Securing Device-to-Cloud Interactions in the Internet of Things Relying on Edge Devices

Elías Grande and Marta Beltrán
Department of Computing, Universidad Rey Juan Carlos, Madrid, Spain

Keywords: Access Control, Delegation of Authorization, Identity Management, Internet of Things, Name-oriented Networking, Reverse Addressing.

Abstract: The Internet of Things (IoT) is not a traditional network, and this is the reason why it presents new and unique challenges such as identification, addressing, naming, authentication or authorization of constrained devices. Edge approaches rely on distributed platforms at the network edge serving as a bridge between the physical world (things and data sources, often very constrained devices) and the IoT-cloud services (digital services offered from full-resource servers in the cloud, often not real-time and bandwidth-consuming). The main contributions of this work are the specification of a new event-driven addressing approach for IoT relying on edge-centric delegation of authorization which appropriately adapts and extends the well-known OAuth 2.0 specification for the IoT and a novel approach for naming constrained devices in large scale scenarios that does not depend on the application domain or on the deployment and implementation details. Furthermore, the definition of the Enrolment and Action flows solving the most important challenges arising in the considered scenario: enrolment at the edge device, name-oriented networking, authentication, and authorization using access control tokens as a mechanism for transferring access rights from one agent (edge device) to another (constrained device).

1 INTRODUCTION

The Internet of Things (IoT) is a very challenging context in which many traditional solutions for addressing, naming, authentication or authorization are not suitable due to its inherent scale, heterogeneity, dynamism, complexity and, in many cases, resource constraints. The new Edge Computing paradigm introduces new devices (such as controllers, hubs, smart gateways or micro data centres) at the edge of the network, near constrained devices and able to communicate with both, these devices and resources, services or applications offered in the cloud. These edge devices can be used to decouple the cloud services from the low-level implementation details of protocols used by constrained devices and to offload security functions, etc. In summary, edge devices can be used as logical intermediaries, brokers or proxies between the physical and the Internet/Web/Cloud layers of IoT, raising interoperability and security levels.

This work is focused on delegating complexity to an edge node relying on publish-subscribe mechanisms to solve addressing (reverse addressing), defining generic naming schemes not dependent on devices' implementation, allowing name oriented networking and delegation of authorization based on federated mechanisms (token-based).

The rest of this paper is organized as follows. Section 2 provides an overview of the related work. Section 3 discusses the primary motivations for this work with some examples and potential use cases. Section 4 describes the considered architecture and presents the proposed addressing, naming schemes and authentication/authorization flows, all of them based on an edge-centric approach. Finally, Section 5 summarizes our main conclusions.

2 RELATED WORK

Previous works have proposed different addressing mechanisms specifically designed for the IoT. The first group of these works rely on hierarchical addressing from IoT-cloud services to constrained devices which forces the cloud services to know the underlying deployment of constrained devices and the network they build (Tanganelli et al., 2018), (Moeini et al., 2019). The second group of these works pro-

Grande, E. and Beltrán, M.
Securing Device-to-Cloud Interactions in the Internet of Things Relying on Edge Devices.
DOI: 10.5220/0009804705590564
In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications (ICETE 2020) - SECRYPT*, pages 559-564
ISBN: 978-989-758-446-6

poses the opposite approach: addressing relies on publish-subscribe mechanisms instead of on one-to-one synchronous communications (Lan et al., 2014), (Cheng et al., 2016). This kind of solution can be much more efficient than the previously discussed due to the large scale of usual IoT projects. However, constrained devices need to subscribe to an event bus without intermediaries, with all the resources consumption and security threats that this implies.

Regarding the definition of naming schemes specific for the IoT, there have also been very interesting researches. One group includes in the name of the device information relative to actions it supports or provides. This conditions the naming scheme to the implementation details of the different constrained devices enrolled to the system, (Arshad et al., 2018), (Hail, 2019). The other group tries to propose solutions independent of devices' properties, functionality or implementation details. Therefore, they are not tied to any specific application domain, but, on the other hand, they cannot take advantage of particular devices' characteristics (Yan et al., 2013), (Lee et al., 2015), showing worse performance figures.

With name-oriented networking, interactions are consumer-driven, and consumers request accesses using only the name of the required resource. In addition, these interactions are protected using a data-centric approach: each piece of information (data, file, command) is individually protected. Furthermore, interaction requests must be stateful to enable smart forwarding and management strategies. Some recent works have proposed lightweight mechanisms to guarantee all these properties in the IoT given usual resource constraints (Mahmoud et al., 2019), (Pahl et al., 2019).

