blocks in the network, because all peers are participating in the process. Proof of work (PoW) is one of the best solutions to solve challenges of public blockchains such as preventing alteration of its data, while this consensus needs to obtain 51% of participants or miners to manage the network. However, the PoW consensus mechanism is not suitable for constrained devices like IoT, due to its high computational complexity.

2.2.2. Permissioned Blockchain (Private Blockchain)

Permissioned (or private) blockchains are the opposite to public blockchains, as they are restricted networks, and each peer joining the network must be authorized by the organization. The mining processes of permissioned blockchains are controlled by authenticated participants. It uses peer-to-peer to notify members of block transactions among participants. A public blockchain needs currency or tokens for the handling of transactions, while a private blockchain does not need it. There are various private blockchain systems, like Hyperledger Fabric [17] and Ripple [19].

In Hyperledger Fabric, all participants in the permissioned blockchain are known and reach a consensus by utilizing the Practical Byzantine Fault Tolerance (PBFT) algorithm [20]. PBFT is one of the consensus mechanisms of the permissioned blockchains that allow only permissioned peers to participate in the process of validating transactions in the network. Ripple also reaches a consensus mechanism by BFT customized in the Ripple Protocol Consensus Algorithm (RPCA) [21].

2.2.3. Consortium Blockchain (Hybrid Blockchain)

A consortium blockchain is a combination of private and public blockchains; the consensus mechanisms of this blockchain are managed by a set of organizations or participants to ensure the validation of transactions [22]. The consortium blockchain allows a group of individuals or organizations to validate blocks, instead of having everyone participate in the process or having only a single entity decide the validation process. Hyperledger Fabric [17] and Hyperledger Burrow [18] are examples of consortium blockchain frameworks. For the consensus mechanism, the consortium blockchain uses consensus algorithms, such as PBFT and Byzantine fault tolerance (BFT) consensus through the Tendermint algorithms, which are not expensive computationally, to validate transactions.

In conclusion, the permission-less blockchain involves high computations to publish new blocks in the network. So, it is not suitable for constrained IoT devices. Therefore, private and consortium blockchains are more suitable for IoT, due to low latency and high transaction [23].

2.3. Smart Contract

The term "smart contract" was proposed by Nick Szabo, and is based on a set of promises and protocols with the aim of securing relationships with public networks [24]. Smart contracts contain contractual clauses that will self-execute as soon as the required conditions are satisfied.

Initially, when blockchain technology started with cryptocurrency like Bitcoin, the smart contract was unavailable. However, later, blockchain systems such as Hyperledger [25] and Ethereum [16] started to support the smart contract concept, due to the contract programmability implemented by them. Basically, the smart contracts are built on newer blockchains. Smart contracts ensure suitable access management and enforcement. The executed contracts are stored as tamper-proof blocks in the blockchain network [26]. The smart contract contributes some advantages to blockchain technology, such as self-execution, transparency, efficiency and removal of third party roles. In addition, the smart contract contributes to the IoT by ensuring security and reliability for all processes carried out in IoT systems, as well as recording and controlling all interactions of IoT entities. Therefore, smart contracts are more applicable to IoT applications.

3. Access Control Classification for IoT Systems

Access control (AC) is a mechanism that enables only authenticated entities to access the resources, as well as the authority to control those resources. In the IoT environment, access control can be classified into centralized and decentralized, and each can be categorized into subparts, as shown in Figure 2.

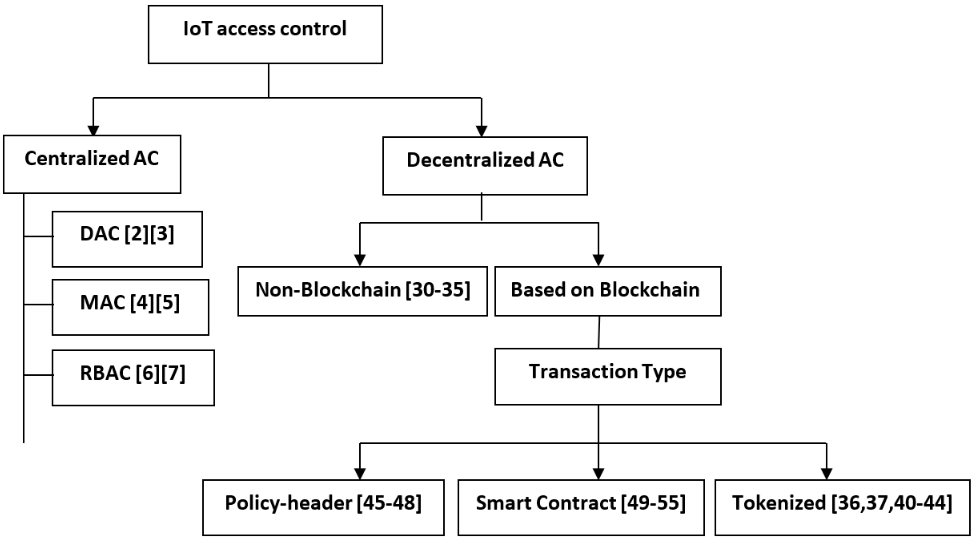


Figure 2. Taxonomy of IoT Access Control Systems.

3.1. Centralized Access Control for IoT

DAC (discretionary access control) [2,3] is one of the centralized access management types that is broadly used for access control. DAC grants legitimate users access to objects based on their identity or user groups. Users can autonomously give their privilege or authority to any other user. DAC was built on the concept of the relationship among users and objects. The access control decisions depend on the subject's access rights. The access matrices represent the access rights that have been assigned to each object. DAC is a simple and flexible AC, and therefore is utilized in real life in IoT deployments, such as where IoT resources are identified by its MAC (media access control) address.

However, it is vulnerable to malicious programs such as trojan horses [27], as well as depending on a central entity, which can lead to a single point of failure. Furthermore, the owner cannot distinguish between legitimate access requests from authorized users and illegitimate access requests from malicious programs.

