Distributed Capability-based Access Control for the Internet of Things

José L. Hernández-Ramos[1*], Antonio J. Jara[2], Leandro Marín[1], and Antonio F. Skarmeta[1]
[1] Department of Information and Communications Engineering
Computer Science Faculty
University of Murcia, 30100 Murcia, Spain
{jluis.hernandez, leandro, skarmeta}@um.es
[2] Institute of Information Systems
University od Applied Sciences Western Switzerland (HES-SO)
Sierre, Switzerland
jara@ieee.org

Abstract

The evolution of the Internet towards the Internet of Things is being deployed in emerging cyber-physical systems such as access control solutions, alert networks, building automation, and the extension of all these systems into Smarter Cities. This extension and proliferation of the technology in our lives is also presenting security challenges, since the unexpected leaks of information, and illegitimate access to data and physical systems could present a high impact in our lives. This work proposes a cryptographic solution against insider threats through a distributed capability-based access control. This access control solution supports the management of certificates, authentication, and authorization processes. The capability-based approach offers benefits in terms of distributed management, support for delegation, traceability of the access, authentication chains to extend scalability and support of standard certificates based on Elliptic Curve Cryptography (ECC). Specifically, it has been designed a capability token for CoAP Resources, which is signed with the Elliptic Curve Digital Signature Algorithm (ECDSA) in order to ensure end-to-end authentication, integrity and non-repudiation. This distributed solution allows the deployment of scenarios without the intervention of any intermediate entity, a distributed scenario with end-to-end access control validation has been implemented, deployed, and evaluated based on the Jennic/NXP JN5139 module. The results obtained through our experiments demonstrate the feasibility of the proposed approach, in numbers, this has required an average of 480 ms to carry out all the validation process (included signature validation in the smart objects).

Keywords: Security, Distributed access control, Cryptographic primitives, Internet of Things

1 Introduction

In recent years, the evolution of the current Internet is leading to a global network of "smart objects" which is commonly referred as *Internet of Things* (IoT). This trend is expected to accelerate in coming years due to cost reduction in hardware and network infrastructure, and the rapid development of communication technologies. This will result in a seamless integration of these objects into the Internet, which will allow the ability to provide mobile and ubiquitous access. A wide range of application domains can take advantage of this resulting paradigm, as shown in some works related to health-care [10, 9], Intelligent Transport Systems [23], smart homes [7], in order to build the next generation of *smarter cities*[2].

Journal of Internet Services and Information Security (JISIS), volume: 3, number: 3/4, pp. 1-16
*Corresponding author: Tel: +34-868888771

This convergence of cybernetic and physical worlds presents a significant enrichment in the user experience, but also several challenges regarding security and privacy, which are considered the main barriers for the deployment of IoT on a broad scale [17]. Unlike traditional contexts, IoT scenarios are expected to be formed by billions of heterogeneous devices with low processing power and memory resources, and interacting with each other through networks with high packet loss and low throughput. Therefore, an appropriate security mechanism for IoT should offer features such as scalability, interoperability, efficiency and lightness in order to properly fit in these emerging scenarios.

Furthermore, privacy is a prerequisite in IoT scenarios (e.g. e-health) since the information that is foreseen to be managed by devices is particularly sensitive. That is why an access control mechanism with the aforementioned characteristics is of imperative need to control the access to this information. In addition, due to the widespread deployment of IoT devices covering everyday *things*, we conjecture that non-expert users will play a greater role defining permissions on their own resources. Therefore, the usability of the access control mechanism is also an important aspect to be considered.

Most of recent proposals have addressed the problem of access control using centralized approaches where a central entity or gateway is responsible for managing the corresponding authorization mechanisms, allowing or denying requests from external entities. Although in these approaches, traditional security mechanisms and access control models such as Role-Based Access Control (RBAC) [6] or Attribute-Based Access Control (ABAC) [26] can be used, end-to-end security between devices and any Internet host can not be achieved. Furthermore, in these scenarios it is necessary to manage the trust of information providers and consumers with this central entity. These problems could be solved by a distributed approach in which "things" are able to make authorization decisions without delegating this process to a different entity. However, in a fully distributed approach, traditional access control models do not meet the requirements imposed by IoT scenarios, introducing lack of flexibility, scalability and usability in environments with billions of devices. It is also necessary to consider the amount of computational resources that are available on the device, since it may not be sufficient to implement a complex access control mechanism. Therefore, the cryptographic algorithms and authentication protocols to differentiate the authorized insiders from outsiders, and consequently avoid the unexpected leaks of the critical data and personal information in ubiquitous ecosystems such as Smart Cities must be properly addressed.

In this paper, we introduce a proposal based on the capability-based access control model. It has some interesting aspects such as the principle of least privilege, and greater adaptation to the requirements presented above. Our proposal, unlike [8], makes use of technologies specifically designed for IoT environments, facilitating a distributed approach in which things themselves are able to make authorization decisions. In addition, these authorization decisions are based on local conditions offering context-aware access control, which is hugely important in typical IoT scenarios. Moreover, unlike other proposals, our work suggests the use of public-key cryptography whose characteristics fit the requirements of IoT regarding scalability and interoperability. Our highly optimized version of Elliptic Curve Digital Signature Algorithm (ECDSA) is implemented inside the thing (i.e., the constrained device such as a door lock, or a sensor) ensuring end-to-end authentication, integrity and non-repudiation, without the intervention of any intermediate entity. The results obtained through our experiments demonstrate the feasibility of our approach and are promising in order to encompass more complex scenarios in the future.

The rest of this paper is structured as follows: Section 2 describes briefly some proposals regarding access control in IoT, Section 3 gives some notions of the different architectures to tackle the problem. Section 4 shows a detailed overview of our proposal and Section 5 provides the results from our experiments. Section 6 offers a discussion about the different approaches for access control in IoT and finally, in Section 7 we end up with some conclusions and an outlook of our future work in this area.

2 Related Work

Recently, there have been various proposals related to access control in IoT with different objectives to address the aforementioned problems regarding emerging IoT scenarios.

