Herausgeber et al. (Hrsg.): Open Identity Summit 2019,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 815

Policy-based Access Control for the IoT and Smart Cities

Olamide Omolola[1] Stefan More[1] Edona Fasllija[1] Georg Wagner[1] Lukas Alber[1]

Abstract: The Internet of Things (IoT) can revolutionize the interaction between users and technology. This interaction generates many sensitive and personal data. Therefore, access to the information they provide should be restricted to only authorized users. However, the limited storage and memory in IoT make it impractical to deploy traditional mechanisms to control access. In this paper, we propose a new access control mechanism based on trust policies adapted from LIGHT^{est}. The proposed protocol also handles delegations in the IoT context elegantly. We provide the protocol overview and discuss its practical applications in the IoT environment.

Keywords: Trust Infrastructure; IoT; Smart City; Access Control; Trust Policy; LIGHT^{est}

1 Introduction

Previous studies estimate that approximately 60% of the world's population is expected to live in major urban areas [Una] by 2030. The United Nations believes that this number will increase approximately by 68% [Unb].

The steady growth of the urban population puts existing urban infrastructure under considerable strain. The future challenge for city planners is the need for more capable infrastructure while using the existing one more efficiently. Around the globe, municipalities are turning towards the Internet of Things and its benefits in infrastructural resilience, improved city services, and management, environmental sustainability, and last but not least operational efficiency - or, in other words, cost reduction.

Many applications are possible using the Internet of Things in smart cities. To mention a few which are of relevance: less traffic congestion using intelligent traffic control, improved public safety, publicly accessible electric car charging, and last but not least an enhanced healthcare system. Individuals living in the cities benefit from these applications.

Many of these applications are sensitive because they deal with personal data or critical public infrastructure. These present a target-rich environment for attackers.

Keeping the information above in mind, one relevant issue of smart cities arises: How can citizens securely access those benefits, without exposing them or the infrastructure to

[1] Technische Universität Graz, Institut für Angewandte Informationsverarbeitung und Kommunikationstechnologie, Inffeldgasse 16a, 8010 Graz, Austria; firstname.lastname@iaik.tugraz.at

security and privacy threats? Typical home IoT standards are usually not applicable in the public domain. Therefore, publicly exposed mechanisms need to cope with this issue.

LIGHT^{est} is an H2020 research project that has received funding from the European Union.[2] The LIGHT^{est} project aims to build a lightweight infrastructure easy and quick verification of electronic transactions. This paper aims to answer the questions asked above using components from LIGHT^{est}. The contributions of this paper are:

- We introduce a policy-based access control model for public IoT services based on the LIGHT^{est} project
- We adapt LIGHT^{est} components such as delegations and trust policies and show their usage in IoT.

This paper is structured as follows: A discussion of the state of the art of access control mechanisms for IoT is given in Section 2, followed by an explanation of the LIGHT^{est} infrastructure in Section 3. We will present our approach in Section 4 and give an overview of the protocol in Subsection 4.1. In Subsection 4.3, we will present an evaluation of our approach. Section 5 concludes and gives an overview of future work.

2 Current State of Access Control in the IoT and Smart Cities

When dealing with public IoT, it is important to ensure that access to devices and resources or their monitoring data is only granted to verified identities that satisfy a specific set of access control rules. This is because IoT devices are inherently attractive targets for attackers due to the privacy-sensitive data and the critical infrastructure they deal generate.

One of the key means to secure and protect devices from those threats is access control mechanisms. Ouaddah et. al. [Ou17] states that access control comprises of authentication, authorization, and accountability. *Authentication* involves the establishment of identities between communicating parties, depending on the communication model that the IoT network applies. *Authorization* deals with the definition of a set of access-control rules in the form of a security policy, and a proper access-control model that is capable of encapsulating this set of rules. Lastly, *accountability* is concerned with guaranteeing the traceability of all the events and actions performed on the IoT system by user or process in a specific entity (device).