Mandatory access control (MAC) is a model policy that determines which parties are permitted to have accessibility rights, and normal users cannot modify the access rules. It completely depends on the central administration [4,28]. This model creates restrictions or security labels [5] to access sensitive data. Sensitive data can be classified as top-secret, secret, and confidential, as well as the trust of the user depending on its trust in the CA (Certificate Authority). In addition, it uses the Bell-LaPadula model to control information flow and allows low dates to be noticeable at high volumes. Therefore, this model guarantees managing authorities, as well as preventing security problems. However, MAC is a crude security management model, and is controlled by a central entity, as well as lacking a dynamic practice. Consequently, this model is not suitable to constrained IoT environments.

Role-based access control (RBAC) started with multi-user and multi-application online frameworks, spearheaded in the 1970s [6]. The RBAC is an access control model that associates permissions with roles, and then assigns users to suitable roles based on their capabilities and responsibilities provided by the organization. The RBAC allows users to provide special roles associated with a particular permission, while DAC and MAC assign users a particular permission. In Figure 3, we illustrate an example of RBAC and how roles are hierarchized. RBAC has these main benefits:

- The role in RBAC most specifies the user's permission in the system.
- Hierarchical roles in RBAC specify the structure of the organization, as well as describing the authorization [7,29]. The whole permissions in the roles of the leaves (children) are included in the roles of the roots (parents).

- Least privilege means giving roles to users to do special tasks.

However, this model does not cover the dynamic and distributed environment in which the location and time are frequently changing. Furthermore, RBAC was developed based on the DAC model, thus inheriting all its limitations. Table 2 shows a comparison table of centralized access control. The comparison is presented in terms of the model and its strengths and weaknesses. The DAC model is good for managing authorizations, due to its flexibility and the user's responsibility for determining it. Nonetheless, the DAC is more vulnerable to trojan horses. The DAC can be applied to small systems.

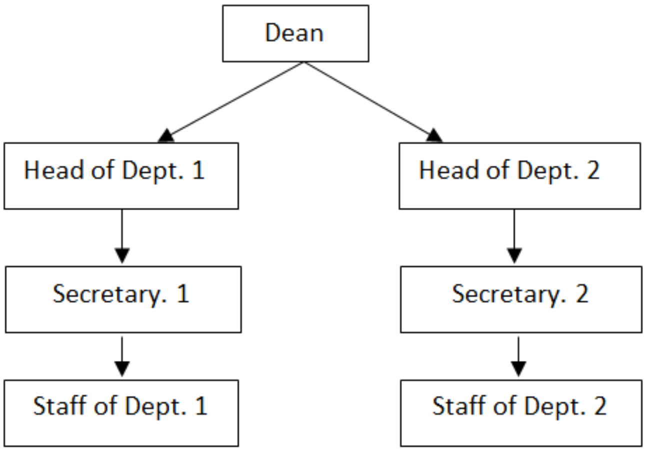


Figure 3. An example of how RBAC roles are hierarchized.

The MAC access control model is more secure and can resist malicious software like trojan horses. However, MAC is a crude security management model without dynamic practice. MAC is more suitable to military systems. RBAC models overcome the weaknesses of MAC and DAC models, by adding some useful features such as hierarchical roles to separate tasks, as well as to describe privileges as least privilege, which allows the user to do a particular task. It can apply to small systems that have hierarchical responsibilities. However, RBAC does not support dynamic environments in terms of time and location. Finally, these models are based on centralized authority and therefore can lead to single point of failure, as well as privacy issues.

Table 2. Comparison Table of Centralized IoT Access Control Systems.

AC Models	Strengths	Weaknesses
DAC [2,3]	-It is easy to implement. -Users that have permission can decide access rights to their resources.	-It does not support the IoT that don't have device identifiers. -It is vulnerable to malicious programs like a Trojan Horse.
MAC [4,5,28]	-It is highly secure, due to the creation of restrictions (security labels). -It can resist malicious software like a Trojan Horse.	-It is difficult and costly to implement and maintain. -It does not adapt to the constrained environment.
RBAC [6,7,29]	-Least privilege. -It allows the creation of hierarchal levels of permission with inheritance.	-It is not suitable for large-scale systems. -It does not support dynamic attributes like time and location. -It is not user-driven.

3.2. Decentralized Access Control for IoT

Decentralized AC can solve issues of authentication, authorization, and validation. However, centralized access control has failed to manage rapidly growing IoT devices with expanding systems. Furthermore, classical AC are under a single authority, which leads to single point of failure, as well as privacy issues. Therefore, decentralized access controls can help to resolve drawbacks of the centralized access controls. In this section, we classified decentralized access control for IoT into two main categories: namely, non-blockchain-based access control and blockchain-based access control, as illustrated in Figure 2.

3.2.1. Non-Blockchain Distributed Access Control for IoT

In this approach, access control is distributed without trusting a central authority over end users. SmartOrBAC [30,31], based on the OrBAC (organization-based access control) model, is a distributed access management model for IoT environments. The SmartOrBAC enhances the existing OrBAC [32] in context-aware concerns, which adapts to the constrained resources of the IoT environments. SmartOrBAC distributes the process of access control into different functional layers.

- Constrained layer: This is the most constrained level that contains the objects and the subjects.
- Less constrained layer: This layer of access control can perform tasks that have higher computation.
- Organization layer: This layer is responsible for specifying policies, as well as roles and privileges of the organization's entities.
- Collaboration layer: This layer allows various organizations to agree among themselves to interact with each other, by utilizing the Principal Authorization Manager (PAM) in the organization layer.

The access control approaches should be lightweight because IoT cannot handle security mechanisms that have high computation and communication, due to its constrained power and storage. However, the SmartOrBAC does not support a lightweight approach to overcome limitations of OrBAC to adapt to IoT environments.