Finally, regarding authentication and authorization of IoT devices, the first group of analysed previous works rely on distributed and cooperative approaches to overcome this challenge, trying to propose simple, lightweight and efficient mechanisms that do not consume all available resources IoT sensors and devices. These mechanisms are very often based on different kinds of cryptographic techniques and rely on specific features of devices, communication protocols and application domains (Cirani and Picone, 2015), (Sciancalepore et al., 2018).

The second group of works rely on centralized or federated approaches, trusting in some sophisticated authorization engine or server (or in a federation of them) to make richer decisions regarding access control (fine-grain, based on context or attributes, risk-based). Many of these works are based on the OAuth 2.0 specification (IETF, 2019). It is an authorization framework that allows third-party applications to access resources on behalf of the resource owner, who has previously consented to it. There are recurrent challenges addressed by works relying on the OAuth specification for the IoT (Arnaboldi and Tschofenig, 2019), (Lagutin et al., 2019). Mainly, the secure storage of credentials used for the device authentication and the management of user consents minimizing non-automated interactions involving this user. Both are especially difficult to solve given the limitations of available resources and the high scalability of IoT projects.

3 MOTIVATION, USE CASES AND ARCHITECTURE

There are a plethora of use cases that could benefit from a new approach overcoming the limitations of previous works related to the identification, addressing, naming, authentication and authorization of constrained devices from IoT-cloud services. Good examples can be found in Smart Cities, Industry 4.0, Smart Agriculture or Healthcare scenarios. All these use cases have essential aspects in common not addressed by previous works:

- Addressing based on an asynchronous publish-subscribe approach may be much more efficient than usual addressing linked to constrained devices or protocols specific features. However, resource consumption levels with current solutions are too high due to direct subscription to the event bus. Furthermore, the security levels achieved are usually not enough for many of these use cases, where some of the applications are critical (healthcare, industry, transportation, etc.).
- Naming solutions proposed in the past do not abstract sufficiently IoT-cloud services from implementation and deployment details (of devices and protocols). Furthermore, if they do, they do not usually enable name-oriented networking, a suitable approach for the considered context and a desirable feature given its stateful data-centric security.
- Standard versions of OAuth, a well-known solution to solve authorization, are not suitable for constrained devices because they work over HTTP and TLS. Solutions based on lightweight application protocols are required instead. Moreover, these solutions should be capable of working with very constrained devices not capable of storing their own credentials securely.
- All previous works trying to adapt OAuth to these scenarios have something in common: they do

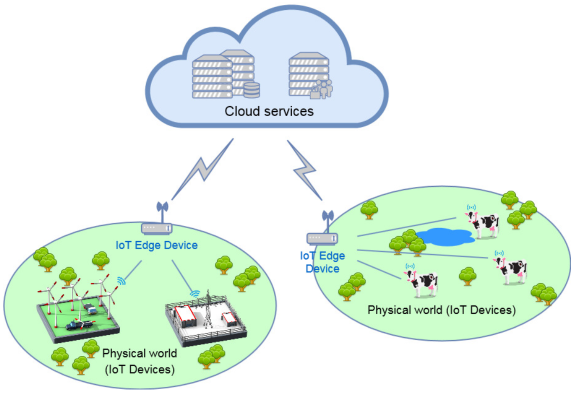

Figure 1: Three-Layer architecture considered in this work.

not focus on the enrolment or registration of devices into the authorization solution, credentials sharing is always performed manually and out-of-band, or it is an issue not explicitly addressed for constrained devices without any secure storage capabilities. This approach does not provide the required levels of scalability in the scenarios mentioned above.

In general, all IoT scenarios requiring high degrees of automation of management because they include large amounts of constrained devices working over lightweight protocols and not capable of storing their own credentials securely, could benefit from the solution proposed in this paper.

The three different roles, shown in figure 1, are considered: IoT-cloud services (running on full-resource servers; delivering computing and communication capabilities, data storage, management or visualization capabilities; and needing to address, name and authorize things to perform certain specific tasks), edge devices (gateways, independent or embedded controllers, sensing terminals or even edge clusters or micro data centres) and constrained devices (things embedded within the physical world with minimal available resources).

Given this architecture, the edge-centric solution proposed in next section specifies how constrained devices can be addressed and named by IoT-cloud services through edge devices which support OAuth 2.0.

According to the considered architecture and our research goals, we first need to enrol constrained devices in the proposed addressing and naming schemes through an edge device. Once a constrained device has been enrolled in the proposed system, it has a name and the IoT-cloud service should have a mean to find it by some kind of address. This name (or identifier) should be unique and agnostic of the authorization server so that identity will be related one-to-one with the provided name, regardless of the edge device in which the constrained device has enrolled. If this feature can be guaranteed, it will support authorization.

