

A Security Framework for Internet of Things

Imane Bouij-Pasquier¹✉, Anas Abou El Kalam¹, Abdellah Ait Ouahman¹,
and Mina De Montfort²

¹ UCA - ENSA, Marrakesh, MOROCCO
imane.pasquier@gmail.com

² Société ARTIMIA, Reims, France

Abstract. As we move towards the Internet of Things (IoT), the number of sensors deployed around the world is growing at a rapid pace. There is a huge scope for more streamlined living through an increase of smart services but this coincides with an increase in security and privacy concerns, therefore access control has been an important factor in the development of IoT.

This work proposes an authorization access model called SmartOrBAC built around a set of security and performance requirements. This model enhances the existing OrBAC (Organization-based Access Control) model and adapts it to IoT environments. SmartOrBAC separates the problem into different functional layers and then distributes processing costs between constrained devices and less constrained ones and at the same time addresses the collaborative aspect with a specific solution. We also apply SmartOrBAC to a real example of IoT and demonstrate that even though our model is extensive, it does not add additional complexity regarding traditional access control model.

1 Introduction

Today we are seeing a shift in our conception of Internet towards a global network of “smart objects”, which we can call the Internet of Things (IoT). This shift is expected to accelerate during the coming years [1,2] due to a fall in hardware costs, internet’s technological maturity and the swift development of communication technology. This will lead to a smooth assimilation of these smart objects into the Internet, which will in turn enable mobile and widespread access. Areas that are expected to be directly affected include healthcare [3,4], supply chain management [5], transport systems [6], agriculture and environmental monitoring [7,8], life at home and more, as we move towards “smart homes” [9–11] and the next generation of “smarter cities” [12].

This extension and proliferation of technology will certainly change our live, but will also present security and privacy challenges [13–15], since unexpected information leaks and illegitimate access to data and physical systems could have a high impact on our lives. Moreover, malicious modifications or denial of service may also cause damage in the context of IoT. This is why the implementation of an access control mechanism that respects both the character of and the

constraints on, smart objects in the IoT environment, is imperative. In this paper we address one of the most relevant security issues – authorization and access control – in the context of distributed, cross-domain systems that consist of resource constrained devices not directly operated by humans. In particular, we focus on the problem where a single constrained device is communicating with several other devices from different organizations or domains. Based on OrBAC [16] access control model, our “Smart OrBAC” proposal is specifically designed for IoT environments. It in fact takes the main features of IoT into account and facilitates a distributed-centralized approach where authorization decisions are based on local conditions, and in this way offers context-aware access control.

The reminder of the paper is organized as follows. Section 2 gives an overview of the literature and discusses the important access control models currently existing in the IoT environment. Afterwards Sect. 3 presents the background needed to understand our new work. The SmartOrBAC access control model is then detailed in Sect. 4 followed in Sect. 5 by a brief description of the implementation. Finally, in Sect. 6, we conclude the paper and present our perspectives.

The main contributions of this work can be outlined as follows:

- Abstraction layers design regarding the specificities of IoT devices.
- SmartOrBAC, our access control model for IoT.
- Collaborative protocol managing in IoT.
- Applying SmartOrBAC to an IoT case study and showing that it does not present additional complexity.

2 Related Work

Zhang and Gong proposed in [17] the UCON model taking into consideration flexibility and heterogeneity in an IoT distributed environment. However, UCON is a conceptual model only, and thus it does not give details on the implementation of the monitoring process. This approach is still not practical.

The CAPBAC model is implemented in a centralized approach in [18] where the proposed framework is based on a central Policy Decision Point (PDP) which handles authorization decisions. Whereas the implementation of capability-based access control in IoT is considered in [19] with an entirely distributed approach without intervention of central entities. The limits of both a purely centralized approach and fully distributed approach will be detailed later on in this paper (see 3.2 Main architectures for IoT access control).

