# ControlChain: Blockchain as a Central Enabler for Access Control Authorizations in the IoT

Otto Julio Ahlert Pinno, André Ricardo Abed Grégio, Luis C. E. De Bona
Computer Science Department, Federal University of Paraná, Curitiba - PR, Brazil
{ojapsilva, gregio, bona}@inf.ufpr.br

***Abstract*—The IoT is pervading our daily activities and lives with devices scattered all over our cities, transport systems, buildings, homes and bodies. This invasion of devices with sensors and communication capabilities brings big concerns, mainly about the privacy and confidentiality of the collected information. These concerns hinder the wide adoption of the IoT. To overcome them, in this work, we present an Blockchain-based architecture for IoT access authorizations. Following the IoT tendency requirements, our architecture is user transparent, user friendly, fully decentralized, scalable, fault tolerant and compatible with a wide range of today's access control models used in the IoT. Finally, our architecture also has a secure way to establish relationships between users, devices and group of both, allowing the assignment of attributes for these relationships and their use in the access control authorization.**

## I. Introduction

The Internet of Things (IoT) emerged with the objective of providing new intelligent services and commodities to facilitate our daily tasks. Its devices are pervading our cities, public buildings, roads, airways, factories, retail stores, offices, hospitals, homes and bodies [1]. With their sensors, communication and information processing capabilities they affect our interactions on all applications domains: personal, home, government, utilities, enterprise and industry [2].

Together with the great features that arise with such integrated systems, there are many security concerns that hinders its broad adoption by users, governments and industries. One of the major concerns is about the control of the devices and data handled by them [2]. Recently, more than 150,000 IoT devices were compromised and the investigations identified the access control as the main responsible for the security breach [3]. Therefore, the adoption of improper access control systems could cause big privacy and economical harm to individuals and enterprises.

A complete access control solution involves three components [4]: authentication, authorization and auditing. The authentication identifies the correct identity of the subject. The authorization verifies if the subject has the rights to do some operation on the object. Finally, the auditing (or accountability) allow the posterior analysis of the realized activities in the system. These components have important roles in securing the system, however the authorization component requires a special attention because it is responsible for enforcing the access rules.

Some of the works in the access authorization field employ three traditional and well known architectures: XACML, OAuth and UMA. However, all these three architectures fail to provide essential IoT access control characteristics, like transparent authorization process for the user, scalability and resilience to wireless intermittent communications. Therefore, even with a lot of effort to bring a suitable access control to the IoT [2], there are design barriers in the traditional architectures that prevent them to achieve a complete success.

In a search for more suitable solutions, some works [1] started using a disruptive technology for the access control, namely Blockchain. The Blockchain is a public, decentralized, Byzantine fault-tolerant and immutable ledger, where registers are appended in a chronological order [5]. It was already employed in a plenty of areas [6], like cryptocurrencies, transportation systems, management of medical records, decentralization of the Web, predictions and applications platforms. The main advantages of the Blockchain are no downtime, no censorship, no fraud and no third-party interference.

FairAccess [1], [7] is an access control framework based on Blockchain. In this framework, the Blockchain is used to distribute access tokens using smart contracts. However, their proposal has some issues, like the support to token-based authorizations only, necessity of contact with the owner of the resource for each new access or each token expiration, the high time cost involved in getting an access permission and the lack of integration of the access control with a proper relationship network that has a big importance in a collaborative and integrated IoT.

In this work, we present a new architecture for access control that are heavily based on the utilization of the Blockchain technology. Our architecture overcome the FairAccess and traditional architectures problems with a complete decentralized and transparent authorization process that is compatible with a wide variety of access control models employed on IoT (requiring minor adaptation efforts) and, finally, our proposal also includes a new method to declare relationship between entities, i.e. users and devices.

The remaining of this work is divided as follows. Section II present some knowledges utilized in the rest of the work. Section III shows the related works. Section IV describe our proposed architecture. Section V present methodologies to adapt known models to our architecture. Section VI compare our architecture with other ones and discuss the viability of the application of the Blockchain in limited resources devices. Finally, Section VII concludes this work.