The authors in [27] present an abstraction of the usage control (UCON) model in IoT. The proposal is based on fuzzy theory and some central entities which manage usage control decisions and trust values of devices and services. Moreover, although some experiments are presented, the practical feasibility of the approach is not demonstrated.

The work presented in [14] provides an approach based on Elliptic Curve Cryptography (ECC) for key establishment and RBAC model for defining access control policies. The proposal makes use of OpenID [20] technology and trustable central entities for authentication purposes. The limited suitability of RBAC for IoT scenarios has already been explained above. Furthermore, experimental results are not provided and its feasibility is not addressed.

The implementation of capability-based access control in IoT is considered in [8]. The proposed framework is based on a central Policy Decision Point (PDP) which handles authorization decisions. Therefore, when the entity tries to access data from a sensor, attaches the capability token to the access request, then the PDP is responsible for deciding whether the entity is authorized or not. The decision is based on the received capability and the internal rules defined for this sensor. Our work is based on the principles of this proposal. However, we enable a distributed approach without intervention of central entities, by adapting the communication technologies and representation to the IoT scenarios requirements.

The proposal described in [15] also makes use of capability-based access control. Capabilities are exchanged in conjunction with a SHA-1 message digest, which is used to check the tampering and forgery of the capabilities. However, they do not provide details on the communication technologies employed, or details on the representation and content of the capability. Finally, ECC-based cryptography is not considered.

The work described in [16] proposes the use of Elliptic Curve Diffie-Hellman (ECDH) to establish shared secret keys between two devices and access control also based on capabilities. Likewise the previous proposal, they do not provide details on the communication technologies employed, or details on the representation and content of the capability. Moreover, its solution is based on the use of a central entity that is responsible for carrying out calculations related to ECC, preventing end-to-end security.

Authors in [1] present a delegation model called Capability-based Context-Aware Access Control (CCAAC). They propose a vision of federated IoT, where a central entity in each domain is in charge of authorizing a delegation request from a delegator, and make the decision about grant it to the delegate.

A close proposal is described in [24], which presents an authorization framework based on assertions as a result of an authorization process based on XACML [18]. This assertion is encoded in JSON [3] and is sent to the end-device (i.e., sensor or constrained device). The end-device takes responsibility for local conditions verification. However, unlike our work, this approach is bound to the use of XACML both for the authorization decision and representing local conditions to be verified on the end-device. Our proposal does not explicitly require the use of servers for authorization and token representation is based on standards specifically designed for use in constrained devices. Furthermore, the use of public key cryptography based on ECC is not considered.

3 Access Control architectures for IoT

This section offers an overview of the main approaches and trends to provide access control logic in IoT scenarios. Specifically, the architectures proposed in [22] are discussed. In addition, the key issues of

these approaches related to access control, highlighting their main advantages and drawbacks for large scale deployments are provided.

3.1 Centralized approach

In a centralized approach, all access control logic is externalized into a central entity or central PDP responsible for filtering access requests based on their authorization policies. This entity could be instantiated by a gateway with direct communication to the devices that it manages, or another entity in a different location, as shown in Figure 1. Thereby, end-devices (i.e., sensors, actuators) play a role limited to as information providers. Therefore, they send their data to the central element responsible for making authorization decisions. In this case, users wishing to access the data provided by the devices, should be connected directly to the interfaces offered by the central entity.

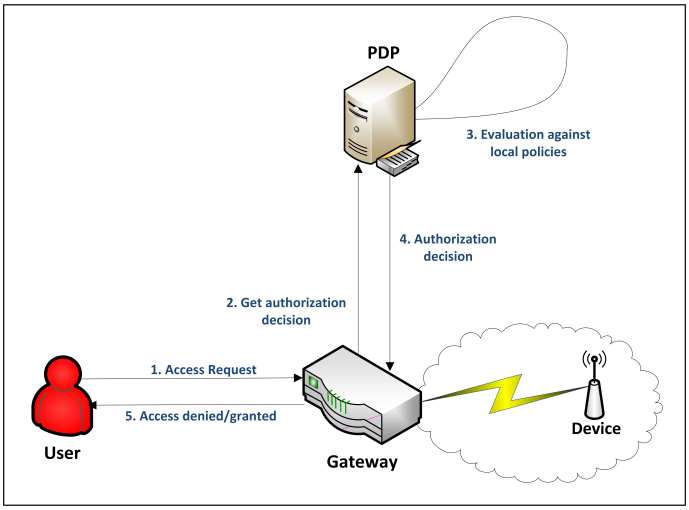

Figure 1: Centralized approach

The main advantage of this approach is that access control logic is located in an entity without constraints of resources, in contrast to the end-devices, featured by constrained capabilities. This entails the usage of standard security and access control technologies. For example, SAML [19] and HTTPS [21] could be used for secure transport of authentication information, as well as XACML [18] could be considered to define complex access control policies. Thus, modifications in the end-devices are not required.

However, there are some major problems when a centralized approach is considered. First, access control decisions are not based on contextual information related to the end-device itself; this contextual information is highly relevant in several IoT scenarios such as safety and alert networks. Second, end-to-end security is compromised because of a central entity is required to make the authorization decisions. Therefore, this entity will need to read the content of the access query, which also compromises the privacy of the requester. Third, in such architectures, it is necessary to manage the trust of information providers and consumers with that entity. Although there may be environments where this vision fit, feasibility in IoT scenarios such as smart cities or ITS is quite doubtful. Finally, due to the fact that a single entity stores and manages all the data from a set of devices, it becomes a single point of failure, consequently, any vulnerability might compromise a vast amount of sensitive information.