The Capability-based Context-Aware Access Control (CCAAC) [20] is a delegation model based on a federated vision of IoT [21], where a central entity in each domain is in charge of authorizing a delegation request from a delegator, and making the decision about granting it to the delegate. However, this vision does not make use of technologies specifically designed for constrained highly context dependent environments such as IoT. Furthermore, the technical requirements in the constrained environment of the different actors involved in the proposed delegation mechanism are missing from this study.

Seitz et al. present in [22] an authorization framework based on XACML [23]. Evaluating XACML policies is too heavy-weight for constrained devices; therefore most of the authorization process is externalized. In order to convey the authorization decision from the external point to the device, an assertion is encoded in JSON [24] and is sent to the end-device (i.e., sensor or constrained device). The end-device takes responsibility for local conditions verification. However, this study does not give information about the central component involved neither about its management within the organization. Also, this proposal is bound to the use of XACML, which is not specifically designed for use in constrained devices.

3 Background

In this section we provide a brief description of some of the core concepts that make up our scheme. First of all, we give in this section an overview of the OrBAC access control model and its benefits over other commonly accepted models. We then propose an overview of the main approaches and trends to provide access control process in IoT scenarios based on the architecture taxonomy proposed in [25].

3.1 Organization-Based Access Control Model (OrBAC)

The Organization-Based Access control model (OrBAC) introduces the concept of organization as a structured group of active entities, in which subjects play specific roles. An activity is a group of one or more actions, a view is a group of one or more objects and a context is a specific situation.

Actually, the Role entity is used to structure the link between the Subjects and the Organizations. The Empower (org, r, s) relationship (or predicate) means that org employs subject s in role r . In the same way, the objects that satisfy a common property are specified through views, and activities are used to abstract actions.

In security rules, permissions are expressed as Permission (org, r, v, a, c), obligations and prohibitions are defined similarly. Such an expression is interpreted as: in the context c , organization org grants role r the permission to perform activity a on a view v .

As rules are expressed only through abstract entities, OrBAC is able to specify the security policies of several collaborating and heterogeneous organizations. Moreover, OrBAC takes the context (e.g., specific situations, time and location constraints) into account. However, despite the several advantages of OrBAC, it is not completely adapted to IoT. In particular, OrBAC is not able to manage collaboration-related aspects. In fact, as OrBAC security rules have the Permission (org, r, v, a, c) form, it is not possible to represent rules that involve several independent organizations. Furthermore, it is impossible to associate permissions to entities belonging to other partner-organizations (or to sub-organizations). As a result, if we can assume that OrBAC provides a framework for expressing

the security policies of several organizations, it is unfortunately only adapted to centralized structures and does not cover the distribution, collaboration and interoperability needs when it comes to cross-domain services as it is the case in IoT scenarios.

In order to overcome the limitations listed above, we suggest, on one hand, to extend OrBAC to include collaboration-related and context aware concepts, and on the other hand, we construct an IoT adapted framework with a new architecture articulated around four functional layers. The resulting framework is called “SmartOrBAC”.

3.2 Main Architectures for IoT Access Control

This section gives an overview of the most commonly used approaches to provide access control in IoT scenarios highlighting their main advantages and drawbacks:

- **Centralized Architecture:** The access control process is externalized into a central entity responsible for the authorization processing and thus, the end-devices (i.e., sensors, actuators) play a limited role and the access control process is located within a non-constrained entity. It follows that the use of standard security protocols normally used in the traditional Web is not restricted. Nonetheless, in IoT scenarios, contextual information is of great importance, while in a centralized architecture, access control decisions are not based on such local information related to the end-device.
- **Distributed Approach:** The access control process is located in the end-devices. An advantage of this approach is that end-devices act smartly, and are autonomous. Moreover it allows real time contextual information to become central to the authorization decision. However, this approach means that each device must be capable of handling authorization processes and having adequate resources which makes it inappropriate for resource-constrained devices.
- **Centralized-distributed Approach:** The end-devices participate partially in the access control decisions. This approach is motivated by the importance of taking into account the context of the end-device while making the decision. It allows the use of standard technologies to perform the authorization process. Nevertheless the transmission of the contextual information to a central entity may cause delays and the value acquired by the end-device will not be the same at the time of making the authorization decision.