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

TABLE I
ADOPTED NOMENCLATURE

| Nomenclature | Meaning |
|---|---|
| Subject | The entity trying to access a resource |
| Action | The activity over an object |
| Object | A generic resource that can be accessed |
| Context | A set of unambiguous variables in the system |
| Context identifier | A pair (entity, variable), where the variable is entity-dependent |
| Miner | Device that append blocks to the Blockchain |
| Identity | A value that uniquely identifies an entity |
| Entity | A subject, object or a set of them |

## II. BACKGROUND

In this section, we present some fundamental concepts used on our proposal. First, we define our used nomenclature in Table I, after we brief describe some of the traditional architecture in Section II-A and, after, we discuss about some of the Blockchain aspects in Section II-B.

### A. Traditional architectures

There are 3 main traditional architectures commonly employed in the IoT access control: XACML [8], OAuth [9] and UMA [10]. Following, we give a brief description of the main characteristics and working manner of them.

**XACML.** The XACML is a standard that includes a declarative fine-grained, attribute-based access control policy language, an architecture and a processing model to evaluate requests. The XACML architecture has many components, however, the Police Enforcement Point (PEP) and Police Decision Point (PDP) can be considered the main ones. The PEP is responsible for enforcing the application of the policies. To do so, it intercepts the request and send it to the PDP evaluate the request. After receiving the evaluation of the PDP, the PEP decides to allow or deny the access.

**OAuth.** The OAuth is a protocol that allow a token-based authorization. Basically, the third party wanting access to a resource asks to the resource owner for a permission to access it. The owner authorizes it, allowing the emission of a token to the third party. Then, the third party can access the resource presenting the token.

**UMA.** The UMA is an OAuth-based protocol that unifies, in the perspective of the resources owner, the control point of resource utilization authorizations. The UMA is composed by 4 types of entities: Resource Owner (RO), Resource Server (RS), Authentication Server (AS) and Client. The RO provides the resource in the RS and control the access to this resource in the AS. The client requests tokens to AS and uses it to access the resources in the RS.

Although these architectures have been employed in a wide range of works [2], none of them bring solutions to the problem caused by the architecture components centralization. This dependency of a central entity make them susceptible to the single point of failure problem and could limit the growing of the IoT. Therefore, their utilization leads to a less fault resilient and less scalable IoT.

### B. Blockchain

With its origin in the cryptocurrencies, more specifically in the Bitcoin [11], the Blockchain was designed to serve as a secure distributed database for the storage of all the digital assets transactions ever made in the platform. This database is, in fact, a public, decentralized, Byzantine fault-tolerant and immutable ledger, where registers are appended in a chronological order.

The Blockchain relies on the existence of special workers, namely miners. The miners are responsible for the append of historic transactions. To do so, they create groups of valid transactions, called blocks and append them in the Blockchain. To avoid attacks, like the double-spending problem[1], and protect the information on the Blockchain the miners must respect some security mechanisms. For example, Bitcoin only consider a new block as a valid one if it has the last block hash inside the new block and the miner discover a nonce that cause the hash of this new block to be below or equal a target value. This nonce discovery is known as proof-of-work. Besides this the Bitcoin also employ a consensus mechanisms to decide between eventual branches that emerge on the Blockchain. The adopted consensus is the choose of the longest chain. With these security measures, it is granted that the older is a history in the Blockchain, the more secure it is from attacks and faults.

Although, the Blockchain was design for Bitcoin transactions, it could be used with any type of information. Some works [7] use it to store applications scripts, namely smart contracts. The use of smart contracts is indicated for scenarios like market trades, register of debts or promises and others [12]. In our work, we use the Blockchain to store access control rules, relationships, contexts and accountability information.

## III. RELATED WORK

Although the division of access control approaches in different layers is not always straightforward, [2] try to standardize this classification based on the OM-AM reference model. They classify the state-of-art in 4 layers, namely Objective, Model, Architecture and Mechanisms. In the objective layer, the access control policy (high level rules) is investigated. The Model layer defines unambiguous means for apply the access control policy. The Architecture layer describes the entities of the access control, their workflow and interactions. The mechanism layer defines the software and hardware used in the access control policy enforcement. Next, we choose to present only works that are somehow related to the architecture layer because our work is more related to this layer.