3.2 Centralized and Contextual approach

This section presents a hybrid approach. End-devices are no longer completely passive entities, since they participate partially in the access control decisions such as presented in the Figure 2. This approach is motivated by the importance of taking into account the context of the end-device for making the decision. The context is particularly important in pervasive environments such as those addressed by the IoT. For example, in the case of e-health, a patient with embedded sensors may require that the hospital staff have access to his data, while he is suffering a medical emergency. Therefore, this type of context-dependent access control is not feasible without the information that the end-devices are able to provide at the time of the access request in terms of location, environmental status (light level, temperature, humidity, CO2 level, etc.), the interaction with other users (buttons), user's status (heartbeat rate, SPo2 level), etc. Hence, regarding the described use case for emergencies, the access control rule can not be applied, consequently, the context-agnostic solution will endanger the patient's privacy.

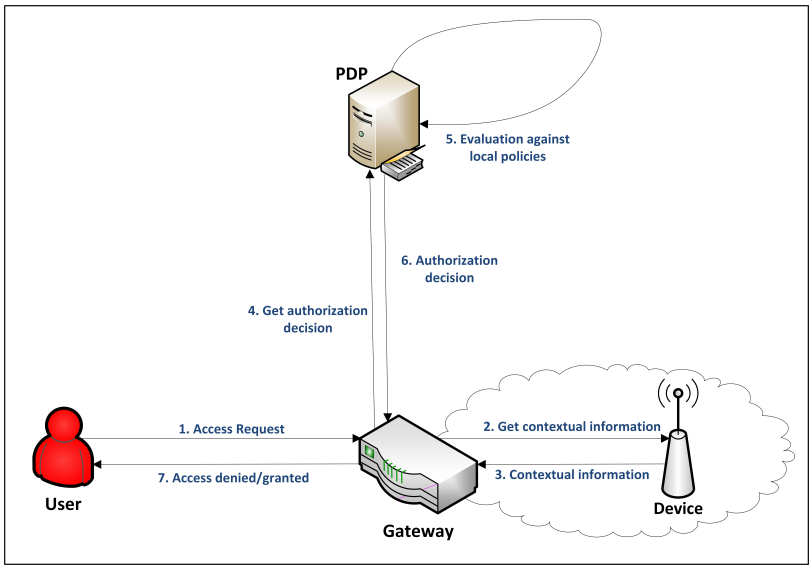

Figure 2: Centralized and contextual approach

At the same way that in the centralized approach, the hybrid scenario allows to use standard technologies to perform the authorization process. However, in this case, an extension of the end-devices is required to transport contextual information request. This transport can be carried out with application protocols such as the Constrained Application Protocol (CoAP) [25].

This approach will also require an access control mechanism to be performed by the end-device in order to provide or not its contextual information. Nevertheless, in a centralized approach, a trust relationship is assumed between the devices and the central entity. Following the previous approach, this trust establishment might be unfeasible in many of the scenarios foreseen for the IoT. Furthermore, another problem arises when the contextual information from the end-device has to be transmitted to a central entity, which introduces delays. Therefore, the value obtained by the end-device is different at the time of making the authorization decision. Finally, end-to-end security can not be achieved.

3.3 Distributed approach

Alternatively, in the distributed architectures, all the access control logic is embedded into the end-devices. These devices are being enabled with capabilities to obtain, process and send information to

other services and entities. For that reason, they are also commonly called ”smart things” or ”smart objects”. Thereby, the end-node is able to carry out the authorization process, without the need for a central entity. Figure 3 shows this approach.

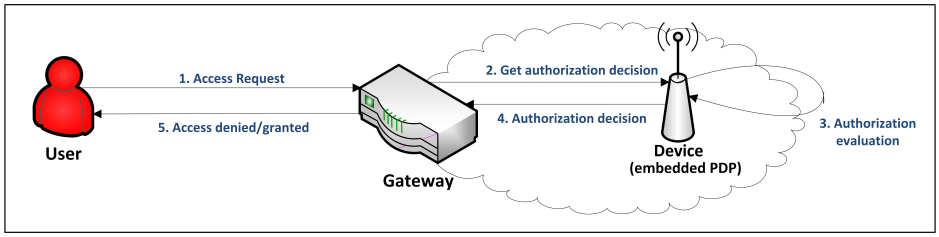

Figure 3: Distributed approach

There are numerous advantages that motivate the use of a distributed approach for access control in IoT scenarios. First, the end-devices are no longer passive entities that only send information to a central entity. End-objects are being smart, and consequently they have capacity to manage their information. Second, unlike previous approaches, devices are able to send information just when necessary, since in this case there is no central entity responsible for gathering the data from the devices in order to make decisions about access control. Third, the removal of intermediate entities enables to carry out end-to-end security for the access requests. Additionally, these devices can cooperate directly with each other in a natural and direct way to exchange information when needed.

The most obvious drawback arises from the need to extend the devices with access control logic. In this case, the inherent features of the traditional access control models, such as RBAC and ABAC, make their implementation unfeasible in resource-constrained devices. Consequently, this approach must be addressed in-depth by analyzing the viability of different access control models or defining new proposals that meet the requirements of a distributed access control approach. Moreover, the access control mechanism must be practical from the security point of view. For that reason, protocols that guarantee basic security properties should be also considered for a final solution. However, these mechanisms are often based on cryptographic primitives that present a high computational cost, since the usually considered security for constrained end-devices such as symmetric-key cryptography does not satisfy the principle of scalability for IoT scenarios. As a conclusion, it is also necessary to define optimized public-key cryptographic primitives that make the access control mechanism feasible and effective.

Previous approaches have several advantages and drawbacks that need to be considered in order to properly address the design of an access control solution for IoT scenarios. While in centralized approaches, standard security mechanisms and protocols can be used to prevent unauthorized access, constraints on end-devices make these solutions need to be adapted towards more lightweight mechanisms in order to make a distributed approach feasible. However, with the current trend to a global network of smart things, this approach presents interesting features regarding scalability, interoperability and context-awareness, making it more suitable to tackle the inherent requirements of the Future Internet.

4 Design of our proposal

This section provides a detailed description of the most important aspects of our proposal. First, a brief explanation of the most relevant features of the capability-based access control model and the necessity for a lightweight solution is presented. Then, we give an overview of the main aspects of our work based on this model for IoT scenarios. In particular, the structure of the capability token and the operation of our proposal are provided.