In our proposal, we choose to design our access control based on the centralized-distributed approach. But unlike other proposals that use this approach, each separate group of components will have a central authorization engine (rather than just having one of these engines centrally performing all the authorization processes). The selection process that determines which entity will act as this engine depends on the contextual properties of the nodes in its group. The aim of this is to make the access control mechanism more time efficient by facilitating a smoother exchange of information between the end device

and the authorization engine (see Fig. 1). This vision is made possible by the fact that in a constrained environment, not all the devices are at the same level of constraint [27–31]. In almost every WSN, less constrained nodes exist, and thus the central authorization server in charge of an area can be implemented on one of them. For more understanding, the next section gives an overview of the different actors involved in the proposed architecture and their properties.

4 SmartOrBAC

This section provides a detailed description of the key aspects of our proposal. We begin with an explanation of the most relevant features of our abstraction layers design followed by an overview of the main aspects of our collaborative solution. Then we present our version of the distributed-centralized architecture and give a structured expression of the *context* concept. Finally we apply our proposal on a typical IoT healthcare scenario.

Before going into details, we first identify the following actors [26]:

- **Resource Server (*RS*):** an entity which hosts and represents a Resource that might contain sensor or actuator values or other information;
- **Resource Owner (*RO*):** the principal that owns the resource and controls its access permissions;
- **Client (*C*):** an entity which attempts to access a resource on a Resource Server;
- **Client Owner (*CO*):** the principal that owns the Client and controls permissions concerning authorized representations of a Resource.

Consequently, in a basic scenario, *C* wants to access *R* located on *RS*. It follows logically that, *C* and / or *RS* are constrained.

4.1 SmartOrBAC Abstraction Layers

The SmartOrBAC architecture proposes, among others, a model based on a partitioning of the access control process into functional layers depending on the capabilities offered on each one. This approach is directly inspired by the fact that each device is constrained to a different level; they are in fact not all uniformly constrained. Note that the term “constrained node” is used according to the RFC 7228 [27].

While processing access control related tasks each layer assists the one below when needed. Note that the authentication process details are out the scope of this study. Only authorization aspects are treated. Four layers are introduced:

Constrained Layer. One or both of *C* and *RS* are presumed to be located in a constrained node, but despite this, must perform access control related tasks. We thus consider that either of them may be unable to manage complex tasks while processing authorization requests. In addition, nodes do not always have

permanent network connectivity. That’s why both of C and RS are considered to be constrained layer actors.

In order to address the limitations present in this layer, a less constrained device is associated to each area of constrained devices. This centric entity is defined by the upper layer called less-constrained layer (see Fig. 1).

Less Constrained Layer. To relieve constrained layer actors from conducting computationally intensive tasks, another layer is introduced. Each group of constrained layer actors is bound to a less constrained layer actor that belongs to the same security domain (see Fig. 1). This link is configured by the entity in charge of the device (see below Organization layer). We call this central element the “Client Authorization Engine” (CAE), on the client side, and Resource Authorization Engine (RAE) on the resource side.

The CAE belongs to the same security domain as C . It assists C in determining if RS is an authorized source for R by obtaining authorization information and supporting C in handling the authorization process.

The RAE belongs to the same security domain as R and RS . It assists RS in determining the correct permissions of C on the requested resource R . RAE obtains authorization information and supports RS in handling the authorization process.