Moreover, the access control mechanisms mentioned above can be differentiated via the entity they focus on in the system (User, Device) and the employed communication model as User Authentication vs. Device Authentication, or User Authorization versus Device Authorization.

[2] LIGHT^{est} means Lightweight Infrastructure for Global Heterogeneous Trust management in support of an open Ecosystem of Stakeholders and Trust schemes.

In subsequent paragraphs, we give an overview of the traditional access control mechanisms that have been investigated for IoT such as ACL, ACM, RBAC, ABAC, and so on. We summarize the design of these existing authorization mechanism and discuss on the granularity of permissions that they can grant and the scalability of their design.

Traditionally, **Access Control Matrices (ACM)** have been used for representing access control mechanisms. An **ACM** is a table that lists Subjects and Objects and defines which Subject can access which Object [AS17]. **ACM**, however, is known to suffer from scalability issues, as the size of this matrix can grow when applied to large-scale IoT systems. The concept of **ACM** was used as the basis for the design of two more access control mechanisms, namely (a) **Access Control Lists (ACL)** (b) **Capability-Based Access Control (CapBAC)**. **ACL** differs from ACM in representing the access control rights as linked lists for each object (resource), eliminating in this way the empty cells that would be present in ACM. However, the scalability of **ACL** is still a major issue, especially in the communication models where this **ACL** has to be stored on resource constrained devices. In contrast with **ACL**, which is Object (Resource) oriented, **CapBAC** focuses on the Subject and uses the Capability Authorization Model. A capability is a communicable, unforgeable token of authority, and its possession by a subject grants the subject the access rights of the capability. One major issue is how to prevent an adversary from stealing the capability.

Another well-known access control paradigm is **Role-Based Access Control (RBAC)** [Sa97]. The basic idea of the **RBAC** model is that it lays its foundations on the user's role, rather than its identity (like **ACL** and **CapBAC**). With **RBAC**, multiple roles can be assigned per subject, and access rights can be defined for these roles. The scalability of the **RBAC** model is highly dependent on the roles being well-designed. The right definition of an acceptable number of roles can be a challenge for IoT systems because systems can grow in size and complexity very quickly.

One major drawback of the access control models that have been analyzed so far is that the rights are granted to a subject either based on their identity (**ACL**, **CapBAC**) or roles (**RBAC**). This leads to coarse-grained access rights that cannot consider other important factors in IoT systems, such as time and location. In pursuit of more fine-grained access control models, **the Attribute-Based Access Control (ABAC)** was developed. **ABAC** uses a set of attributes of objects, subjects, and environment to create access tokens. The approach is far more flexible and attractive for IoT systems when compared to the identity or role-centric models. On the other hand, choosing a proper set of attributes and the computation complexity of access policies are some of the main challenges of the **ABAC** model.

A more recent development is the design of the Relationship-based Access Control model (**ReBAC**) which was mainly adopted as a social media access control mechanism. This model grants access rights based on a binary relationship between the entity and the resource [AS17]. In **ReBAC**, an entity can or cannot access a resource based on the relationship that the entity has with the owner of the resource (device). Other alternative

methods to the traditional access control mechanisms have been proposed, such as **CWAC**, **CRBAC**, and **GTRBAC**. In the **Context-Aware Access Control (CWAC)** model [Ki05] access rights are granted based on the context of the subject and object. **CRBAC** [KT08] dynamically integrates **CWAC** and **RBAC**. The **General Temporal RBAC (GTRBAC)** model [Jo] is an extension of the **RBAC** method that can express temporal constraints.

3 The LIGHT^{est} infrastructure

LIGHT^{est} [BL16] aims to create a trust framework for cross-border verification. This trust framework leverages existing infrastructure to provide trust verification of electronic transactions across borders. One such infrastructure is the Domain Name System (DNS). DNS is a hierarchical naming system for devices connected to a network or to the internet. DNS maps human-readable domain names to IP addresses. Domain Name System Security Extensions (DNSSEC) is a suite of protocols that provides origin authentication of DNS data, authenticated denial of existence, and data integrity to the underlying DNS protocol.