4.1 Capability-based access control

Traditional access control models such as RBAC or ABAC have been widely used in a multitude of authorization scenarios so far. However, the suitability of them for a distributed access control in IoT devices has not been demonstrated. These models usually require a consistent definition of the meaning of roles and attributes, as well as complex access policies, which negatively affects their adequacy to the scenarios envisaged by IoT with billions of end-devices. Therefore, there is a need to consider alternative approaches to popular access control models in order to make a distributed approach possible. In recent years, the access control model based on capabilities (CapBAC) has been postulated as a realistic approach to be used in IoT [8]. Capability-based access control offers relevant features such as the principle of least privilege, a higher degree of usability and flexibility, and therefore, a greater adaptation to cope with the requirements of the Future Internet.

The key concept of this approach is the capability. The concept of capability was originally introduced in [4] as *"token, ticket, or key that gives the possessor permission to access an entity or object in a computer system"*.

Additionally, the token must be tamper-proof and unequivocally identified in order to be considered in a real scenario. Optionally, it can be defined a set of rights that have been granted to the entity holding the token. One aspect to note in CapBAC is that an entity, which wishes to access certain information from another entity, requires to send a token together the request. Thus, the entity that receives the capability already knows the right level (i.e., permissions) that the requester has been granted when need to process the request. This simplifies the authorization mechanism, and it is a relevant feature in scenarios with resources-constrained devices, since complex access control policies are not required.

CapBAC, and its application for the IoT, has been deeply analyzed in [8], which is based on the ideas from [5]. The proposal describes a centralized approach in which a PDP is responsible for managing the access control mechanism. For this purpose, the capability is based on standard technologies such as XML, SAML and XACML, which is sent by a requester when trying to access the information of a end-device. However, the use of these technologies is not appropriate in order to make possible a fully distributed access control. In particular, it is necessary to adopt lightweight mechanisms, in order to make feasible its integration in constrained resources devices. Furthermore, security aspects as entity authentication or digital signature are not addressed. Additionally, they offer support for some important aspects such as delegation support (a subject can grant privileges to another subject) or revocation of capabilities (capabilities can be revoked by properly authorized subjects), which are inherent to CapBAC.

4.2 Capability token

In order to implement our distributed capability-based access control approach, we chose JSON [3] as format to represent the capability token because of its suitability in constrained environments, such as those suggested by IoT scenarios. Figure 4 shows a capability token example, which has been used to demonstrate the feasibility of our proposal. Below, a brief description of each field is provided.

- **Identifier (ID)** (*16 bytes*). This field is used to unequivocally identify a capability token. A random or pseudorandom technique will be employed by the issuer to ensure this identifier is unique.
- **Issued-time (II)** (*10 bytes*). Following the notation of [12], it identifies the time at which the token was issued as the number of seconds from 1970-01-01T0:0:0Z.
- **Issuer (IS)** (*variable size*). The entity that issued the token and, therefore, the signer of it.
- **Subject (SU)** (*56 bytes*). It makes reference to the subject to which the rights from the token are granted. A public key has been used to validate the legitimacy of the subject. Specifically, it is

```
{
  "id": "0h7be34m_0q2cx-7",
  "ii": 1369300359,
  "is": "jara@um.es",
  "su": "lnnH3/IYZz/pqBbSUd+JOyMtNCM=g2o3XRd/3r7iZSjpIIX9BRRULtc=",
  "de": "coap://aaaa::2/",
  "si": "uPTor1jxykFQUGxXnRVRm0l+uZM=kebPXS9VERYSyX3VFeHH9gW/yQI=",
  "ar": [
    {
      "ac": "GET",
      "re": "temperature",
      "f": 1,
      "co": [
        {
          "t": 5,
          "v": 25,
          "u": "Cel"
        },
        {
          "t": 6,
          "v": 21,
          "u": "Cel"
        }
      ]
    }
  ],
  "nb": 1369300359,
  "na": 1369300500
}
```


Figure 4: Capability Token example

based on ECC-based public-key cryptography, therefore, each half of the field represents a public key coordinate of the subject using *Base64*.

- **Device (DE)** (*variable size*). It is a URI used to unequivocally identify the device to which the token applies.
- **Signature (SI)** (*56 bytes*). It carries the digital signature of the token. As a signature in ECDSA is represented by two values, each half of the field represents one of these values using *Base64*.
- **Access Rights (AR)**. This field represents the set of rights that the issuer has granted to the subject.
  - **Action (AC)** (*variable size*). Its purpose is to identify a specific granted action. Its value could be any CoAP method (GET, POST, PUT, DELETE).
  - **Resource (RE)** (*variable size*). It represents the resource in the device for which the action is granted.
  - **Condition flag (F)** (*1 byte*). Following the notation in [13], it states how the set of conditions in the next field should be combined. A value of 0 means AND and a value of 1 means OR.

  - **Conditions (CO)**. Set of conditions which have to be fulfilled locally on the device to grant the corresponding action.
    * **Condition Type (T)** (*1 byte*). The type of condition to be verified as stated by [13].
    * **Condition value (V)** (*variable size*). It represents the value of the condition.
    * **Condition Unit (U)** (*variable size*). It indicates the unit of measure that the value represents. Its value could be any of those stated by [11].

- **Not Before (NB)** (*10 bytes*). The time before which the token must not be accepted. Its value cannot be earlier than the II field and it implies the current time must be after or equal than NB.
- **Not After (NA)** (*10 bytes*). It represents the time after which the token must not be accepted.

4.3 Distributed CapBAC operation

The basic operation of our proposed distributed capability-based access control is shown in Figure 5. Below, we clarify the different steps of the process.