FairAccess [1], [7] is an access control framework that uses smart contracts and a Blockchain where they are stored. The smart contracts are used to trade fulfillments of access control policies for access tokens. Different from our proposal, the disadvantages of their solution are fourfold. First, each time a token expires or are revoked, the subject needs to contact the owner for the token generation. Second, for a new token

[1] https://en.bitcoin.it/wiki/Double-spending

be usable, it is required at least two blocks to be mined to the Blockchain, i.e. it is costly and could take a lot of time to the access be granted. Third, they only support token based authorizations. Fourth, they don't give any solution for allowing the usage of relationships in the process of granting access to a subject. The only relationship information that could be inferred is: with two identities (public keys) it is possible to know if one is parent of the other in the tree of generated keys. In our proposal, each identity could be linked with any other identity, giving attributes and characteristics to the linked ones.

[13], [14] and [15] also employ Blockchain in the access control. [13] controls the access permissions to medical record data of patients. It employs smart contract between patients, providers and third parties to grant permissions of access. [14] controls the actions performed in a video rights management system. [15] uses the Blockchain to store, query and share data. The data is stored in a off-Blockchain DHT network and only the pointer to this data is stored in the Blockchain. Differently from our architecture, these works only allow creation of ACL-like rules and don't support other information in the authorization process, like the environment context.

IBM Watson IoT [16] is a platform that, between a wide range of services, manage and control the access to IoT devices. It also allows device data to get published into a private blockchain in order to reduce the dependence of a central management entity in the data access. However, all the configuration of the access control is centralized.

## IV. ControlChain

The ControlChain is an architecture to provide access control in IoT. It was created with the following principles in mind: *Decentralization.* The expected grow of the IoT requires a decentralized solution. *Resilience.* Make an architecture with no single points of failure and resilience to data corruption. *Off-line working.* The dominant IoT communication media type, i.e. wireless, is known to be instable and could lead to intermittent connections, so the continuous operation in a disconnected environment is necessary. *Low processor usage for authorizations.* In the IoT, some devices will be restricted in the power processing capacity. Therefore, the ControlChain is a decentralized, resilient, allow off-line work and has low processor usage profile on the authorization process.

The Figure 1 presents an overview of the ControlChain. Although nothing prevents all the information to be in one unique Blockchain, for organization and to facilitate the explanation, we divided the database of the ControlChain in 4 different Blockchains: Context Blockchain, Relationships Blockchain, Rules Blockchain and Accountability Blockchain. It is important to note that, based on the Blockchain principles, once an information is published on one of these Blockchain it is part of the information history and cannot be erased, however, we allow the update of the Blockchain contents with the creation of new registers. Next, we explain how these 4 Blockchains works.

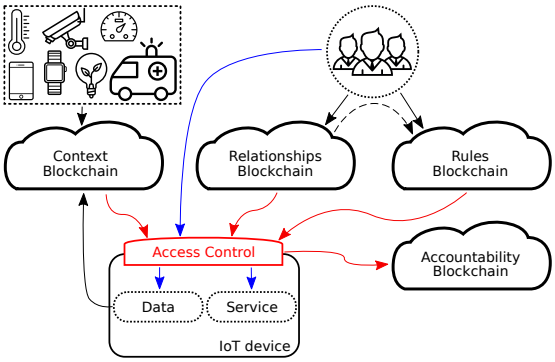


Fig. 1. Architecture overview

**Relationships Blockchain.** The Relationships Blockchain is responsible for the storage of the public credentials and relationships of all entities (See Table I) in the system. In fact, there is no differentiation between users, devices or groups in our proposal. Each entity has an owner that has complete control over it in the system. A relationship is a unilateral reference to another entity with an optional set of attributes to it. The relationships of the entities is important in our architecture because it could be used in the authorization decisions.

There are two possible types of relationship references: Blockchain-dependent and external. The *Blockchain-dependent* reference is a link created with identifications tied to Blockchain registers (for example, chronological number of the block and line number of the register inside the block). The *external* reference is a link based on Blockchain external identification (for example, a public key). It is a dynamic reference that always is interpreted as a pointer to the most recent update of an entity in the Blockchain, if it exists there. Note that it also allows the reference to entities outside the Blockchain. The choose of best type of reference is use case dependent. In Table II we give some directions to the best choice based on the use case requirements. The "+" signal shows the most indicated type for the requirement.