LIGHT^{est} uses the DNSSEC root key [HS12] as the global trust anchor. All trust decisions made with LIGHT^{est} can be traced back to this trust anchor. The LIGHT^{est} infrastructure consists of the following components; Trust Scheme Publication Authority (TSPA), Trust Translation Authority (TTA), and a Delegation Provider (DP); and an Automated Trust Verifier (ATV).

In general, someone provides a transaction to the ATV for verification. The transaction is usually signed by the creator[3] of the transaction. The ATV verifies that the transaction is signed correctly and then proceeds to verify to which trust scheme the transaction belongs[4].

In a situation where a verifier uses a different trust scheme from the transaction, the TTA provides translations from one trust scheme to another. ATV can query the TTA for an equivalent trust scheme and use the translation for verification.

Delegations are also supported in LIGHT^{est} since the creator of a transaction could have been empowered to create the transaction on behalf of another entity. The DP provides the validity information and revocation status of a delegation to the ATV using an OCSP-like protocol [WOM17].

The whole process listed above is configured according to the verifier's specific needs with the use of a trust policy. A trust policy is written in a Trust Policy Language adapted from Prolog [MS18].

[3] The creator of the transaction signs the transaction with his signing certificate's private key.

[4] This means that the ATV checks under which trust scheme the certificate that signed the transaction belongs and thereby attributes the transaction to that trust scheme.

4 Approach: On-device authorization

Utilization of resources in smart cities, like IoT devices, is usually restricted to authorized entities alone. This restriction of resource usage and under what conditions the usage is allowed can be enforced by trust policies. Trust policies are rules written in a machine-readable language (in this case, the Trust Policy Language) that describe conditions for certain actions. For example, a trust policy for access control can restrict the access to a certain person or group of people - therefore requiring certain identities. Trust policies can formulate generic rules, e.g., based on context, location, and time.

Furthermore, trust policies can take the readings of sensors into account. It is, therefore, possible to grant or deny access based on a complex set of rules. With an access control based on trust policies, it is quite easy to empower another person to access the IoT device on behalf of the original device owner (or administrator). This empowerment is called delegation and this is an integral part of this approach.

We propose an approach where an ATV component is running directly on a device is performing access control decisions based on trust policies. The trust policy is stored securely[5] in the IoT device. This enables complex use-cases and scenarios by providing all the features that the LIGHT^{est} architecture supports.

4.1 Protocol Overview

This subsection explains the verification process for a client requesting access to an IoT device. We assume that there is an Access-Request client on the user's device that can generate the necessary Access-Request. We also assume that the IoT device is running the Automatic Trust Verifier. The mode of communication between the Access-Request client and the Automatic Trust Verifier on the IoT device can vary depending on the desires of the user. It could be through Bluetooth, NFC, Wi-Fi and many more. This option is left open for the user to decide the most suitable for the use case. We outline protocol steps as shown in Figure 1:

1. **Step 1-2** The user creates an access request using an Access Request client on any device of its choice and signs it with its key ID. The key ID is usually the private key of its key pair or any other form of well-known IDs. The user sends this access request to the IoT device using any means of its choice.
2. **Step 3-6** On receiving the access request, the IoT device starts a challenge-response protocol (any secure, lightweight challenge-response protocol can be used at this stage) and determines if the user still holds the key pair.