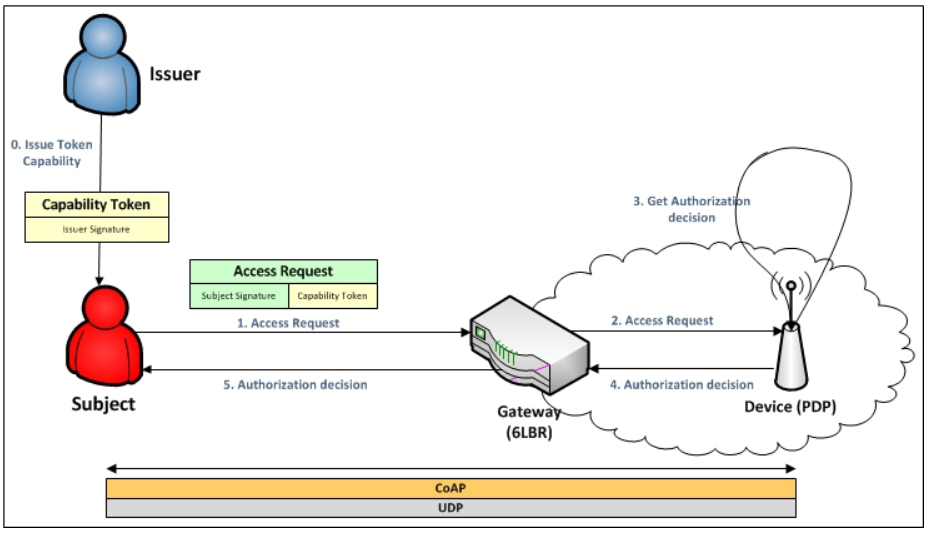

Figure 5: Distributed capability-based approach

- **Issue Capability Token**. As initial step, the *Issuer* entity, usually instantiated by the device owner, issues a capability token to the *Subject* to be able to access that device. Additionally, in order to avoid security breaches, the *Issuer* signs this token. As described in the previous section, this generation is based on the ECDSA algorithm, and its value is attached to the capability token.

  The process of how to generate the token is outside the scope of this document. For example, in some contexts, the generation of this token might require a previous authentication process between *Issuer* and *Subject*. Conversely, in other IoT scenarios, this token might be sent only to subject which meet certain conditions, e.g., based on geolocation. Therefore, this stage requires a deeper analysis and will be explored in future work in order to assess their adaptability to different IoT scenarios.

- **Access Request**. Once the *Subject* has received the capability token, it attempts to access the device data. For this purpose, it generates a CoAP request, in which the token is attached. The

inclusion of the token into the request has been carried out using the *payload* field, and the *Content-Format* header to indicate the representation format of the payload of the request. In addition, the *Request-Uri* option is used to specify the specific resource to be accessed in the *Device*. Finally, the *Subject* also signs the request itself using ECDSA algorithm, whose value is attached to that request by adding a new header called *Signature*.

Following the Figure 5, this request does not have to be read by any intermediate entity. In fact, the entity which is denoted as gateway could be instantiated by a 6LowPAN Border Router (6LBR) without ability to understand the content of the message, i.e., it has basic routing functionalities.

- **Get Authorization Decision**. When the *Device* receives the access request, the authorization process is carried out. First, the application checks the validity of the token as well as the rights and conditions to be verified. Then, due to the cost of these operations, the *Issuer* signature is verified and *Subject* is authenticated. A more detailed description of this process is given in Section 4.4.

- **Return Authorization Decision**. Finally, once the authorization process has been completed, the *Device* generates a CoAP response based on the authorization decision. In the case of a unauthorized request, a *Unauthorized 4.01* response is returned, indicating that the *Subject* is not authorized to perform the requested action. Otherwise, the value of the answer will depend on the content of the request.

4.4 Authorization process

This section analyses the authorization process carried out in the end-device. This procedure is launched after a CoAP request containing the capability token is received. The Figure 6 shows a block diagram with the steps to be completed by the device in order to provide an authorization decision.

- **A. Check that the token is valid**. First, the device checks whether the token is valid when it is received or not. The fields *II*, *NB* and *NB* are used for this purpose. In the case of the token is not valid, the authorization process stops and the request is not authorized.

- **B. Check that the action is granted**. For each one of the access rights, the device checks if the action is permitted. That means, check that the CoAP method matches the *AC* field in a specific *AC* element. Additionally, the value of the *RE* element is compared to the *Request-URI* option of the CoAP request. In the case that one of these verifications fails, the next *AR* element is taken for evaluation, if any.

- **C. Check that the conditions are fulfilled**. After checking the action on a specific resource is permitted, the conditions have to be verified on the device. For this, the field *F* and the set of conditions specified in *CO*, are used. If conditions are not met, the next *AR* element is taken for evaluation, if any.

- **D. Check that the signature is valid**. Before granting the permission for a specific request, the signature of the token specified in the *SI* field has to be verified. For this purpose, the end-device makes use of the Issuer's public key; this consideration is suitable since the issuer will be usually instantiated by the device owner. This phase is left to the end due to the cost required by the cryptographic operations. In addition, an optimized version of ECDSA based on shifting primes has been considered, since this reduces the time needed to verify the signature contained in the token.