4 EDGE-CENTRIC SECURITY IN THE IoT

4.1 Enrolment Flow

This flow is related to the constrained device enrolment, enabling an edge device to identify a single device (or a group of them) that may, at some point, require interaction with an IoT-cloud service. This flow should allow constrained devices to negotiate their own access scopes for these interactions through edge devices solving their addressing and naming.

First, this flow avoids the need for performing a strong authentication that would consume all available (and limited) resources at the constrained device. In order to identify and to authenticate this device this work proposes the use of an identity token issued after the validation of a soft fingerprint built through attributes regarding device's context, static or dynamic (Yang et al., 2019). The device fingerprint can be built relying on its hardware properties, physical or logical addresses, user agent, geo-location, etc. or on behavioural features coming from network protocol analysis (used protocols, headers and payloads sizes, inter-arrival times, etc.). The identity token works as a unique and opaque identifier representing the device's context stored in the cloud during its enrolment.

Second, avoiding a robust authentication model allows one edge device, even with limited resources too, to enrol and to manage a significant number of constrained devices. Third, both soft fingerprint built through attributes regarding the device's context and the identity token generated from that context, allow network independence because there are not authentication mechanisms based on the underlying communication protocol.

Figure 2 shows the proposed Enrolment flow, consisting of the following steps:

1. The constrained device requests to the edge device for the available scopes it has allowed as an OAuth 2.0 client. The application protocol must support this request; it may be a GET or a POST depending on the specific edge device implementation and selected protocol.

2. The edge device retrieves from this request and register all the attributes that describe the current context (the aforementioned soft fingerprint) of the constrained device. If the request was a GET,

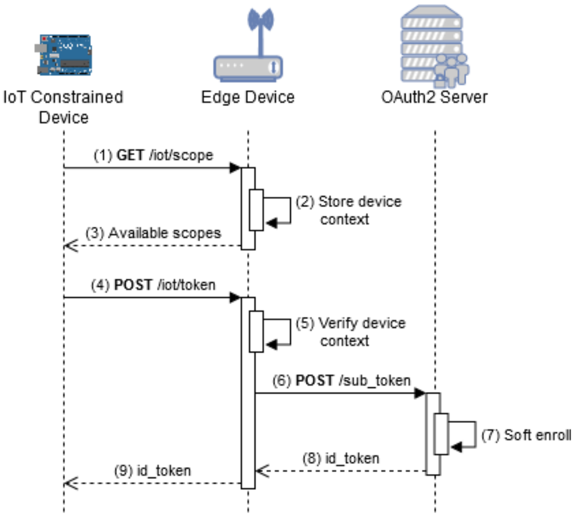

Figure 2: Enrolment Flow.

the context includes the logical address of the constrained device and its user agent depending on the headers included in the protocol implementation. If the request was a POST, the payload could also include the device's physical address, its geolocation, etc.

3. The edge device replies to the constrained device with the set of supported scopes.
4. Once the constrained device chooses which scope among the available best fits its needs, it requests for an identity token valid only for this specific scope.
5. The edge device validates that the requested scope is supported and verifies if the context of the received request matches the previously registered one.
6. If the scope is supported and the context is verified successfully, the edge device requests to the OAuth 2.0 server the delegation of authorization based on its own access token. At this moment, the OAuth 2.0 authorization server links the identity of the specific constrained device with the access token of the edge device for generating a new identity token which represents the new authorization granted to the constrained device. This delegation can be made as many times as needed with all devices which initiate the enrolment flow through the same edge device. This delegation hierarchy allows to the cloud services the ability to revoke all identity tokens linked to a concrete access token only revoking this access token in the same way a Public Key Infrastructure works with the intermediate certificate authorities and the certificates issued by them. In this step, it is assumed that the access token has been previously negotiated between the edge device and the OAuth 2.0 server securely following the traditional specification.
7. The OAuth 2.0 server makes a soft enrolment of the constrained device based on its context (propagated by the edge device), and it links this context to the access token of the edge device. After that, the identity token for the constrained device is generated with a short expiration time (which varies from one application domain to another but that should not exceed one hour), and this token is also linked to the access token of the edge device.
8. The generated identity token is sent back to the edge device. As in any other capability-based model (tokens are traditional access control capabilities which provide permissions to access protected resources), the token must always be transmitted over a secure channel.
9. Finally, the identity token is replied to the constrained device (again over a secure channel) and therefore, the enrolment procedure at this edge device is concluded.