[5] The owner of the IoT device can use any secure means of storage available

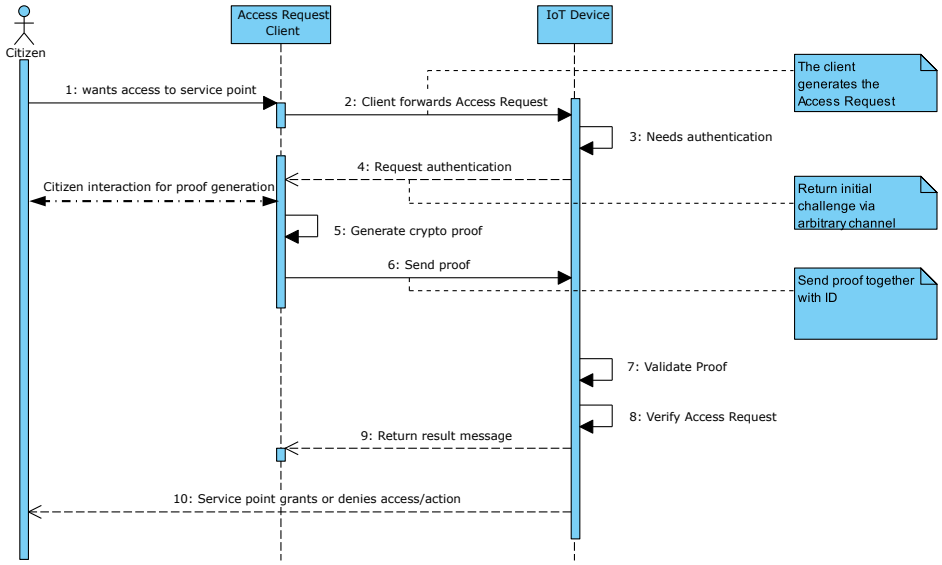

Fig. 1: Protocol Overview.

3. **Step 7** Once the IoT device confirms that the user requesting access is in possession of the ID, the IoT device sends the access request to the Automatic Trust Verifier.
4. **Step 8-10** The Automatic Trust Verifier on the IoT device verifies that the access request fulfils the Trust Policy stored on the IoT device and if a delegation is available, it verifies if the delegation is valid and whether it conforms to the trust policy, too.

4.2 Verification Process on IoT Device

The Verification process on the IoT device begins when the Automatic Trust Verifier (ATV) on the IoT device verifies that the Access-Request is properly signed. After this is verified, the ATV extracts the ID [6], Command and Delegation. The next step is to verify the ID alongside with the delegation. The ID is checked for validity, but the process varies depending on the kind of ID. If there exists a delegation, the ATV verifies the revocation status of the delegation which is stapled (added) to the delegation itself. Once the revocation status is checked, and the delegation is still valid, the verification proceeds and the ATV checks the resource that the identity can access. The restrictions on resources are provided by the Trust Policy which is stored on the IoT device. If the Access-Request Command section conforms to the allowable resources as specified by the Trust Policy, access is granted. The entire process is shown in Figure 2

[6] The ID embedded into the access request is the public key of the private key that created the access request.