Capability-based access control (CapBAC) is one of the access management models for the IoT, which is capable of granting access rights by one entity to another entity. The capability can be either a ticket, key, or token that provides access to an entity or an object in a computer system [33]. Moreover, the token should be immutable, as well as clearly identified to adapt to real scenarios. Alternatively, it could be described as a set of rights that grants the subject the token. Additionally, CapBAC is a subject that wishes to get to certain data from an object. So, they need to send a token to request processing; then, the object recognizes the permission level of the subject (requester) that has already been granted to allow the processing of the request. This simplified authorization ensures that the distribution of access control mechanisms are easy. In [34], three different access control systems for the Internet of Things were implemented, namely centralized mechanisms, centralized and contextual mechanisms, and decentralized mechanisms. The authors discussed each mechanism and its strengths and weaknesses. Furthermore, the distributed approach is more applicable to IoT environments, due to its scalability and interoperability. However, this approach suffers from lack of dynamicity, as well as not being considered context-aware management. The authors [35] developed access management for IoT based on CapBAC, which is a part of the European FP7 IoT@Work project. This access management supports delegations, as well as giving more flexibility than centralized approaches such as RBAC. However, the CapBAC has some drawbacks, as it is not fine-grained access control as well as not considering context. Furthermore, it needs to issue capabilities to all entities in the authorization process for large-scale systems. We illustrate in Table 3 the strengths and weaknesses of non-blockchain distributed access control approaches of IoT.

Table 3. Comparison Table of Non-Blockchain Distributed Access Control.

AC Mechanism	Strengths	Weaknesses
SmartOrBAC [30,31]	- Contains improved "context" notation to adapt to IoT environments. - Enhances security policy management.	-It does not support lightweight mechanism.
CapBAC [34,35]	-Least privilege. - Supports delegation and revocation. - Distributed. -Usable and flexible.	-It must issues capabilities to all entities in large scale systems. -It does not consider context.

3.2.2. Blockchain-Based Decentralized Access Control for IoT

This section provides a review of the relevant literature on decentralized access control for IoT based on blockchain. It contains a classification of access control based on transaction methods of the blockchain, as shown in Figure 2. In addition, it provides a summary and comparison in Tables 4 and 5.

Token-Based Access Control for IoT Based on Blockchain

Tokenization can be defined as a digital signature that holds accessing privileges created by the owner of the object to his requestor, in order to allow access to a particular object based on his address. Therefore, a user without a token cannot access the owner's resources. IoT access control based on tokenization with blockchain technology brings many benefits in IoT environments, such as decreasing communication costs with validating only once by utilizing a signature, and tokens can be represented in different access-control (AC) actions. In addition, blockchain technology facilitates the process of validating and verifying tokens by the IoT devices without a third party, which diminishes security issues, such as single point of failure as well as privacy preservation. Many researchers have mentioned IoT access control based on blockchain with the tokenized approach, so we illustrate and summarize with a comparison table of these approaches.

FairAccess [36] presents a completely decentralized access control and authorization management structure for IoT devices. This proposed framework defines a token for authorization; therefore, the defined token as a digital signature serves as the authority permitting requesters to access a particular object based on their address. In fact, according to this proposed framework, the blockchain is represented as a database that keeps all access approach policies by combining pairs (target-object, requester) and forming transactions. In addition, it acts as a logging database, which guarantees auditing work. In the FairAccess system, the authorization token is used to provide access to the requester or receiver to a particular object based on its identification address. On the other hand, the smart contract is utilized by expressing the policies of access control as a scripting language. However, this framework has some drawbacks such as authorization through tokenization only and terminating a token or a new request for access must be communicated to the owner, which causes a delay. Furthermore, there is no integration between access control and legitimate relationship networks that features an enormous significance in a combinable and integrated IoT.

IoTChain [37] is a fully decentralized access control architecture for IoT resources. The IoTChain architecture is the integration of the Object Security Architecture (OSCAR) framework [38], which produces keys for users and sends them to owners to secure their resources. The Authentication and Authorization for Constrained Environments (ACE) structure [39] is used to authorize requesters to access IoT resources. This proposed framework provides an end-to-end solution to authorize access to IoT resources securely. In the IoTChain approach, the owner of the resource defines access privileges in smart contracts, which self-execute to generate access tokens to the users when the required conditions are satisfied. A request is made by the client to the key server in order to join the blockchain network, and then it checks the client is authorized or not. When the client authorizes and receives group keys, then he downloads and decrypts the encrypted resources. However, the framework has some issues, such as low scalability, and assessment of its performance, as well as the fact that robustness requires implementing various applications over the IoTChain system.

Outchakoucht et al. [40] proposed a framework for IoT access control, based on blockchain and machine learning algorithms. In this system, the resource owner defines the access control policies in the smart contract to control access to his resources. When a user sends a request to access a resource, the smart contract is automatically executed if the request meets certain conditions, and then it generates a token to the requestor for authorization. In addition, this system utilizes the blockchain to guarantee decentralized access control, while the reinforcement learning algorithm dynamically optimizes access policies. However, this framework is not implemented and does not demonstrate its feasibility for privacy and security testing.

Xue et al. [41] presented an access management for smart homes based on private blockchain. In this system, the administrator has a smart home that identifies policies of access control and manages constrained devices, as well as authenticating visitors. A visitor initializes a transaction to access smart devices, and the administrator authenticates the visitor based on its stored public key, as well as checking his access rights based on the policy header of a private blockchain; when the transaction is validated, the administrator generates a token and key to the visitor to access the IoT devices in the home. In addition, the proposed solution accesses all records kept on the blockchain to avoid alteration, due to its immutability. Although the proposed solution enhances the security of accessing the smart home devices, its policies on access control are not self-enforced, and do not mention how to secure and trace outsourced data as off-blockchain storage.

Maesa et al. [42] presented an access control mechanism based on blockchains, that produces policies as well as control access rights to resources. In this approach, the resource owner creates two different tokens based on Bitcoin transactions; the first is used to transfer access rights between subjects, while the second is used for updating or revoking policies credited to the owner. This framework stores the actual policies and user attributes in an external system, while the blockchain contains a link from that external system. The communication between the system components, such as external system, resource owner, and subjects, is based on the policy decision, which is either accept or reject. Policy outsourcing reduces space complexity in the blockchain, but does not benefit from blockchain features such as availability, security, or immutability. In addition, policy enforcement in this model is not self-executed.