Organization Layer. In the real world, C and R are under the control of some physical entities. These entities are commonly called ROr “Resource Organization” and COr “Client Organization” (see Fig. 2). In order to keep close to

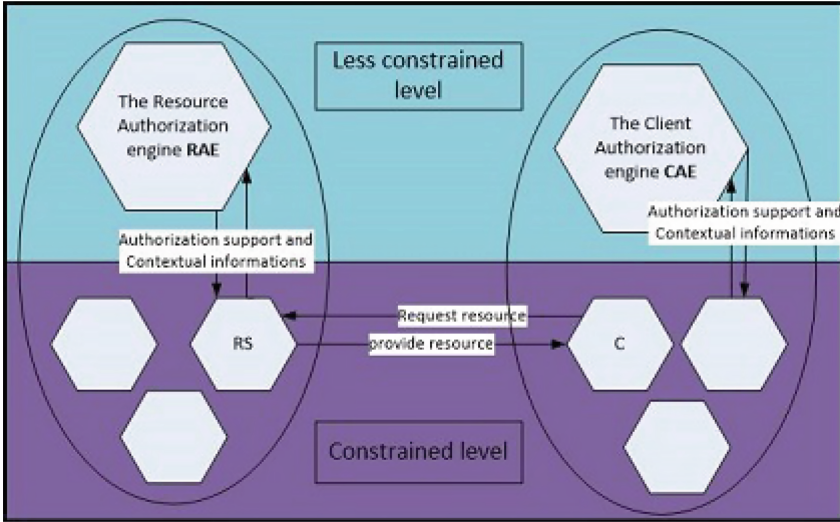


Fig. 1. Constrained and less constrained layers defined according to a centralised-distributed approach

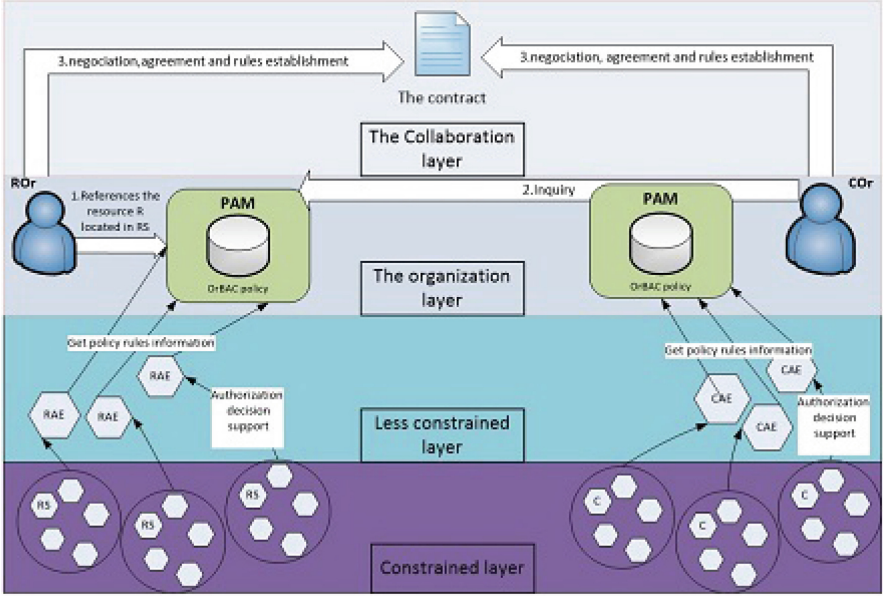


Fig. 2. Management of cross domain requirement in IoT environment

reality and to the OrBAC environment, we represent this entity by Organizations. Thus, each organization specifies the security policy for its devices and structures them in security domains.

The client organization *COr* is in charge of the entity proceeding to the resource request and thus, must specify security policies for *C*, including with whom *C* is allowed to communicate. This means that *COr* has to define authorized sources for a resource *R*. *COr* also configures *C* and *CAE* in order to make them belong to the same security domain.

The resource Organization *ROr* belongs to the same security domain as *R* and *RS*. *ROr* is in charge of *R* and *RS* and thus, must specify the authorization policies for *R* and decides with whom *RS* is allowed to communicate. That means that *ROr* has to configure if and how an entity with certain attributes is allowed to access *R*. *ROr* also configures *RS* and *RAE* in order to make them belong to the same security domain.