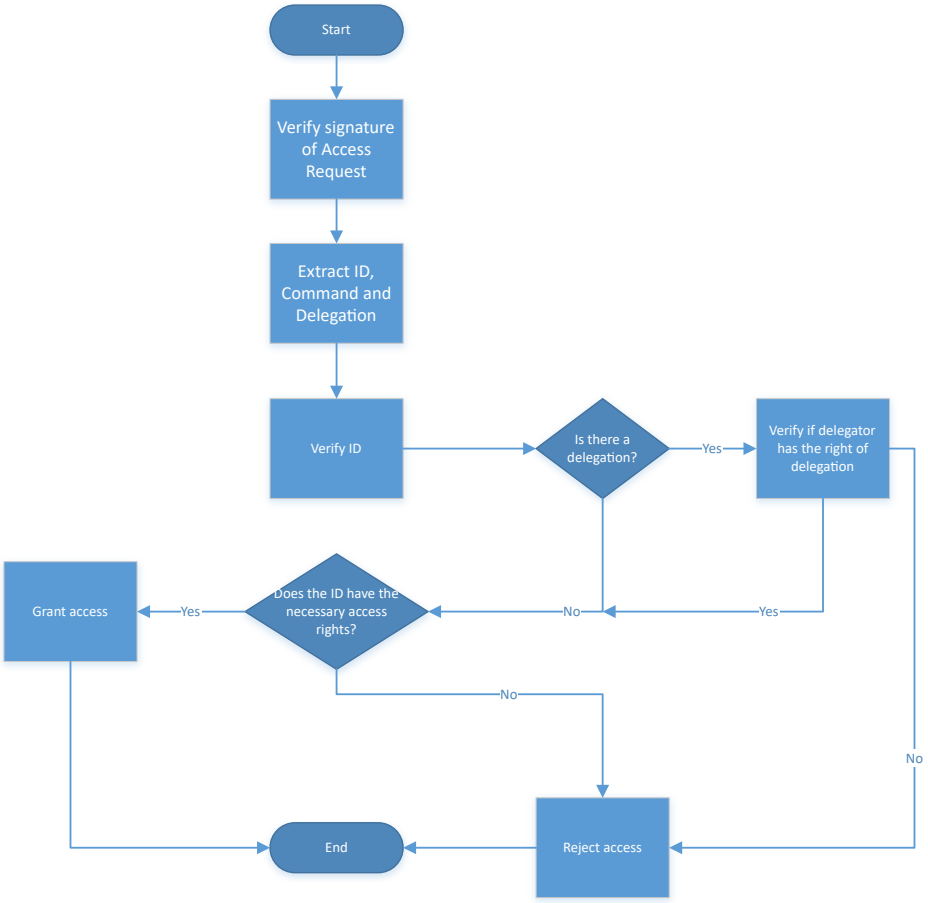

Fig. 2: Verification Process on the IoT Device.

4.2.1 Access-Request Format

The Access-Request consists of two main sections and an optional section: namely the ID section, the Command section, and delegation section. The ID section contains the Public key of the resource requester. This key is the counterpart of the Private key used to sign the Access-Request. This can also contain any form of ID that the resource requester uses. The Command section lists the resources that the user wants to access. If the IoT device does not have multiple resources or the specific levels of access are not defined in the Trust Policy, the Command section is ignored. This Command section gives the owner of the IoT device fine-grained control of the resources on the device.

Figure 3 shows the sections in the Access-Request Format.

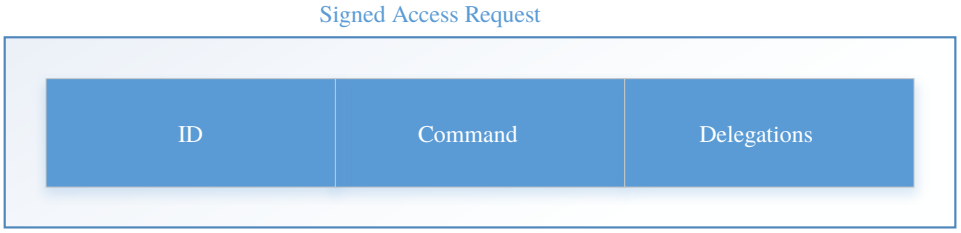

Fig. 3: Access-Request.

The optional section carries the delegation information that the owner of the device assigned to the resource requester. This could be non-existent in some cases where delegation is not necessary. The delegation information contains information about resources the access requester can delegate to another or use to access the resources.

4.2.2 Delegation Format

Delegations occur when an entity, i.e., a mandator, gives another entity, i.e., a proxy, the authority to act on its behalf. Different data formats exist for delegations. We propose that delegation data format should be lightweight because IoT devices have memory and storage limitations. This delegation data format is based on the structure presented in [WOM17]. The structure of the delegation data format can be seen in Figure 4.