BlendCAC [43] is a capability-based access control system for the IoT, based on blockchains. In this model, the domain coordinator is a resource owner who defines policies and controls services and devices, while the cloud has globally controlled access policies and provides services. A user is a requester who wants to access a resource. A capability token is generated by the owner when receiving the user's request, and then is stored in the smart contract as a new token. The cloud service provider decides on the user's service request to provide access, or not, based on the token retrieved from the blockchain. Furthermore, this model permits only managers to do capability revocation. Although this framework provides distributed capability-based access control with scalability and light weight for the IoT, it is not deployed in real-world IoT applications as well, as it has a latency in its block validation process inherited from the utilized Ethereum platform.

Fotiou et al. [44] proposed an IoT access management solution based on blockchains. In this model, the clients cannot interact directly with IoT devices or gateways, but can interact over the blockchain. A client needs to access IoT devices must have at least one token, while the increasing number of tokens that have clients means more access rights. In addition, the smart contract checks whether ot not the clients have the required number of tokens to invoke it, while the IoT gateways check the user's role at the time, as well as resource location based on gateway access policies. The RPC servers allow clients and IoT devices to be aware of the events generated by a smart contract. This framework is implemented based on the Ethereum blockchain.

Table 4 shows a comparison of token-based access control approaches for the IoT. The approaches are compared in terms of access control model, blockchain type, transaction model, and other main features. These approaches employ tokenization access control to grant or revoke access to IoT resources by the requesters, based on the owner's policy. However, these tokenization approaches are not automatically executing policies of access control to enforce it. In addition, some of these frameworks can authorize through tokenization only, and terminating a token or requesting new access must communicate that to the owner, which causes latency. To overcome transaction latency for tokenization access control approaches in IoT systems, the policies of the access control may be built on a smart contract, which automatically executes the policies to control access rights and detect unwanted activities.

Table 4. A Comparison Table of Token-Based Access Control Approaches.

Reference	Access Model	Blockchain Type	Transaction-Model	Main Features Utilized
FairAccess [36]	Attribute-based AC	Public	Token/Smart contract	- It uses Wallets to broadcast a user's transaction in the P2P network. -Pseudonymous technique is used to ensure the privacy of the users.
IoTChain [37]	ACE Authorization model	Private	Token/Smart contract	-It uses the ACE model for authentication. - It utilizes the OSCAR model for end-to-end security.
Outchakoucht et al. [40]	General-AC	Public	Token/Smart contract	- Reinforcement learning algorithm gives dynamic and auto-re-adjust policies for access control.
Xue et al. [41]	General-AC	Private	Token	- Access policies stores in the blockchain header as policy file that allows the owner to check access privileges of the visitors.
Maesa et al. [42]	Attribute-based AC	Public	Token	-It is used for the implementation by XACML policy model. -It utilizes Policy Administration Point (PAP) and Policy Enforcement Point (PEP) to enforce AC policies with blockchain.
BlendCAC [43]	Capability-based AC	Private	Token/Smart contract	- It used a JSON model for implementing tokens and certificates.
Fotiou et al. [44]	Role-based AC	Public	Token/Smart contract	-It used ERC20 standard based on Ethereum to implement token-based access control.

Policy-Header-Based Access Control for IoT Based on Blockchain

In this mechanism, the access control policies are embedded into the policy header of the local blockchain, which allows the owner of the network to manage all incoming and outgoing transactions in the local blockchain network. Patil et al. [45] proposed a secured lightweight system for smart greenhouse farms based on blockchain technology. In this solution, private local blockchains have policy headers that are responsible for controlling all sending and receiving transactions. In addition, this solution groups IoT nodes into a cluster that can elect any cluster head. The cluster heads control the overlay network. In order to access IoT devices, all transactions coming from the overlay network can be validated and authorized by miners of the local blockchain, because its policy header has an access management list defined by the owner. To overcome the high-cost computation of the PoW mechanism, they used a policy header, which continuously updates its policy, as well as transferring these policies to the newly generated blocks. Although this solution has presented strong security mechanisms to achieve confidentiality, integrity and availability, it is not executing its access control policies automatically.

Dorri et al. [46–48] proposed a lightweight blockchain system to secure IoT; this solution is exemplified by a smart home, which has three layers of cloud storage, an overlay network, and smart home. In this system, the access control policies are stored in the policy header of a private (local) blockchain, to ensure the authorization of the devices, as well as to enforce the owner's resource policy. In order to secure communications inside or outside the home, every home has a powerful machine called a miner or controller that maintains a private blockchain to provide monitoring and access management to the IoT. In this solution, all constrained devices are inside the smart-home tier, which is centrally controlled by the miner (controller). Therefore, there is no need for a PoW consensus to validate blocks, while devices of other smart homes in the network receive broadcasted keys to perform transactions as long as the key is valid. On the other hand, to stop granted access rights, the controller invalidates the shared key and notifies the devices. This framework has various types of transactions, such as access, storage, and monitoring, which handle various operations in the system. The overlay network is like a peer-to-peer network that consists of smart home, cloud storage, and user's devices. In this network, the nodes are grouped into clusters and every cluster elects a cluster head (CH) to manage the blockchain. In order to store data in the cloud, the IoT device sends a request to the controller, and then the controller checks if requester has been granted access permission. Therefore, the controller creates a store transaction for an authenticated requestor, and then sends data to the cloud for storage. However, in this mechanism, access control policies are not self-enforced.

Smart Contract-Based Access Control for IoT Based on Blockchain

A smart contract is a set of executable codes that automatically enforces agreements or terms and conditions between parties. The main benefit of utilizing a smart contract to access management in IoT systems is that the policies of AC are enforced automatically by the smart contract, as well as offering high computing capability to reach numerous access management methods.

Novo [49] is a fully decentralized access management program in IoT-based blockchain technology. This solution excludes IoT devices from the blockchain to overcome network overheads. In addition, this framework has managers, management hubs, and agent node entities, while the managers are responsible for defining new policies in the architecture, as well as interacting with the smart contract to decide to register or de-register. Moreover, the management hubs can request permission on behalf of the IoT devices from the blockchain, by utilizing the call method. The agent node is responsible for deploying smart contracts in the framework. This framework utilized a single smart contract to cover all processes. When a request is received from IoT devices, the gateway nodes or management hubs send an access request to the blockchain network on behalf of the constrained devices and then, the smart contract checks and broadcasts it on the blockchain network for approval. Lastly, IoT devices receive permission or rejection to access the resource. Although this proposed architecture gains scalability due to distributing query permissions through management hubs, the proposed architecture will face security issues if the manager is malicious.