TABLE II
Directions for the choose of the reference type

| | **Identifications Types** | |
|---|---|---|
| | Blockchain-dependent | External |
| If the latest information inside a referenced entity need to be evaluated | | + |
| If the information inside a referenced entity need to be evaluated as it was in the time of the authorization | + | |
| If references to entities outside the Blockchain are allowed | | + |
| If it is only allowed references to entities on the Blockchain | + | |

The Figure 2 shows two examples (numbered as 1 and 2) of Relationships Blockchain. In this figure, each square is a block, the leftmost square is the genesis block, contiguous line arrows are the default links between the blocks, dashed line arrows are relationships, dotted line arrows are relationships that are dependent of the relationship references type chosen

and the hatch represents the block owner. It's important to remainder that one block could have more than one entity and not necessarily the blocks are labeled with their type or function inside the use case, like the user, device or group in the figure. However, to simplify these examples, we represent only one entity per block and label the blocks.

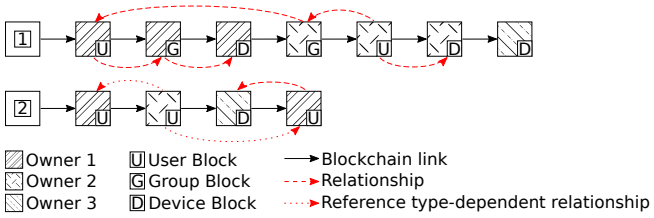


Fig. 2. Relationship overview

In the first example, the left part shows the relationship between an user, a group and a device (all owned by the Owner 1): the user references the group and the group references the device. In the middle of the example, there is a Owner 2 user with a group (for example, with the attribute "friends") that links to the Owner 1 user. Finally, could exist entities without references or been referenced like the last entity of the Blockchain. As can be seen, the type of relationship adopted in this example is the external one because there is entities referencing other after-appended entities.

The second part of the Figure 2 presents an example of a behavior that depend on the chosen reference type. In this example, the Owner 1 user was created on the second block with no relationship and was referenced by the Owner 2 user in the third block. After, the Owner 1 user was updated to reference the Owner 3 device. With the Blockchain-dependent reference type, the Owner 2 user stays referencing the first version of the Owner 1 user, i.e. the one without references. With the external reference type, the Owner 2 user relationship is automatically pointing to the updated Owner 1 user, i.e. the one with the reference to the Owner 3 device.

**Context Blockchain.** The Context Blockchain store contextual information obtained from sensors, processed data and manual inputs. This contextual information can be used in the authorization decision. For example, suppose there is an access rule with the following statement: "8k resolution videos can only be accessed when the router reports that the network traffic is low". In this situation, the access control will find the report of the router in the Context Blockchain and check its state before allowing the access.

**Accountability Blockchain.** The Accountability Blockchain register information about permissions or denies of access to object. The information required to registered is described in the Rules Blockchain. These information could be used for accountability and auditing of accesses, and for checking the sanity of the system. Furthermore, the information stored in this Blockchain could also be used like the contextual ones.

**Rules Blockchain.** The Rules Blockchain keep the authorization rules defined by owners to their objects or by objects to themselves. The big challenge faced by this Blockchain is making it generic enough to be compatible with the big variety of access control models and mechanisms used in the IoT: RBAC [17]–[19], ABAC [20], UCON [21], CapBAC [22]–[25], ACL [13]–[15] and others [26], [27]. Each one capable of fulfill different IoT scenarios requirements. Without loss of generalization, we identified 3 types of access control mechanisms (based on ACL, Capability and Attribute) that with some minor additions could lead to the compatibility with a lot of models. These minor additions are the append of context conditions, obligations and a list of information to be registered on the Accountability Blockchain. The context conditions are Boolean expressions that are build using context identifiers (See Table I) of the Context Blockchain. The obligations determine routines (for example, accept an agreement) that the subject must accomplish to get the access authorization. Finally, the third addition describe the access information that should be recorded on the Accountability Blockchain by the access control. We call these mechanisms based blocks as ACL rule block (allows a list of subjects for each object), capability rule block (allows a list of objects for each subject) and attribute rule block (allow a list of subjects' and objects' attributes).

The process to transform access control models to more generic mechanisms is illustrated in Figure 3. We propose the utilization of a *Decoder*. This Decoder receives the access control model and rules and translate them to mechanisms supported in our architecture. In some cases, the users need to provide additional information. For example, suppose a rule says that only subjects with the role "manager" can have access to the management system. This role can be seen as an attribute of the entity subject, however, at least one entity should be pointed as the official attribute provider for the evaluation of the rule.