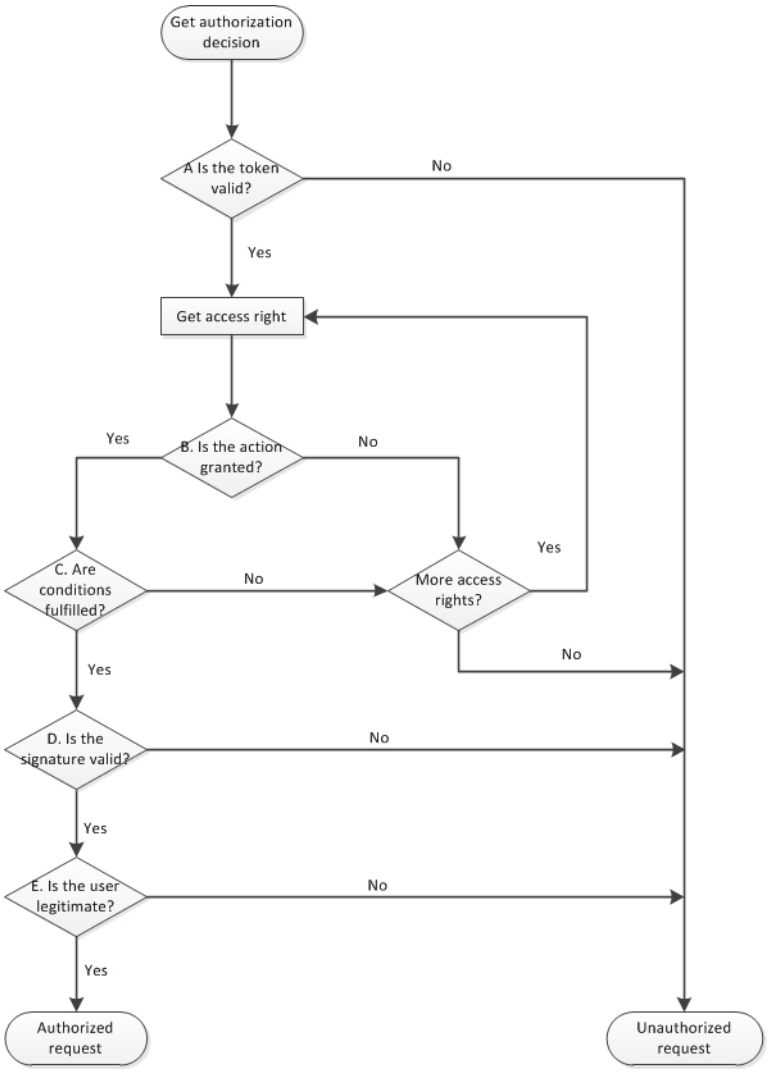

Figure 6: Functional diagram for authorization process

- **E. Check that the user is legitimate**. Finally, the end-device must be sure the subject of the request is a legitimate user. For this purpose, the end-device makes use of the public key of the subject which is attached in the *SU* element of the capability token, and the value of the signature which is contained in the *Signature* header of the CoAP request. As in the previous operation, this validation is left to the end, since it has a high computational cost, thereby, in case that some of the previous requirements fails, this checking is omitted.

5 Evaluation

Once the most important details of our distributed approach have been presented, a test bed has been designed to test our proposal. Below, we offer a description of the experimental results in order to demonstrate the feasibility of the proposed solution.

5.1 Test bed features

According to Figure 5, the elements of our architecture are: the end-device, which will accept access requests and will perform the functionality described in the previous section, the issuer, responsible for generating the capability token, the subject, which will send requests to the device, and a 6LBR, forwarding access requests to the device itself. As the capability token generation stage is out the scope of this document, and the 6LBR functionality is trivial, we focus on the features of the end-device and subject (user).

The resource-constrained device of our scenario was implemented in a JN5139 mote equipped with Contiki OS. JN5139 is a low power, low cost wireless microcontroller suitable for IEEE802.15.4 applications with 16MHz clock. The CPU of the JN5139 is a 32-bit load and store RISC processor with low power consumption for battery powered applications. The ROM is 192 kB in size, organized as 48k x 32-bit words, and RAM contains 96 kB organized as 24k x 32-bit. Moreover, the subject has been implemented in Java on a common computer. In a real scenario, it is expected that this entity is instantiated by devices without resource constraints such as smartphones and tablets that can be used by non expert users to access data from a particular device.

5.2 Experimental results

For the proposed scenario, 100 tests have been carried out. According the results obtained from these tests, using the token capability example shown in Figure 4, the average time required for the operation of our scenario is **480.96 ms**. This time includes the Round-Trip delay Time (RTT) since the subject sends the request until it gets the authorization response. At this point, it is worth mention that this time assumes that the subject is authorized to perform the action and, therefore, all steps of the authorization evaluation must be completed. In the case any of the steps in the evaluation fails, the authorization process will be immediately aborted, without carrying out all authorization stages. Therefore, in case the subject is unauthorized, the time to get the decision will be always lesser. Next, we give a brief description of the average times obtained from each of the main phases in the operation. These times are summarised in Table 1

Stage	Time (ms)
A, B, C	52.39
D	213.93
E	214.64
Total	**480.96**

Table 1: Evaluation results for each stage

According to Figure 6, this stage corresponds to the functional block labelled as *D*. This time includes the process for validating the signature which is contained in the capability token, including base64-decoding as well as the time required for calculating the digest value of the token, using the MD5 algorithm. The average time obtained for this step is **213.93 ms**.

This step matches with the functional block labelled as *E* in Figure 6. This process includes the time required to validate the signature included in the CoAP request, including base-64 decoding, the time needed for calculating the digest value of the request, using the MD5 algorithm. Additionally, this demands the time for base64-decoding of the subject's public key, which is within the capability token. The average time for this step is **214.64 ms**.

Finally, this phase includes the rest of procedures required to obtain the authorization decision, which are labelled as *A*, *B* and *C* in the Figure 6. Also, this period covers the time for parsing JSON capability

token and CoAP access request. The average time obtained for this step is **52.39 ms**.

6 Discussion

Due to constraints in the IoT devices, centralized proposals to provide access control logic have received great acceptance so far. However, a distributed approach should be considered in order to cope with the challenges regarding scalability and end-to-end security, which can not be fulfilled by traditional approaches. For example, building automation and access control solutions such as a typical scenario with a smart phone and a smart door lock cannot rely on centralized servers requiring Internet-connectivity for several reasons. First, this alternative increases the cost of the solution with an additional service to be subscribed. Second, the user experience can be affected with connectivity problems. Finally, this solution can not provide a suitable level of scalability for the current trend towards billions of end-devices (smart objects) connected to the Internet.

In order to make a fully distributed approach feasible, we propose the use of access control based on capabilities since this offers all the security of a centralized mode in terms of validation of the issuer, subject authentication and authorization validation process, but at end-to-end level. Thereby, the proposed approach is not presenting the dependencies regarding connectivity to make use of the service and subscriptions to maintain it available. Instead, the interaction is built over a peer-to-peer scenario making it much more intuitive and natural to use from the human experience, providing more scalability and removing the requirements of infrastructure which are required by centralized approaches.