In terms of using various smart contracts to create an IoT access control, Zhang et al. [50] introduced an IoT access control solution based on the smart contract. The framework contains three different contracts. In multiple access control contract (ACC), every point of this ACC can give a single function of access control to the subject-object pair. Judge contract (JC) is a detector and judge of misbehavior of ACC subjects. Register contract (RC) is responsible for maintaining ACC, as well as functions of misbehavior penalties. Furthermore, IoT gateways represent its IoT devices to request a resource, as well as enforcing AC policies that are well-defined in the smart contract. The access request will be granted if required conditions are met and there is no detected misbehavior. Nonetheless, as it is not deployed in the real world, IoT systems were not stress-tested to evaluate their feasibility.

Huang et al. [51] presented a secure IoT data transferring approach based on the blockchain. The framework has a management layer that consists of three different contracts. Access contract is responsible for providing trusted access control to the data based on capability access control. Communication contracts are responsible for traceability and storing all transactions for data transfers in the IoT. Auto exchange contracts are responsible for sending permission rights to the requester after authorization. In this platform, the requester or data demander sends an access request to the smart contract. If the requester meets the required conditions, the smart contract automatically executes and grants access; otherwise, the request will be denied. However, this framework has latency because of the use of the Ethereum platform.

Huh et al. [52] proposed an IoT device management solution utilizing the Ethereum platform. They used a smart contract to control the constrained devices, while using a public key system on the smart contract for authentication. In this system, they used a meter contract for storing electricity, while the policy contract defines the policies of IoT devices. Moreover, the IoT devices are responsible for changing their normal mode to an energy-saving mode if they meet a predefined specified policy. Authenticating the user requires matching with his public key, while all users' public keys are stored on the smart contract. So, if an unauthenticated user tries to access data, it would be detected and rejected easily. However, there are some issues, such as the used platform not supporting light clients and suffering transaction latency.

Using the smart contracts with multi-layered blockchains for IoT access management has the benefit of separating the responsibilities of constrained devices to overcome the public visibility of the blockchain.

Ali et al. [53] developed a decentralized access control mechanism for IoT-based blockchains. This solution is a multi-layered blockchain architecture. In order to access resources, a requester must authenticate by consortium blockchain to allow or deny accessing the sidechain's resources. On the other hand, the sidechain is responsible for keeping track of the data events created by IoT devices. In this framework, they used the smart contract concept to self-enforce policies of access control on consortium layer as well as the sidechain, and it also utilized peer-to-peer IPFS storage to overcome the issue of centralized authority over IoT data handling. However, it did not stress-test various cases to evaluate feasibility, and it has a transaction latency due to the platform used.

Jiang et al. [54] proposed a cross-chain architecture based on multilayer blockchains for decentralized access IoT management. The framework utilized a consortium blockchain to secure access to the whole system, while sidechains (Tangle) record IoT data creation events. In order to access the resource of IoT clusters, the user's public key must first be stored in the smart contract of the consortium network, then signing its private key to be verified. The requester can get an encrypted file if authenticated successfully; otherwise, the request will be denied. The authors implemented access management policies on smart contract of the consortium blockchain, while smart contracts of the sub-tangles grantees to store IoT data creation events securely, as well as allowing communication only with authenticated devices. However, the use of the Byzantine fault tolerance (BFT) consensus mechanism causes some issues like network overhead. In addition, this solution does not address privacy protection for the users.

Jo et al. [55] presented a structural health monitoring system for IoT, based on a hybrid blockchain. This framework consists of two networks; the core network is responsible for mining nodes as well as utilizing a PoW consensus algorithm for verification, while the nodes of the edge network are responsible for access management and identification of the participants. In this solution, any participants with a key and signature can join the network. In addition, the terms and policies are implemented in the smart contract to self-execute in order to make decisions based on a predefined specific value that enables managing reactions or creating alerts. However, there is latency in the block mining process, due to the used consensus mechanism.

Smart contract-based access control in IoT would provide security and reliability in terms of accessing resources, as well as controlling and storing all transactions in the IoT systems, and it provides self-enforcing policies in the access control system. Table 5 summarizes a comparison of the smart contract-based access controls approaches to IoT. The approaches are compared in terms of access control model, consensus mechanism, blockchain platform, and other main features. In addition, using a blockchain platform and consensus mechanisms can affect the IoT system performance. For example, PoW is not suitable for the limited resources of IoT devices, due to its low transaction and high computation cost. The smart contract-based AC approaches mostly achieve their aim of protecting IoT data and devices, but still may have some drawbacks, such as transaction latency, scalability, and privacy issues.

Table 5. A Comparison Table of Smart Contract-Based Access Control Approaches.

Reference	Access Model	Consensus Mechanism	Blockchain Platform	Main Features Utilized
Novo [49]	General AC	PoW	Ethereum	- Excluded IoT devices from blockchain network. - Manager enables to register IoT devices and verify it.
Zhang et al. [50]	Attribute-based AC	PoW	Ethereum	- Used three different smart contracts: ACCs, JC and RC. - Access-control method based on a pair (subject, object).
Jo et al. [55]	General-AC	PoW	Ethereum	- Used two-tiered blockchain architecture. - Used two types of smart contracts main-contract and sub-contract.
Jiang et al. [54]	General-AC	BFT	Hyperledger fabric and IOTA Tangle	- Used consortium blockchains as main controller part and Multiple blockchain for IoT clusters. - Off-chain IPFS for P2P storage.
Ali et al. [53]	General-AC	PoW and Tendermint	Ethereum and Monax	-Multi-tiered blockchain architecture. -Distributed IPFS storage.
Huh et al. [52]	General-AC	PoW	Ethereum	-RSA cryptographic system for generating keys. - Used Ethereum account as constrained devices.
Huang et al. [51]	Capability-based AC	PoW	Ethereum	- Data Access Ticket (DAT) allows user to access data.