Subsequently, on the client side, *COr* defines authorized sources for *R*, and on the Resource side, *ROr* configures if and how an entity can access *R*. In order to do this, *ROr* and *COr* must have already agreed on the terms of such a service and on how to organize and structure this collaboration. An agreement is passed between the two entities before this interaction takes place (see below Collaboration layer: a cross domain access control).

Collaboration Layer: A Cross Domain Access Control. As seen above, the OrBAC access model does not handle the collaborative interaction

aspects. To overcome this limitation, we suggest enhancing OrBAC with new collaboration related concepts. This issue is addressed at the organization layer, by making a prior agreement between the involved organizations (as shown in Fig. 2) where the access rules to a given resource are jointly defined according to the OrBAC format by organizations that interact.

In order to manage this new agreement, we will use the entity, located in the Organization layer, called Principal Authorization Manager “PAM”. From the *RS* point of view, this agreement, which is interpreted in terms of access rules, will be treated just like all the other rules concerning local interactions. The complexity of the external interaction authorization management is hidden from the end constrained device, which keeps the same authorization processing no matter the nature of the client. This abstraction is made possible by the establishment of a fourth layer that manages the cooperation between different organizations.

Basically, SmartOrBAC begins with the publication and negotiation of collaboration rules as well as the corresponding access control rules. First, each organization determines which resources it will offer to external partners, and then references them into the *PAM*. At this point, other organizations can contact it to express their wish to use these specific referenced resources. To do that, the *COr* and the *ROr* negotiate and come to an agreement concerning the use of the resource *R*. Then, they establish a contract and jointly define security rules concerning access to *R*. The *COr*’s and *ROr*’s exchange format and the contract aspect will be discussed in a future paper. In the rest of this section, let us focus on access control rules. These rules are registered – according to an OrBAC format – in the *PAM* of both organizations. Parallel to this, *COr* creates locally a “virtual resource” called *R_image* which represents (the remote) *R* in the client organization side. Then *COr* adds a rule in its OrBAC policy base to define which entities can invoke *R_image* (see Figs. 2 and 3).

4.2 Enhancing OrBAC for Context Awareness

Unlike traditional services where the concept of context is limited to a finite set of use cases, in the IoT environment, the concept is getting wider by taking on an ambient character in order to allow services taking into account the contextual information collected in real time by the different sensors [20]. The Context used in defining the SmartOrBAC rule is a set of contexts (C_{Set}) with different types (C_{Type}). The type of context can be a concrete property such as time or location, but also security related context such as authentication and trust level. In order to take the context into account in the access control decision, each of the context types has to be evaluated with a certain constraint (C_{Const}).

The overall context definition in SmartOrBAC can be expressed with the following notation:

$$TYPES \in \{authLevel, trustLevel, time, location \dots\} \quad (1)$$

$$C_{Set} = \{C_{Type(1)}, C_{Type(2)}, \dots, C_{Type(n)}\} \quad (2)$$

$$\begin{aligned} &\text{Where } C_{Type(1)}, C_{Type(2)}, \dots, C_{Type(n)} \text{ in TYPES} \\ &C_{Const} = \langle C_{Type(i)} \rangle \langle OP \rangle \langle VALUE \rangle \end{aligned} \quad (3)$$

where OP is a logical operator, i.e., $OP \in \{>, <, \geq, \leq, =, \neq\}$, and VALUE is a specific value of C_{Type} . Finally, we define C as a set of context constraints C_{Const}

$$C = \{C_{Const(1)}, C_{Const(2)}, \dots, C_{Const(n)}\} \quad (4)$$

4.3 Scenario

In order to illustrate SmartOrBAC, we apply the different concept detailed above in a typical healthcare scenario [28,29].

Assume that John, a man with a heart condition, has opted for an assisted living service that is provided by a medical center. John uses a device that monitors his heart rate and his position; his home is also equipped with multiple sensors and actuators (temperature sensor, humidity sensor, luminosity sensor...). In case of a cardiac arrest the heart monitor automatically sends an alarm to an emergency service, transmitting John's current location.