The detailed description of the fields is described in [WOM17]. The field *information* contains the *version* of the delegation format and the delegation's *sequence* number. The field *issuedDate* contains the issuance date of the delegation, and it differs from the *validity* field which contains the validity period of the delegation. The *issuer* field contains the identity of the mandator while the *proxy* field contains the identity of the proxy. The *ds:signature* field contains the signature of the delegation. The most notable part of the representation is the *actions* field. This field describes the allowed action(s) that have been

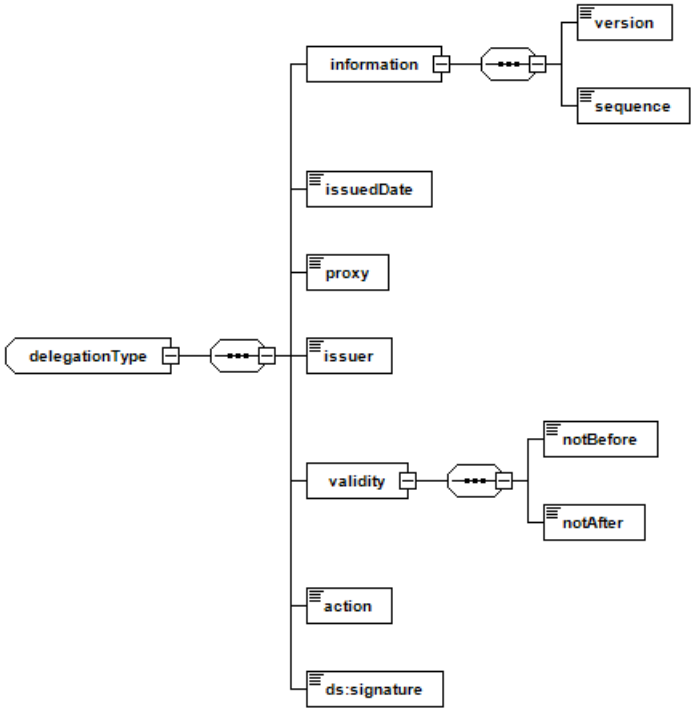

Fig. 4: Delegation data format from [WOM17]

delegated to the user creating the access request. The resource requester needs to add the delegation that it was given to the access request. From Figure 4 we can easily construct the correct data structure with the same field names.

4.3 Protocol Considerations

In this section, we discuss some considerations of our approach.

Unlike the access control models based on identity alone (**ACL, CapBAC**) or roles (**RBAC**), the policy-based access control mechanism provides more fine-grained control because policies can be written to grant or deny rights based on the identity, the role of a subject and other unique conditions such as the environmental or internal conditions.

The policy-based access control is also scalable since a single policy can be written and deployed on several IoT devices and executes a different set of rules depending on the environmental and internal variables of each IoT device.

ABAC is the closest approach to the policy-based access control proposed in this paper but differs in the scalability since different IoT devices on a common network will need different configurations while in the proposed policy-based access contol, a single policy can execute differently on different IoT devices depending on the device conditions.

A major constraint of our approach is the fact that the IoT device must have enough storage and computational resources to run the ATV. Besides, we assume that the IoT device is connected to the Internet. The IoT device needs the Internet to query the external components such as Delegation Provider. The protocol is lightweight and makes it easy to delegate access to others.

5 Conclusion and Future Work

This paper proposed a policy-based access control mechanism that is based on concepts from the LIGHT^{est} project. LIGHT^{est} aims to make trust verification of electronic transactions easier while also leveraging existing infrastructure such as the DNS (Domain Name System). The access control mechanism proposed in this paper allows fine-grained control on the IoT resources. The fine-grained control is reflected in its ability to express complex access control rules via TPL and handle delegations.

In this paper, an ATV runs on a smart device. As part of future work, the ATV will be moved away from the IoT device to an independent system that the IoT device can query. This results in trust verification as a service and frees the IoT device from running an ATV, which frees more computation power and resources from the device. These freed resources can be used for other tasks. We want to achieve this by using an identity derivation scheme

to help the device establish trust in a user/command without talking to the ATV directly. In such a scheme, a command contains all the information needed by the ATV to authenticate the user and execute the trust policy.

In addition, further features of the LIGHT^{est} architecture will be introduced to smart devices. Examples of such features include trust translations, which can be used to verify if an identity in trust scheme A is equivalent to an identity in trust scheme B.