4. Discussion and Open Issues

In this work, we covered different access management for IoT. The existing access control mechanisms for IoT are based on central authority management, which is responsible for security issues. This mechanism easily handles access policies and authentication processes because of its reliance on a centralized management server that is kept for all access control policies. On the other hand, the centralized approach to IoT access control has some drawbacks, such as a single point of failure, due to storing all policies of the AC in centralized storage, and the central administrator can alter all operations between users, IoT devices, and service providers, which raises security and privacy issues. In addition, this mechanism is not user-centric, because they cannot control or share their own data. Moreover, centralized access control mechanisms cannot work with increasingly constrained devices, because they do not scale well.

Fortunately, the blockchain features of decentralization and immutability contribute to IoT's access control to overcome the drawbacks of centralized AC. Furthermore, the smart contract's automatic execution of the AC policies can remove the need for third parties. Recently, researchers have developed access control mechanisms for IoT based on blockchains in order to grant or deny access. In addition, to ensure the access right of requesters, they wrote access control policies, either in tokens as blockchain transactions, smart contracts, or the block header as a policy file. Although the blockchain features contributed to IoT access management to overcome some challenges, they also introduced other challenges. A public blockchain requires high computational power for block mining processes where the IoT devices are constrained resources, and all participants in the network can see the transactions, which raises privacy issues. On the other hand, in a private blockchain, all participants are known and restrict users within the same organization, which increases privacy. However, it is not a fully decentralized network, so it cannot ensure accountability. Consequently, this is a guide for choosing a suitable blockchain platform for your IoT application. The most open issues in this work are:

- Privacy leakage: Blockchain technologies contribute to AC of IoT systems to protect data privacy by utilizing digital signatures and the encryption of stored data. However, in a majority of the proposed works, user information can be analyzed on the blockchain network, which raises user privacy issues. Hence, effort is required to integrate privacy preservation techniques such as anonymization or mixing with blockchain-based AC for IoT systems.
- Limited resources of IoT: The mining process poses a key challenge to IoT devices because of their limited storage and power, although the off-chain option that outsources data from the blockchain can decrease latency and improve performance. However, it can have security and privacy issues, so future research will focus on low latency and low computation consensus algorithms that can be adapted to the IoT environment.
- Lack of access control enforcement in the IoT environment due to the distributed architecture of IoT.

Finally, Table 6 shows a comparison table of access control approaches for IoT. The approaches are compared in terms of scalability, distribution, user-centric, privacy protection, self-enforcing policies, and security. However, these features of scalability, dynamicity, and interaction among users are the main requirements of any AC mechanisms for IoT. Furthermore, a lightweight feature is also required to develop any IoT access control mechanism due to constrained devices of the IoT, which cannot handle high computation processes. Therefore, maintaining the balance between auditability and privacy in blockchain-based access control is a challenging task. Thus, future works of blockchain-based access control should focus more on security, user privacy, self-enforcing policies, and lightweight access control for IoT.

Table 6. Comparison Table of Access Control Approaches for IoT. ✓: fully achieved, ❋: partially achieved, ✗: not achieved.

	Scalability	Distribution	User-Centric	Data-Privacy	User-Privacy	Self-Enforcing-Policies	Security
DAC [2,3]	✗	✗	✗	✗	✗	✗	❋
MAC [4,5]	✗	✗	✗	✗	✗	✗	✓
RBAC [6,7]	✗	✗	✗	✗	✗	✗	✓
SmartOrBAC [30,31]	✗	✓	✗	✗	✗	✗	✓
CapBAC [34,35]	❋	✓	✓	✗	✗	✗	❋
FairAccess [36]	❋	✓	✓	✓	✓	✓	✓
IoTchain [37]	✓	✓	✓	✓	✗	✓	✓
Outchakoucht et al. [40]	❋	✓	✓	✓	✗	✓	✓
Xue et al. [41]	❋	✓	✓	✓	✗	✗	✓
Maesa et al. [42]	❋	✓	✓	✗	✗	✗	❋
BlendCAC [43]	✓	✓	✓	✓	✗	✓	✓
Fotiou et al. [44]	✓	✓	✓	✓	✗	✓	✓
Novo [49]	✓	✓	✓	✓	✗	✓	✓
Zhang et al. [50]	✓	✓	✓	✓	✗	✓	✓
Huang et al. [51]	✓	✓	✓	✓	✗	✓	✓
Huh et al. [52]	✓	✓	✓	✓	✗	✓	✓
Ali et al. [53]	✓	✓	✓	✓	✗	✓	✓
Jiang et al. [54]	✓	✓	✓	✓	✗	✓	✓
Jo et al. [55]	✓	✓	✓	✓	✗	✓	✓
Patil et al. [45]	✓	✓	✓	✓	✗	✗	✓
Dorri et al. [46–48]	✓	✓	✓	✓	✗	✗	✓

5. Conclusions

With the growing number of IoT devices, control by a central authority raises security and privacy issues. In this paper, we briefly reviewed IoT paradigms and the blockchain concept and provided a classification of access control for IoT into two main parts, centralized and decentralized access control, each part of which has subcategories. A comparison table of the discussed access control mechanisms is provided. Furthermore, current classical access control mechanisms for IoT systems are under single authority management, which can cause a single point of failure or privacy leaks.

In addition, we mentioned scalability, user-centric, privacy, and security as well as the self-enforcing policies of each mechanism. Blockchain features such as decentralization, tamper-proof, and smart contract can help resolve IoT security issues. However, blockchain-based access control mechanisms raise privacy issues that need to be integrated through privacy-preserving techniques with blockchain-based access control. Finally, we mentioned some open issues and future directions for the study of security and privacy in IoT access control mechanisms.