Furthermore, our results show that a distributed approach can be feasible, but more effort to overcome security obstacles is needed, which can be demonstrated in future work with the instantiation of the proposal in real IoT scenarios and ECC optimizations. In this work, an optimized ECDSA implementation for constrained devices based on shifting primes has been employed. In addition, a Public Key Cryptography solution based on ECC has allowed the usage of certificates and standards Public Key Infrastructures (PKIs), which facilitates the convergence into a common and interoperable security infrastructure.

Finally, even when the solution is feasible and a working proof-of-concept has been developed, capability-based access control requires the definition of a model for dynamic and context-based management of the conditions in order to reach a real market. The commissioning of the capabilities and the evolution of them is a new research area that needs to be explored. One of the major challenges of these solutions, once the technical issues are solved, is to provide an intuitive and secure interaction with the user, devices, and ecosystem, in order to make it useful and feasible for real environments. We need to take into account that for the emerging ecosystems based on smart objects, it is not feasible the commissioning of each device one by one as well as the modification of the conditions through an explicit interaction with the user. Therefore, a highly relevant task is the proper definition of the life-cycles of all the appliances, users and smart devices for issues such as the upgrade of the systems for new devices, users or requirements.

7 Conclusions and future work

The network security is a pending challenge for the IoT and related technologies such as 6LoWPAN. The IoT requires the boundary protection and confidentiality enhancement through the usage optimized of cryptographic primitives such as ECC, and authentication protocols to differentiate the authorized insiders from outsiders. Security for the IoT is a major requirement since the inclusion of cyber-physical systems is being extended to real-life scenarios such as hospitals, home automation, building access

control solutions, and smart cities. Therefore, the expected leaks of proprietary and private information can lead to significant damage to personal and corporate entities.

For that reason, this work has presented a fully distributed approach for certification and authorization, without intermediate entities implementing access control logic. This has offered the possibility to exploit the IoT potential in terms of end-to-end connectivity based on IPv6, access to the resources through CoAP methods (RESTFul interface), and finally to provide an access control solution on top for CoAP Resources.

Our solution has been implemented taking into account the constraints of the existing smart objects in terms of communication and processing power. Thereby, the proposed capability token has been represented with JSON and, in addtion, ECC optimizations have been used to carry out the cryptographic operations, which enables expensive tasks such as signature validation are made at the end-device. These optimizations and design considerations have made possible its development and evaluation over a real platform based on the Jennic/NXP JN5139 chipset. The time required for the evaluation of the capability token has been under 0.5 seconds (480.96 ms), making it totally feasible for a real scenario.

The future work is focused on exploring additional features of the capability-based access control such as revocation management and delegation, as well as its application into specific IoT scenarios. The delegation process requires a signature validation per level of legation. Since the signature validation is the most expensive step, additional security enhancements based on ECC optimizations will be evaluated. Furthermore, a theoretical analysis regarding resistance against security attacks and threats of our proposal (including the validation into mathematical frameworks such as AVISPA) will be carried out. Finally, another relevant research line related to this work is the consideration for additional privacy enhancement through techniques such as the use of pseudonyms or anonymous capabilities.

Acknowledgment

This work has been made possible by the means of the Excellence Researching Group Program from Foundation Seneca (04552/GERM/06), the FP7 European Project Inter-Trust (grant agreement 317731), IoT6 (grant agreement no: 288445), and the grant from the Spanish ministry for education under the FPU program (AP2009-3981).

References

[1] B. Bayu, P. N. Mahalle, N. R. Prasad, and R. Prasad. Capability-based access control delegation model on the federated IoT network. In *Proc. of the 15th International Symposium on Wireless Personal Multimedia Communications (WPMC'12), Taipei, China*, pages 604–608. IEEE, September 2012.

[2] M. Castro, A. Jara, and A. Skarmeta. Smart Lighting Solutions for Smart Cities. In *Proc. of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA'13), Barcelona, Spain*, pages 1374–1379. IEEE, March 2013.

[3] D. Crockford. RFC 4627: The application/json Media Type for Javascript Object Notation (JSON). IETF RFC 4627, July 2006. `http://www.ietf.org/rfc/rfc4627.txt`.

[4] J. Dennis and E. V. Horn. Programming Semantics for Multiprogrammed Computations. *Communications of the ACM*, 9(3):143–155, 1966.

[5] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. RFC 2693: SPKI Certificate Theory. IETF RFC 2693, September 1999. `http://www.ietf.org/rfc/rfc2693.txt`.

[6] D. Ferraiolo, J. Cugini, and R. Kuhn. Role-based access control (RBAC): Features and motivations. In *Proc. of 11th Annual Computer Security Application Conference*, pages 241–248, 1995.

[7] V. Gungor, D. Sahin, T. Kocak, S. Ergüt, C. Buccella, C. Cecati, and G. Hancke. Smart Grid and Smart homes: Key Players and Pilot Projects. *Industrial Electronics Magazine, IEEE*, 6(4):18–34, December 2012.

[8] 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*, 58(5-6):1189–1205, September 2013.

[9] R. S. H. Istepanian, A. Jara, A. Sungoor, and N. Philips. Internet of things for M-health applications (IoMT). In *Proc. of the 1st AMA IEEE Medical Technology Conference on Individualized Healthcare, Washington, USA*. IEEE, March 2010.

[10] A. Jara, M. Zamora, and A. Skarmeta. An Internet of Things—based personal device for diabetes therapy management in ambient assisted living (AAL). *Personal and Ubiquitous Computing*, 15(4):431–440, April 2011.

[11] C. Jennings, J. Arkko, and Z. Shelby. Media Types for Sensor Markup Language (SENML). IETF Internet-draft (work in progress), October 2012. `http://tools.ietf.org/html/draft-jennings-senml-10`.