4.2 Event-driven Addressing and Action Flow

Once a constrained device is enrolled in an edge device, it has an identity token, and therefore, it is able to interact with an IoT-cloud service through this edge device. Reverse addressing is the method proposed in this work to enable this interaction. This kind of addressing is based on event-driven mechanisms which allow building large-scale distributed applications within IoT. The routing is based on publish-subscribe mechanisms instead of on one-to-one synchronous communications. Figure 3 shows the proposed Action flow required to support this reverse addressing, consisting of the following steps:

1. The constrained device sends a request to the edge device in which it has enrolled via a GET including two different headers: Proxy-Uri and ETag. The Proxy-Uri header refers to the cloud service URI where the constrained device needs to perform the access with the edge device acting as an intermediary. The ETag header includes the identity token of the constrained device obtained during the Enrolment flow. These two headers may be supported by the selected application protocol or may be defined from scratch.
2. The edge device translates the lightweight request method to the corresponding HTTP method

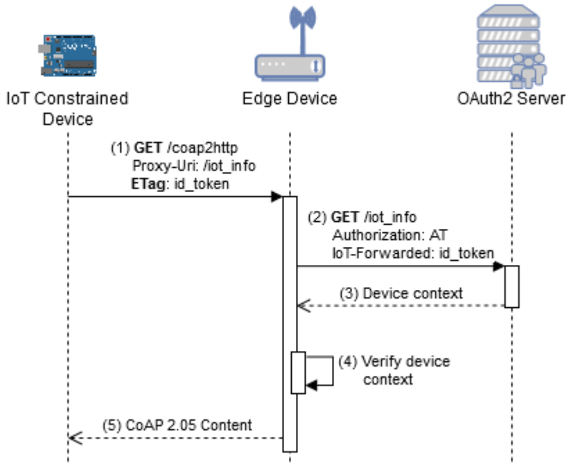

Figure 3: Action Flow.

and sends the request to the URI included in the Proxy-Uri header of the initial request. In this request to the OAuth 2.0 server, the identity token and the access token of the edge device are also required. In the example shown in figure 3, the GET method is translated into an HTTP GET method.

3. The OAuth 2.0 server retrieves the context information associated with the identity token, including actions or orders needed by the cloud service to perform the requested interaction, and it sends back all information to the edge device.
4. The edge device verifies that the context of the constrained device asking to interact with the cloud service matches the available context information retrieved from the OAuth 2.0 server. If the context does not match, information retrieved from the OAuth 2.0 server will not be shared.
5. Finally, the edge device translates the HTTP response to the corresponding response in order to act as an intermediary and to propagate the information to the constrained device. In this case, only the actions or orders coming from the cloud service would be propagated because the constrained device does not need information about its own context.

4.3 Naming

A structured hierarchy of identifiers that can be understood across domains, ecosystems, owners, communities and vendors is required. It would facilitate data sharing, but it is unlikely that all IoT-cloud services and edge devices agree on a single common network-naming scheme. Names can be identifiers if they are unique in some scope, even if constrained devices using these names move within the network or enrols in a different edge device. This kind of names can be used, for example, to verify the integrity or provenance of sensing data, to guarantee non-repudiation or to perform name-based search or discovery of constrained devices.

The use of a naming hierarchy minimizes the probability of names' collision while making easy to check the veracity and uniqueness of a given name and the mapping of the name to a specific device within a broad deployment. It also may improve scalability since new tags or fields could be added if required, supporting name aggregation. Finally, a hierarchical scheme provides excellent compatibility with the existing Internet naming solutions.

In this work, we propose a naming scheme based on XRIs (G. Wachob, 2003), because they provide human and machine-friendly formats which can be expressed as URIs if needed. Furthermore, they enable persistence. The name or identifier of a constrained device can be built using five tags (one more than it was proposed in (van Thuan et al., 2014)):
xri:// Domain Tag / Region Tag / Zone Tag / Edge Tag / ConstrainedDevice Tag.

The first tag is the "Domain" tag. In this work, the name or identifier of a constrained device is agnostic of the identity provider or the OAuth server. In this way, the name is unique, and each constrained device is named in an interoperable way across the systems. This tag starts with a "=" or a "+" depending on if the domain is a person, or another kind of IoT business domain respectively.

The second and third tags are "Region" and "Zone" respectively. These tags represent how the constrained devices are grouped geographically or organizationally. They are optional tags because the deployment of the constrained devices could be not so complex even though that deployment was global. These tags start with a "@" because they are geolocation data or an organizational unit.

The fourth tag is the "Edge" tag, used to identify the gateways, independent or embedded controllers, sensing terminals or even edge clusters or micro data centres to which the constrained devices are connected. This tag is the first in the name hierarchy linked to the technology and to the physical world, and it starts with a "+" because it may be any IoT agent.