A doctor, who monitors John's health remotely from the medical center, receives an alarm that John has fainted. An ambulance is instructed to go to assist John. A smart driving application is used by the ambulance to reach John's home as quickly as possible.

The situation requires the interaction of the following organizations: "smart home of John", "the medical center", "the ambulance", and "the police department for traffic jams monitoring".

First of all, each of these organizations determines which device's re-sources it will offer to external partners. At this stage, we find in the *PAM* of John's smart home's organization resources such as the heart monitor resource. The medical center organization makes an inquiry to the *PAM*. As soon as the target resource is found, the negotiation phase begins between the *ROR* of the smart home and the *COR* of the medical center. The resulting contract is then transcript in terms of authorization rules regarding the OrBAC format for both of the medical center and smart home of John. More precisely, if the agreement between the two organizations is: "Assigned doctor from medical center have the permission to remotely actuate the implanted cardioverter defibrillator from the heart monitor device in the heart attack emergency context", the *ROR* of Smart home should:

- have (or create) a rule that grants the permission to a certain role (e.g., *Doctor*) to actuate the heart monitor: *Permission(smart home, Doctor, vital equipment, Actuating, Cheart_attack_Emergency)*. Note that, from John's smart home's point of view, every user playing the "Doctor" role will have this permission;
- create a "virtual user" noted *v_user_doctor* that represents the medical center for its use of the implanted cardioverter defibrillator (see Fig. 3);
- add the following *Empower(smart home, v_user_doctor, Doctor)* association to its rule base. This rule grants the user medical center's doctor the right to play the Doctor role.

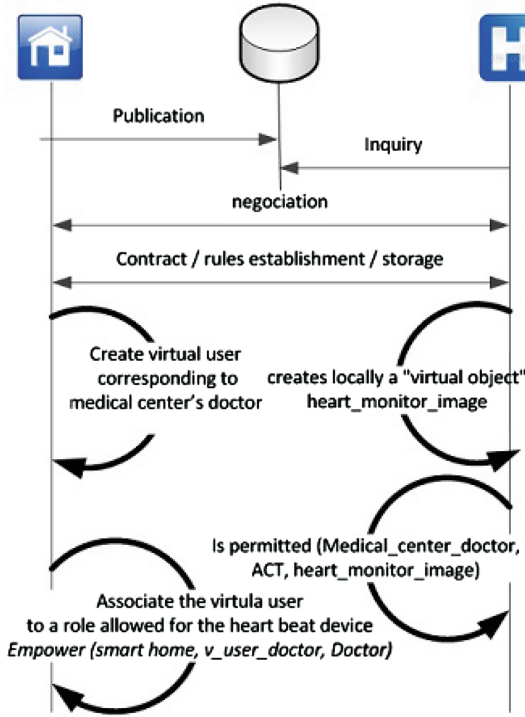


Fig. 3. Virtual user and virtual object in SmartOrBAC

In parallel, the *COr* of the medical center creates locally a “virtual object” *heart_monitor_image* which represents the (remote) implanted device (the resource made available by John’s Smart Home), and adds a rule in its OrBAC base to define which of the medical center’s roles can invoke *heart_monitor_image* to use the real heart monitor. Let’s assume that the assisted living dispositive is a set of different devices (sensors and actuators) with different capabilities. We also assume that the specific device *RS* of heart monitoring that the medical center tries to access is located in the constrained layer, such as the client device *C* used by the doctor in the medical center. The link between the *RS* and its corresponding *RAE* located in the less constrained layer has already been configured by the *ROr* of John’s smart home. The same applies for the *CAE* and *C* that have been already configured by the *COr* of medical center.

Before the doctor’s device *C* in the medical center sends an actuating request to the heart monitoring device *RS*, it asks the corresponding *CAE* in the medical center for assistance in order to determine if the local image of *RS* (*heart_monitor_image*) is an authorized source.

At this moment, *CAE* starts evaluating the authorization policy rules, using as object the *heart_monitor_image*. Note that at this level, the external nature of the heart monitor device is unknown. Then, if information about policy rules is needed, a request is sent to the *PAM* of the medical center. Once this process

is completed, if RS is an allowed source, an actuating request is directly sent to the heart monitoring device.