[12] M. Jones, J. Bradley, and N.Sakimura. JSON Web Token (JWT). IETF Internet-draft (work in progress), July 2013. `http://tools.ietf.org/html/draft-ietf-oauth-json-web-token-11`.

[13] S. Li, J. Hoebeke, F. V. den Abeele, and A. Jara. Conditional observe in CoAP. IETF Internet-draft (work in progress), June 2013. `http://tools.ietf.org/html/draft-li-core-conditional-observe-04`.

[14] J. Liu, Y. Xiao, and C. L. P. Chen. Authentication and Access Control in the Internet of Things. In *Proc. of the 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW'12), Macau, China*, pages 588–592. IEEE, June 2012.

[15] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad. Identity driven Capability based Access Control (ICAC) for the Internet of Things. In *Proc. of the 6th IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS'12), Bangalore, India*, pages 49–54. IEEE, December 2012.

[16] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad. Identity Establishment and Capability based Access Control (iecac) scheme for Internet of Things. In *Proc. of the 15th International Symposium on Wireless Personal Multimedia Communications (WPMC'12), Taipei, China*, pages 187–191. IEEE, September 2012.

[17] D. Miorandi, S. Sicari, F. Pellegrini, and I. Chlamtac. Internet of Things: Vision, Applications & Research Challenges. *Ad Hoc Networks*, 10(7):1497–1516, September 2012.

[18] T. Moses. Extensible Access Control Markup Language (XACML) Version 2.0, 2005.

[19] N. Ragouzis, J. Hughes, R. Philpott, E. Maler, P. Madsen, and T. Scavo. Security Assertion Markup Language (SAML) V2.0 Technical Overview, 2008.

[20] D. Recordon and D. Reed. Openid 2.0: a platform for user-centric identity management. In *Proc. of the 2nd ACM workshop on Digital identity management (DIM'06), Alexandria, USA*, pages 11–16. ACM, October-November 2006.

[21] E. Rescorla. RFC 2818: HTTP Over TLS. IETF RFC 2818, May 2000. `http://tools.ietf.org/html/rfc2818`.

[22] R. Román, J. Zhou, and J. López. On the Features and Challenges of Security and Privacy in Distributed Internet of Things. *Computer Networks*, 57(10):2266–2279, July 2013.

[23] J. Santa, M. A. Zamora-Izquierdo, A. J. Jara, and A. F. Skarmeta. Telematic platform for integral management of agricultural/perishable goods in terrestrial logistics. *Computers and Electronics in Agriculture*, 80:31–40, January 2012.

[24] L. Seitz, G. Selander, and C. Gehrmann. Authorization Framework for the Internet-of-Things. In *Proc. of the 14th IEEE International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'13), Madrid, Spain*, pages 1–6. IEEE, June 2013.

[25] Z. Shelby, K. Hartke, and C. Bormann. Constrained Application Protocol (CoAP). IETF Internet-draft (work in progress), June 2013. `http://tools.ietf.org/html/draft-ietf-core-coap-18`.

[26] E. Yuan and J. Tong. Attributed based access control (ABAC) for web services. In *Proc. of the 12th IEEE International Conference on Web Services (ICWS'05), Orlando, USA*. IEEE, July 2005.

[27] G. Zhang and W. Gong. The Research of Access Control Based on UCON in the Internet of Things. *Journal of Software*, 6(4):724–731, April 2011.

Author Biography

José L. Hernández-Ramos received the B.S. degree and the M.S. degree in 2012 from the University of murcia. Since 2013 he is a research fellow and Ph.D student in the Department of Information and Communications Engineering at the same university, where he has been participating in several European research projects. His main research interests are related to the application of novel security and privacy mechanisms in the Internet of Things.

Antonio J. Jara is the Vice-chair of the IEEE Communications Society Internet of Things Technical Committee, CTO and co-founder of the Smart Cities company viBrain Solutions, Assistant Prof. PostDoc at University of Applied Sciences Western Switzerland (HES-SO) from Switzerland, He did his PhD at the Intelligent Systems and Telematics Research Group of the University of Murcia (UMU) from Spain. He received two M.S. (Hons. - valedictorian) degrees. The first in Computer Science from UMU in 2009, where Master Thesis explored the "Internet of things in clinical environments", and a second M.S. Computer Science degree dealing with advanced networks and artificial intelligence from UMU in 2010. Master Thesis pertained to Mobility protocols for 6LoWPAN. He has been associated with the Department of Information and Communication Engineering, UMU, since 2007, where he has been working on several projects related to the WSNs (6LoWPAN and ZigBee) and RFID applications in building automation and healthcare. He is especially focused on the design and development of new protocols for security and mobility for Future Internet of things, which is the topic of my Ph.D. He has also carried out a Master in Business Administration (MBA). He has published over 100 international papers, As well, he holds one patent, and he has participated in several Projects about the Internet of Things.

Leandro Marín is Associate Professor at University of Murcia (UMU) in the Department of Applied Mathematics. He did his degree in Mathematics (Hons. - valedictorian) in UMU and the Ph.D. in Mathematics in the same university. He also did a M.Sc. in the University of Glasgow. His reseach is divided between the category theory and the most applied cryptography in cooperation with the Intelligent Systems and Telematics Research Group of the UMU, with more than 30 contributions on journals and international conferences. He has also worked as consultant in topics related with cryptography and security for international companies and developed cryptographic software for smart devices.

Antonio Skarmeta is the Vice-chair of the IEEE Communications Society Internet of Things Technical Committee. He received the M.S. degree in Computer Science from the University of Granada and B.S. (Hons.) and the Ph.D. degrees in Computer Science from the University of Murcia Spain. Since 2009 he is Full Professor at the same department and University. Antonio F. Gómez-Skarmeta has worked on different research projects in the national and international area, like Euro6IX, 6Power, Positif, Seinit, Deserec, Enable, Daidalos, ITSS6, and IoT6. His main interested is in the integration of security services at different layers like networking, management and web services. Associate editor of the IEEE SMC-Part B and reviewer of several international journals, he has published over 90 international papers and is member of several program committees.