## V. Decoder: Model Adaptation for Our Architecture

In this section, we present directions to easily automate the adaptation of IoT authorization models to the three suggested mechanisms based blocks for the Rule Blockchain. Table III presents the mapping of the elements of the RBAC [28], OrBAC [29], ABAC [30], UCON [31] and CapBAC [32] models to these suggested mechanisms. The mapping of ACL mechanism is not presented in the table, however, it is straight forward: subjects and objects are entities and the allowed actions are defined in the ACL Rule Block. Note also that the UCON model requires that the object constantly monitor the Blockchain for changes that violates the rules and obligations of the current accesses and interrupt any access with violated rules and obligations.

## VI. Analysis of ControlChain

This section is divided in two parts. In the first part, we compare ControlChain with other architectures and, in the second part, we discuss the viability of the ControlChain usage in scenarios with limited resource devices.

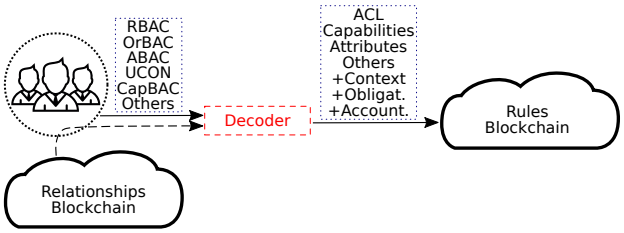


Fig. 3. Transformation of access control models to the mechanisms

TABLE III
MAPPING OF THE MODELS TO THE SUGGESTED RULE BLOCKS

| ControlChain | RBAC | OrBAC | ABAC | UCON | CapBAC |
|---|---|---|---|---|---|
| Entity | Sub. | Sub. | Sub. | Sub. | Sub. |
| Entity att. | Role | Role | Sub. att. | Sub. att. | - |
| Entity | Obj. | Obj. | Obj. | Obj. | Obj. |
| Attr. Rule Block(*) | Rule | Rule | Rule | Auth. | - |
| -(**) | - | Activ. | - | - | - |
| Entity att. | - | View | Obj. att. | Obj. att. | - |
| Entity | - | Org. | - | - | - |
| Context identifiers | - | Cont. | Cont. | Cont. | - |
| Cap. Rule Block | - | - | - | - | Rule |

**Abbreviations:** Attribute (Att.); Capability (Cap.); Subject (Sub.); Object (Obj.); Activity (Activ.); Organization (Org.); Context (Cont.); Authorization (Auth.)
**-** The model/architecture does not directly support this element
**(*)** When applicable, it requires the official attribute provider, subject and object attribute names and expected values, allowed actions, objects with the allowed actions, contexts and obligations
**(**)** The conversion of activities to actions need to be performed by OrBAC subjects and objects

### A. Comparison

The comparison of ControlChain with the XACML, OAuth, UMA and FairAccess is presented in the Table IV. After, the justifications of the evaluations are discussed.

TABLE IV
ARCHITECTURES COMPARISON

| | XACML | OAuth | UMA | FairAccess | ControlChain |
|---|---|---|---|---|---|
| Scalability | - | - | - | + | + |
| Fault tolerant | - | - | - | +- | + |
| No third-parties | - | - | - | + | + |
| New authorization | + | + | + | - | -(*) |
| Get authorization | + | + | + | -(*) | + |
| Integr. relationship | - | - | - | - | + |
| Compatibility | + | - | - | - | + |
| Low object overhead | + | + | + | + | + |

(*) Exclusively dependent of the type of proof and dissemination speed of blocks

**Scalability.** FairAccess and ControlChain were designed with the decentralized principle in mind and, thus, they are more scalable approaches than the others (XACML, OAuth and UMA).

**Fault tolerant.** This criterion evaluates the impact caused by failures on devices or communication links. Naturally, the decentralized ones are more fault tolerant than the centralized ones. However, the FairAccess requires that the resource owner publishes a token every time an access is requested. This gives ControlChain a slight better evaluation.

**No third parties.** The dependence on third parties could prevent the detection of censorships, frauds and interferences. All the centralized architectures depend on third parties and, thus, only FairAccess and ControlChain receives positive evaluations.

**New Authorization.** This criterion evaluates the latency to make or change an authorization. The centralized architectures XACML, OAuth and UMA have low latencies in these activities because the update of their database is straightforward. Thus, they received a positive evaluation. For FairAccess and ControlChain, the complete analysis of this criterion is dependent of the blocks dissemination speed and, mainly, the type of proof used in the blocks mining. As the most common and known type is the proof-of-work and this type of proof imposes a considerable latency, we give a negative evaluation. It is important to note that, the FairAccess has a centralized token issuer (the owner) that is similar to the centralized architectures, but a contract with a token published on the Blockchain cannot be modified.