Finally, we have the "Constrained Device" tag. This tag represents anything embedded within the physical world with minimal available resources. They can be, for example, any kind of constrained devices like sensors or actuators and therefore, this tag starts with a "+" in the same way that the previous tag.

5 CONCLUSION

This paper has proposed an event-driven addressing and novel XRI-based naming approach for the Internet of Things, relying on a delegation of authorization mechanism based on OAuth 2.0 that enables the authenticated and authorized interaction of constrained devices and IoT-cloud services through edge intermediaries. The proposed solution has demonstrated in different projects that it is scalable (allowing automated enrolment in large scenarios) and interoperable (based on an event-driven polling approach and on the same concepts that standard OAuth 2.0 implementations, only extending them). Enough to solve addressing issues in almost all scenarios with adequate efficiency, fault tolerance and security. Furthermore, abstracting IoT-cloud services from low-level implementation details.

ACKNOWLEDGMENT

This research has been partially supported by the Madrid region (EdgeData, Grant Ref. P2018/TCS-4499) and by the Ericsson-URJC Chair (”Data Science applied to 5G”).

REFERENCES

Arnaboldi, L. and Tschofenig, H. (2019). A formal model for delegated authorization of IoT devices using ACE-OAuth. In *4th OAuth Security Workshop 2019 (OSW 2019)*.

Arshad, S., Shahzaad, B., Azam, M. A., Loo, J., Ahmed, S. H., and Aslam, S. (2018). Hierarchical and flat-based hybrid naming scheme in content-centric networks of things. *IEEE Internet of Things Journal*, 5(2):1070–1080.

Cheng, B., Zhu, D., Zhao, S., and Chen, J. (2016). Situation-aware iot service coordination using the event-driven soa paradigm. *IEEE Transactions on Network and Service Management*, 13(2):349–361.

Cirani, S. and Picone, M. (2015). Effective authorization for the Web of Things. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 316–320.

G. Wachob, D. Reed, M. L. D. M. D. M. (2003). XRI requirements and glossary. http://xml.coverpages.org/XRI-REQv110.pdf.

Hail, M. A. (2019). IoT-NDN: An IoT architecture via named data netwoking (ndn). In *2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pages 74–80.

IETF (2019). OAuth 2.0 device authorization grant. https://tools.ietf.org/html/draft-ietf-oauth-device-flow-15.

Lagutin, D., Kortesniemi, Y., Fotiou, N., and Siris, V. A. (2019). Enabling decentralised identifiers and verifiable credentials for constrained Internet-of-Things devices using OAuth-based delegation. In *Proceedings of the Workshop on Decentralized IoT Systems and Security (DISS)*. Internet Society.

Lan, L., Li, F., Wang, B., Zhang, L., and Shi, R. (2014). An event-driven service-oriented architecture for the internet of things. In *2014 Asia-Pacific Services Computing Conference*, pages 68–73.

Lee, S., Jeong, J., and Park, J. (2015). DNS name autoconfiguration for IoT home devices. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, pages 131–134.

Mahmoud, A., Mahyoub, M., Sheltami, T., and Abu-Amara, M. (2019). Traffic-aware auto-configuration protocol for service oriented low-power and lossy networks in IoT. *Wireless Networks*, pages 4231–4246.

Moeini, H., Yen, I., and Bastani, F. (2019). Service specification and discovery in iot networks. In *2019 IEEE International Conference on Web Services (ICWS)*, pages 55–59.

Pahl, M., Liebald, S., and Lübben, C. (2019). VSL: A data-centric internet of things overlay. In *2019 International Conference on Networked Systems (NetSys)*, pages 1–3.

Sciancalepore, S., Piro, G., Caldarola, D., Boggia, G., and Bianchi, G. (2018). On the design of a decentralized and multiauthority access control scheme in federated and cloud-assisted cyber-physical systems. *IEEE Internet of Things Journal*, 5(6):5190–5204.

Tanganelli, G., Vallati, C., and Mingozzi, E. (2018). Edge-centric distributed discovery and access in the internet of things. *IEEE Internet of Things Journal*, 5(1):425–438.

van Thuan, D., Butkus, P., and van Thanh, D. (2014). A user centric identity management for internet of things. In *2014 International Conference on IT Convergence and Security (ICITCS)*, pages 1–4.

Yan, Z., Kong, N., Tian, Y., and Park, Y. (2013). A universal object name resolution scheme for IoT. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 1120–1124.

Yang, K., Li, Q., and Sun, L. (2019). Towards automatic fingerprinting of IoT devices in the cyberspace. *Computer Networks*, 148:318–327.