Author Contributions: Conceptualization, A.I.A.; investigation, A.I.A.; validation, F.E.E., K.J., K.A. and A.S.A.-M.A.-G.; writing—original draft preparation, A.I.A.; writing—review and editing, A.I.A.; supervision, F.E.E. and K.J.; project administration, F.E.E. and K.J.; funding acquisition, A.S.A.-M.A.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (RG-18-611-38). The authors, therefore, gratefully acknowledge the DSR technical and financial support.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AC	Access Control
IoT	Internet of Things
DAC	Discretionary Access Control
MAC	Mandatory Access Control
RBAC	Role-based Access Control
tps	transaction per second
PoS	Proof of Stake
PoW	Proof of Work
PBFT	Practical Byzantine Fault Tolerance
RPCA	Ripple Protocol Consensus Algorithm
BFT	Byzantine Fault Tolerance
OrBAC	Organization-based Access Control
CapBAC	Capability-based Access Control
OSCAR	Object Security Architecture
ACE	Authentication and Authorization for Constrained Environments
RPC	remote procedure call

References

1. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A survey on IoT security: Application areas, security threats, and solution architectures. *IEEE Access* **2019**, *7*, 82721–82743. [CrossRef]
2. Downs, D.D.; Rub, J.R.; Kung, K.C.; Jordan, C.S. Issues in Discretionary Access Control. In Proceedings of the 1985 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 22–24 April 1985; p. 208. [CrossRef]
3. Jayant, D.B.; Swapnaja, A.U.; Sulabha, S.A.; Dattatray, G.M. Analysis of DAC MAC RBAC Access Control based Models for Security. *Int. J. Comput. Appl.* **2014**, *104*, 6–13. [CrossRef]
4. Bell, D.E.; La Padula, L.J. *Secure Computer System: Unified Exposition and Multics Interpretation*; (No. MTR-2997-REV-1); MITRE Corp: Bedford, MA, USA, 1976.
5. Denning, D.E. A lattice model of secure information flow. *Commun. ACM* **1976**, *19*, 236–243. [CrossRef]
6. Sandhu, R.S. Role-based access control. In *Advances in Computers*; Elsevier: Fairfax, VA, USA, 1998; Volume 46, pp. 237–286, ISBN 0-12-012146-8.
7. Sandhu, R.; Bhamidipati, V.; Coyne, E.; Ganta, S.; Youman, C. The ARBAC97 model for role-based administration of roles: Preliminary description and outline. In Proceedings of the Second ACM Workshop on Role-Based Access Control, Fairfax, VA, USA, 5–8 November 1997; pp. 41–50.
8. Pearsall, K. Manufacturing supply chain challenges-globalization and IOT. In Proceedings of the 2016 6th Electronic System-Integration Technology Conference (ESTC), Grenoble, France, 13–15 September 2016; pp. 1–5.
9. Huckle, S.; Bhattacharya, R.; White, M.; Beloff, N. Internet of things, blockchain and shared economy applications. *Procedia Comput. Sci.* **2016**, *98*, 461–466. [CrossRef]
10. Ouaddah, A.; Mousannif, H.; Abou Elkalam, A.; Ouahman, A.A. Access control in the Internet of Things: Big challenges and new opportunities. *Comput. Netw.* **2017**, *112*, 237–262. [CrossRef]
11. Nakamoto, S.; Bitcoin, A. A peer-to-peer electronic cash system. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 1 April 2020).
12. Elrom, E. Blockchain Nodes. In *The Blockchain Developer: A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-based Projects*; Apress: Berkeley, CA, USA, 2019; ISBN 978-1-4842-4846-1; 978-1-4842-4847-8.
13. Tschorsch, F.; Scheuermann, B. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2084–2123. [CrossRef]
14. Xu, X.; Weber, I.; Staples, M.; Zhu, L.; Bosch, J.; Bass, L.; Pautasso, C.; Rimba, P. A taxonomy of blockchain-based systems for architecture design. In Proceedings of the 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, Sweden, 3–7 April 2017; pp. 243–252.
15. Hassan, M.U.; Rehmani, M.H.; Chen, J. Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions. *Future Gener. Comput. Syst.* **2019**, *97*, 512–529. [CrossRef]

16. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
17. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference ACM, Porto, Portugal, 23–26 April 2018; pp. 1–15.
18. Hyperledger Burrow—Hyperledger. Available online: https://www.hyperledger.org/projects/hyperledger-burrow (accessed on 13 April 2020).
19. Pilkington, M. Blockchain technology: Principles and applications. In *Research Handbook on Digital Transformations*; Edward Elgar Publishing: Cheltenham, UK, 2016.
20. Castro, M.; Liskov, B. Practical Byzantine fault tolerance. *Oper. Syst. Des. Implement. (OSDI)* **1999**, *99*, 173–186.
21. Chase, B.; MacBrough, E. Analysis of the XRP ledger consensus protocol. *arXiv* **2018**, arXiv:1802.07242.
22. Gu, J.; Sun, B.; Du, X.; Wang, J.; Zhuang, Y.; Wang, Z. Consortium blockchain-based malware detection in mobile devices. *IEEE Access* **2018**, *6*, 12118–12128. [CrossRef]
23. Vukolić, M. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In *International Workshop on Open Problems in Network Security*; Camenisch, J., Kesdoğan, D., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 112–125.
24. Szabo, N. Formalizing and securing relationships on public networks. *First Monday* **1997**, *2*. [CrossRef]
25. Hyperledger—Open Source Blockchain Technologies. Available online: https://www.hyperledger.org/ (accessed on 30 March 2020).
26. Macrinici, D.; Cartofeanu, C.; Gao, S. Smart contract applications within blockchain technology: A systematic mapping study. *Telemat. Inform.* **2018**, *35*, 2337–2354. [CrossRef]
27. Schroeder, M.D. Cooperation of Mutually Suspicious Subsystems in a Computer Utility. (No. MAC-TR-104); Doctoral dissertation, Dept. Electrical Eng., Massachusetts Institute of Technology, Cambridge, MA, USA, 1972.
28. Qian, X.; Lunt, T.F. A MAC policy framework for multilevel relational databases. *IEEE Trans. Knowl. Data Eng.* **1996**, *8*, 3–15. [CrossRef]
29. Sandhu, R.; Munawer, Q. The ARBAC99 model for administration of roles. In Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC'99), Phoenix, AZ, USA, 6–10 December 1999; pp. 229–238. [CrossRef]
30. Ouaddah, A.; Bouij-Pasquier, I.; Abou Elkalam, A.; Ouahman, A.A. Security analysis and proposal of new access control model in the Internet of Thing. In Proceedings of the 2015 International Conference on Electrical and Information Technologies (ICEIT), Marrakesh, MOROCCO, 25–27 March 2015; pp. 30–35. [CrossRef]
31. Bouij-Pasquier, I.; Abou El Kalam, A.; Ouahman, A.A.; De Montfort, M. A security framework for internet of things. In *Cryptology and Network Security*; Reiter, M., Hill, C., Eds.; Springer: Cham, Switzerland, 2015; pp. 19–31.
32. Kalam, A.A.E.; Baida, R.E.; Balbiani, P.; Benferhat, S.; Cuppens, F.; Deswarte, Y.; Miege, A.; Saurel, C.; Trouessin, G. Organization based access control. In Proceedings of the POLICY 2003, IEEE 4th International Workshop on Policies for Distributed Systems and Networks, Lake Como, Italy, 4–6 June 2003; pp. 120–131. [CrossRef]
33. Dennis, J.B.; Van Horn, E.C. Programming semantics for multiprogrammed computations. *Commun. ACM* **1966**, *9*, 143–155. [CrossRef]
34. Hernández-Ramos, J.L.; Jara, A.J.; Marin, L.; Skarmeta, A.F. Distributed capability-based access control for the internet of things. *J. Internet Serv. Inf. Secur. (JISIS)* **2013**, *3*, 1–16. [CrossRef]
35. Gusmeroli, S.; Piccione, S.; Rotondi, D. A capability-based security approach to manage access control in the internet of things. *Math. Comput. Model.* **2013**, *58*, 1189–1205. [CrossRef]
36. Ouaddah, A.; Abou Elkalam, A; Ait Ouahman, A. FairAccess: A new Blockchain-based access control framework for the Internet of Things. *Secur. Commun. Netw.* **2016**, *9*, 5943–5964. [CrossRef]