**Get authorization.** This criterion evaluates the latency to get an authorization. XACML, OAuth, UMA and ControlChain have almost real-time authorizations. The FairAccess has a bigger latency because, after the off-line request to the resource owner, it requires 2 sequential blocks to be mined before the access is granted. Thus, it is the only that receives negative evaluation.

**Integrated relationship.** The ControlChain is the only architecture designed to directly allow relationships on rules. Therefore, it was the only one positively evaluated.

**Compatibility.** This criterion evaluates the compatibility of the architectures with the plenty of models currently employed in the IoT. XACML and ControlChain are compatible with a considerable quantity of models and then they received a positive evaluation. OAuth, UMA and FairAccess can operate with almost any model in their background, however they only operate with access tokens in the foreground and, thus, they were negatively evaluated.

**Low object overhead.** This criterion evaluates how much the object is overloaded by the authorization process. In the centralized architectures, all the authorization process is externalized to a powerful entity by design. However, nothing prevents the FairAccess and the ControlChain to externalize their authorization process too. In fact, the ControlChain can facilitate the automation of this process when it is necessary. A device could search, in the Relationships Blockchain, for other devices of the same owner and ask their support in the authorization process. Therefore, all the architectures were positively evaluated.

### B. Viability with limited resources devices

One important factor about an access control architecture for the IoT is it viability in scenarios with limited resource devices. In this section, we discuss how the main technology used in the ControlChain (the Blockchain) and the evaluation of rules can be compatible with these scenarios.

**Growing of the Blockchain.** The size of an entire Blockchain could be a problem to devices with low storage space. However, devices doesn't need to store the full Blockchain. For example, the storage could be performed

with some replication factor. Beside this, more limited devices could also filter all the non-important information and store only the ones it judges to be important. For example, a device could store only rules related to it and both, contexts and relationships, related to these stored rules.

**Speed of new registers.** With the arise of many new registers in a short period of time, devices with less resources could not be capable of keep up with the updates. However, the number of registers in a block could be limited and the speed of new blocks can be adjusted by, for example, changing the difficult of the proof-of-work imposed to miners. Beside this, different Blockchains could be used for different systems.

Finally, the restricted devices could also find support in other devices, like those with the same owner in the Relationships Blockchain.

## VII. Conclusion

The Blockchain has interesting features desired for a wide range of domain applications. One example of these domains is the access control. Although other works already use it in this domain, they only explored a little of the great potential of the Blockchain. In this work, we proposed an architecture for access control based on the Blockchain. Our architecture is fully decentralized (requiring no third-party), scalable, user transparent, user friendly, fault tolerant and compatible with a wide range of access control models employed in the IoT. Furthermore, our architecture also includes a secure way of creating relationships, assigning attributes for them and using them in the access control.

## References


[1] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "Fairaccess: a new blockchain-based access control framework for the internet of things," *Security and Communication Networks*, pp. n/a–n/a, 2017, sCN-16-0184.

[2] A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A. Ouahman, "Access control in the internet of things: Big challenges and new opportunities," *Computer Networks*, vol. 112, pp. 237–262, 2017.

[3] B. Krebs, "Hacked cameras, dvrs powered today's massive internet outage," https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/, Oct 2016, visited on 3 Nov 2016.

[4] R. S. Sandhu and P. Samarati, "Access control: Principle and practice," *Comm. Mag.*, vol. 32, no. 9, pp. 40–48, Sep. 1994.

[5] M. Swan, *Blockchain: Blueprint for a New Economy*. O'Reilly Media, 2015.

[6] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang. (2016) Blockchain challenges and opportunities: A survey. [Online]. Available: http://inpluslab.sysu.edu.cn/files/blockchain/blockchain.pdf

[7] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, *Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT*. Cham: Springer International Publishing, 2017, pp. 523–533.

[8] A. A. A. El-Aziz and A. Kannan, "A comprehensive presentation to xacml," in *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*, Oct 2013, pp. 155–161.

[9] D. Hardt, "The oauth 2.0 authorization framework," Internet Requests for Comments, RFC Editor, RFC 6749, October 2012.