Acknowledgments

The LIGHT^{est} project is partially funded by the European Commission as an Innovation Act as part of the Horizon 2020 program under grant agreement number 700321.

References

[AS17] Alramadhan, M.; Sha, K.: An overview of access control mechanisms for internet of things. In: Computer Communication and Networks (ICCCN), 2017 26th International Conference on. IEEE, pp. 1–6, 2017.

[BL16] Bruegger, B. P.; Lipp, P.: LIGHT^{est} - A Lightweight Infrastructure for Global Heterogeneous Trust Management. In: Open Identity Summit 2016, 13.-14. October 2016, Rome, Italy. Pp. 15–26, 2016.

[FM] FM Conway: A smart solution for parking in London, Accessed: 2018-10-18, URL: https://www.smartparking.com/media/1190/smart-solution-parking-london.pdf.

[Ga] Gartner: Gartner Says Smart Cities Will Use 1.6 Billion Connected Things in 2016, Gartner Newsroom, Accessed: 2018-10-18, URL: https://www.gartner.com/newsroom/id/3175418.

[HS12] Hoffman, P. E.; Schlyter, J.: The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA, RFC 6698, Aug. 2012, URL: https://rfc-editor.org/rfc/rfc6698.txt.

[Jo] Joshi, J. B.; Bertino, E.; Latif, U.; Ghafoor, A.: Generalized Temporal Role Based Access Control Model (GTRBAC)(Part II)–Expressiveness and Design Issues. Submitted to the IEEE Transaction on Knowledge and Data Engineering/ .

[Ki05] Kim, Y.-G.; Mon, C.-J.; Jeong, D.; Lee, J.-O.; Song, C.-Y.; Baik, D.-K.: Context-aware access control mechanism for ubiquitous applications. In: International Atlantic Web Intelligence Conference. Springer, pp. 236–242, 2005.

[KK17] Kulkarni, S.; Kulkarni, S.: Communication Models in Internet of Things: A Survey. International Journal of Science Technology & Engineering 3/, p. 3, 2017.

[KT08] Kulkarni, D.; Tripathi, A.: Context-aware role-based access control in pervasive computing systems. In: Proceedings of the 13th ACM symposium on Access control models and technologies. ACM, pp. 113–122, 2008.

[Mi05] Microsoft: The STRIDE Threat Model, 2005.

[Mo] Mordor Intelligence: Smart Public Safety, Smart Healthcare, Smart Buildings, and Geography - Growth, Trends and Forecast (2018 - 2023), Mordor Intelligence Industry Report, Accessed: 2018-10-18, URL: https://www.mordorintelligence.com/industry-reports/smart-cities-market.

[MS18] Mödersheim, S.; Schlichtkrull, A.: The LIGHTest Foundation, EN 1601-2321, DTU Compute Technical Report, June 2018.

[Ou17] Ouaddah, A.; Mousannif, H.; Elkalam, A. A.; Ouahman, A. A.: Access control in the Internet of Things: Big challenges and new opportunities. Computer Networks 112/, pp. 237–262, 2017.

[Sa97] Sandhu, R.: Rationale for the RBAC96 family of access control models. In: Proceedings of the 1st ACM Workshop on Role-Based Access Control [C]. 1997.

[Una] United Nations: The World's Cities in 2016, United Nations Data Booklet, Accessed: 2018-10-18, URL: %7Bhttp://www.un.org/en/development/desa/population/publications/pdf/urbanization/the_worlds_cities_in_2016_data_booklet.pdf%7D.

[Unb] United Nations: World Urbanization Prospects: The 2018 Revision, United Nations Report, Accessed: 2018-10-18, URL: https://population.un.org/wup/Publications/Files/WUP2018-KeyFacts.pdf.

[WOM17] Wagner, G.; Omolola, O.; More, S.: Harmonizing Delegation Data Formats. In. Oct. 2017.