Once the request is received, the authorization decision process begins on the smart home organization side. For that, the device sends an authorization process request, with contextual information, to the corresponding RAE in John's smart home. The latter evaluates the authorization decision regarding authorization rules in John smart home's PAM especially those detailed above where the subject is v_user_doctor . The result is sent to RS which, in turn, sends an access response to the doctor's device.

5 Implementation

The transmissions between the different entities included in our Framework (C/RS , C/CAE , RS/RAE) are done via the CoAP [30] protocol (Constrained Application Protocol), which is a specialized Web transfer protocol that is intended for use in resource-constrained Internet devices. Like HTTP, CoAP is based on the wildly successful REST model: servers make resources available under a URL, and clients access these resources using methods such as GET, PUT, POST and DELETE.

Since the XML representation is too verbose for efficient transmission over limited channels, we use JSON-based notation for our authorization requests and responses. In fact JSON [24] (JavaScript Object Notation) is a lightweight data-interchange format that efficiently reduces the size of the transmitted messages between C and RS devices and optimizes the processing time.

The device part of our framework (especially C and RS) was implemented on an example platform: the Arduino Mega 2560 board³. This board features a 16 MHz processor, 256 kB of Flash Memory, 8 kB of SRAM, and 4 kB of EEPROM. We choose this board in order to test our approach on the lowest performance of the end constrained devices. The board was programmed in Java using a custom implementation of the CoAP protocol stack and the assertions were wrapped in JSON format using the standard Java API (javax.json.*).

6 Conclusion

Our SmartOrBAC access model is specifically designed for the IoT environment and it is conceived through an abstraction layer design that makes use of a deep understanding of the IoT paradigm as it is used in the real world. For these smart services, contextual information is a leading element in decision making therefore only a real-time consideration of this information will achieve smartness. For this reason, we enhanced the "context" notion (originally present in OrBAC) in order to fit the IoT requirements.

Understanding that users belonging to an organization need to dynamically access resources controlled by other organizations we also extended our model with specific collaborative mechanisms where the same OrBAC security policy can be used for local as well as external access. In this way, SmartOrBAC

improves the management of the security policy and reduces considerably its complexity.

In our future work, we will explore how to make the SmartOrBAC model more effective and we will go deeper in the study of the negotiation process and the e-contract format. 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 assertions.

References

1. European Commission: Internet of things in 2020 road map for the future. Working Group RFID of the ETP EPOSS, Technical report (2008)
2. Guillemin, P., Friess, P.: Internet of things strategic research roadmap. The Cluster of European Research Projects, Technical report (2009)
3. Istepanian, R.S.H., Jara, A., Sungoor, A., Philips, N.: Internet of things for M-health applications (IoMT). In: Proceedings of the 1st AMA IEEE Medical Technology Conference on Individualized Healthcare, IEEE, Washington, USA (2010)
4. Jara, A., Zamora, M., Skarmeta, A.: An internet of things-based personal device for di-abetes therapy management in ambient assisted living (AAL). *Pers. Ubiquit. Comput.* **15**(4), 431–440 (2011)
5. Chaves, L.W.F., Decker, C.: A survey on organic smart labels for the internet-of-things. In: 2010 Seventh International Conference on Networked Sensing Systems (INSS), pp. 161–164 (2010)
6. Santa, J., Zamora-Izquierdo, M.A., Jara, A.J., Skarmeta, A.F.: Telematic platform for integral management of agricultural/perishable goods in terrestrial logistics. *Comput. Electron. Agric.* **80**, 31–40 (2012)
7. Burrell, J., Brooke, T., Beckwith, R.: Vineyard computing: sensor networks in agricultural production. *IEEE Pervasive Comput.* **3**(1), 38–45 (2004)
8. Radu, V.: Stochastic Modeling of Thermal Fatigue Crack Growth. *Applied Condition Monitoring*, vol. 1. Springer, Heidelberg (2015)
9. Gungor, V., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., Hancke, G.: Smart grid and smarthomes: key players and pilot projects. *IEEE Ind. Electron. Mag.* **6**(4), 18–34 (2012)
10. Kovatsch, M., Weiss, M., Guinard, D.: Embedding internet technology for home automation. In: 2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–8. Bilbao, Spain, IEEE (2010)
11. Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y., Jansen, E.: The gator tech smart house: a programmable pervasive space. *Computer* **38**(3), 50–60 (2005)
12. Castro, M., Jara, A., Skarmeta, A.: Smart lighting solutions for smart cities. In: Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA 2013), Barcelona, Spain, pp. 1374–1379. IEEE (2013)
13. Miorandi, D., Sicari, S., Pellegrini, F., Chlamtac, I.: Internet of things: vision, applications & research challenges. *Ad Hoc Netw.* **10**(7), 1497–1516 (2012)
14. Sundmaeker, H., Guillemin, P., Friess, P., Woelffle, S.: Vision and challenges for realising the internet of things. *Eur. Comm. Inf. Soc. Media* (2010)
15. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. *Comput. Netw.* **54**(15), 2787–2805 (2010)