[10] Kantara Initiative, Inc., "User-managed access (uma)," https://kantarainitiative.org/confluence/display/uma/Home, Apr 2017, visited on 5 Apr 2017.

[11] Bitcoin, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008, visited on 11 Apr 2017.

[12] Ethereum Project, "A next-generation smart contract and decentralized application platform," https://github.com/ethereum/wiki/wiki/White-Paper, Mar 2017, visited on 14 Mar 2017.

[13] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, Aug 2016, pp. 25–30.

[14] S. Fujimura, H. Watanabe, A. Nakadaira, T. Yamada, A. Akutsu, and J. J. Kishigami, "Bright: A concept for a decentralized rights management system based on blockchain," in *2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Sept 2015, pp. 345–346.

[15] G. Zyskind, O. Nathan, and A. . Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, May 2015, pp. 180–184.

[16] IBM, "Watson internet of things," https://www.ibm.com/internet-of-things/, Aug 2017, visited on 31 Aug 2017.

[17] E. Barka, S. S. Mathew, and Y. Atif, *Securing the Web of Things with Role-Based Access Control*. Cham: Springer International Publishing, 2015, pp. 14–26.

[18] J. Jindou, Q. Xiaofeng, and C. Cheng, "Access control method for web of things based on role and sns," in *2012 IEEE 12th International Conference on Computer and Information Technology*, Oct 2012, pp. 316–321.

[19] G. Zhang and J. Tian, "An extended role based access control model for the internet of things," in *2010 International Conference on Information, Networking and Automation (ICINA)*, vol. 1, Oct 2010, pp. V1–319–V1–323.

[20] N. Ye, Y. Zhu, R. chuan Wang, R. Malekian, and L. Qiao-min, "An efficient authentication and access control scheme for perception layer of internet of things," *Applied Mathematics & Information Sciences*, vol. 8, no. 4, pp. 1617–1624, jul 2014.

[21] G. Zhang and W. Gong, "The research of access control based on UCON in the internet of things," *Journal of Software*, vol. 6, no. 4, apr 2011.

[22] J. L. Hernndez-Ramos, A. J. Jara, L. Marn, and A. F. S. Gmez, "Dcapbac: embedding authorization logic into smart things through ecc optimizations," *International Journal of Computer Mathematics*, vol. 93, no. 2, pp. 345–366, 2016.

[23] P. Mahalle, B. Anggorojati, N. Prasad, and R. Prasad, "Identity authentication and capability based access control (iacac) for the internet of things," *Journal of Cyber Security and Mobility*, vol. 1, no. 4, pp. 309–348, 3 2013.

[24] B. Anggorojati, P. Mahalle, N. Prasad, and R. Prasad, *Secure Access Control and Authority Delegation Based on Capability and Context Awareness for Federated IoT*. River Publishers, 5 2013, pp. 135–160.

[25] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," *Mathematical and Computer Modelling*, vol. 58, no. 56, pp. 1189 – 1205, 2013, the Measurement of Undesirable Outputs: Models Development and Empirical Analyses and Advances in mobile, ubiquitous and cognitive computing.

[26] J. Bernal Bernabe, J. L. Hernandez Ramos, and A. F. Skarmeta Gomez, "Taciot: multidimensional trust-aware access control system for the internet of things," *Soft Computing*, vol. 20, no. 5, pp. 1763–1779, 2016.

[27] R. Neisse, I. N. Fovino, G. Baldini, V. Stavroulaki, P. Vlacheas, and R. Giaffreda, "A model-based security toolkit for the internet of things," in *2014 Ninth International Conference on Availability, Reliability and Security*, Sept 2014, pp. 78–87.

[28] D. Ferraiolo and R. Kuhn, "Role-based access control," in *In 15th NIST-NCSC National Computer Security Conference*, 1992, pp. 554–563.

[29] A. A. E. Kalam, R. E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin, "Organization based access control," in *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, June 2003, pp. 120–131.

[30] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (ABAC) definition and considerations," National Institute of Standards and Technology (NIST), Tech. Rep., jan 2014.

[31] J. Park and R. Sandhu, "The uconabc usage control model," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 1, pp. 128–174, Feb. 2004.

[32] B. Anggorojati, N. R. Prasad, and R. Prasad, "Secure capability-based access control in the m2m local cloud platform," in *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE)*, May 2014, pp. 1–5.