37. Alphand, O.; Amoretti, M.; Claeys, T.; Dall'Asta, S.; Duda, A.; Ferrari, G.; Rousseau, F.; Tourancheau, B.; Veltri, L.; Zanichelli, F. IoTChain: A blockchain security architecture for the Internet of Things. In Proceedings of the 2018 IEEE wireless communications and networking conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6.
38. Vučinić, M.; Tourancheau, B.; Rousseau, F.; Duda, A.; Damon, L.; Guizzetti, R. OSCAR: Object security architecture for the Internet of Things. *Ad Hoc Netw.* **2015**, *32*, 3–16. [CrossRef]
39. Seitz, L.; Selander, G.; Wahlstroem, E.; Erdtman, S.; Tschofenig, H. Authentication and Authorization for Constrained Environments (ACE). Internet Engineering Task Force, Internet-Draft draft-ietf-aceoauth-authz-07. 2017. Available online: https://datatracker.ietf.org/doc/html/draft-ietf-ace-oauth-authz-07 (accessed on 19 April 2020).
40. Outchakoucht, A.; Hamza, E.S.; Leroy, J.P. Dynamic access control policy based on blockchain and machine learning for the internet of things. *Int. J. Adv. Comput. Sci. Appl* **2017**, *8*, 417–424. [CrossRef]
41. Xue, J.; Xu, C.; Zhang, Y. Private Blockchain-Based Secure Access Control for Smart Home Systems. *KSII Trans. Internet Inf. Syst.* **2018**, *12*, 6057–6078.
42. Maesa, D.D.F.; Mori, P.; Ricci, L. Blockchain based access control. In *Distributed Applications and Interoperable Systems*; Chen, L.Y., Reiser, H.P., Eds.; Springer: Cham, Switzerland, 2017; pp. 206–220.
43. Xu, R.; Chen, Y.; Blasch, E.; Chen, G. BlendCAC: A smart contract enabled decentralized capability-based access control mechanism for the IoT. *Computers* **2018**, *7*, 39. [CrossRef]
44. Fotiou, N.; Pittaras, I.; Siris, V.A.; Voulgaris, S.; Polyzos, G.C. Secure IoT access at scale using blockchains and smart contracts. In Proceedings of the 2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Washington, DC, USA, 10–12 June 2019; pp. 1–6.
45. Patil, A.S.; Tama, B.A.; Park, Y.; Rhee, K.H. A framework for blockchain based secure smart green house farming. In *Advances in Computer Science and Ubiquitous Computing* ; Loia, V., Sung, Y., Eds.; Springer: Singapore, 2017; pp. 1162–1167.
46. Dorri, A.; Kanhere, S.S.; Jurdak, R. Blockchain in internet of things: challenges and solutions. *arXiv* **2016**, arXiv:1608.05187.
47. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. Blockchain for IoT security and privacy: The case study of a smart home. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13–17 March 2017; pp. 618–623.
48. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. A Lightweight Scalable Blockchain for IoT security and anonymity. *J. Parallel Distrib. Comput.* **2019**, *134*, 180–197. [CrossRef]
49. Novo, O. Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet Things J.* **2018**, *5*, 1184–1195. [CrossRef]
50. Zhang, Y.; Kasahara, S.; Shen, Y.; Jiang, X.; Wan, J. Smart contract-based access control for the internet of things. *IEEE Internet Things J.* **2018**, *6*, 1594–1605. [CrossRef]
51. Huang, Z.; Su, X.; Zhang, Y.; Shi, C.; Zhang, H.; Xie, L. A decentralized solution for IoT data trusted exchange based-on blockchain. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017.
52. Huh, S.; Cho, S.; Kim, S. Managing IoT devices using blockchain platform. In Proceedings of the 2017 19th International Conference on Advanced Communication Technology (ICACT), Bongpyeong, Korea, 19–22 February 2017; pp. 464–467.
53. Ali, M.S.; Dolui, K.; Antonelli, F. IoT data privacy via blockchains and IPFS. In Proceedings of the Seventh International Conference on the Internet of Things ; ACM Press: Linz, Austria, October 2017; pp. 1–7.
54. Jiang, Y.; Wang, C.; Wang, Y.; Gao, L. A cross-chain solution to integrating multiple blockchains for IoT data management. *Sensors* **2019**, *19*, 2042. [CrossRef]
55. Jo, B.W.; Khan, R.M.A.; Lee, Y.S. Hybrid blockchain and internet-of-things network for underground structure health monitoring. *Sensors* **2018**, *18*, 4268. [CrossRef]