16. Kalam, A.A.E., Baida, R.E., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Mieke, A., Saurel, C., Trouessin, G.: Organization based access control. In: 4th IEEE Workshop on Policies for Distributed Systems and Networks (pOLICY). pp. 120–134. Italy, 4–6 July 2003
17. Zhang, G., Gong, W.: The research of access control based on UCON in the internet of things. *J. Softw.* **6**(4), 724–731 (2011)
18. Gusmeroli, S., Piccione, S., Rotondi, D.: A capability-based security approach to manage access control in the internet of things. *Math. Comput. Modell.* **58**(5–6), 1189–1205 (2013)
19. Hernandez-Ramos, J., Jara, A.J., Marin, L., et al.: Distributed capability-based access control for the internet of things. *J. Internet Serv. Inf. Secur.* **3**(3/4), 1–16 (2013)
20. Bayu, B., Mahalle, P.N., Prasad, N.R., Prasad, R.: Capability-based access control delegation model on the federated IoT network. In: Proceedings of the 15th International Symposium on Wireless Personal Multimedia Communications (WPMC 2012), Taipei, China, pp. 604–608. IEEE (2012)
21. Prasad, R. (ed.): My personal Adaptive Global NET (MAGNET). Signals and Communication Technology Book. Springer, Netherlands (2010)
22. Seitz, L., Selander, G., Gehrmann, C.: Authorization framework for the internet-of-things. In: 14th IEEE International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2013), Madrid, Spain, pp. 1–6. IEEE (2013)
23. Moses, T.: Extensible Access Control Markup Language (XACML) Version 2.0 (2005)
24. Crockford, D.: RFC 4627: The application/json media type for javascript object notation (2006)
25. Roman, R., Zhou, J., Lopez, J.: On the features and challenges of security and privacy in distributed internet of things. *Comput. Netw.* **57**(10), 2266–2279 (2013)
26. Gerdes, S., Seitz, L., Selander, G., Bormann, C.: An architecture for authorization in constrained environments. IETF, Draft (2015)
27. Bormann, C., Ersue, M., Keranen, A.: Terminology for constrained-node networks. IETF RFC 7228 (2014)
28. Memon, M., Wagner, S.R., Pedersen, C., Beevi, F., Hansen, F.O.: Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. *Sensors* **14**, 4312–4341 (2014)
29. Dohr, A., Modre-Opsrian, R., Drobics, M., Hayn, D., Schreier, G.: The internet of things for ambient assisted living. In: 2010 Seventh International Conference on Information Technology: New Generations (ITNG), pp. 804–809 (2010)
30. Shelby, Z., Hartke, K., Bormann, C.: Constrained application protocol (CoAP) IETF RFC 7252 (2014)
31. Mainetti, L., Patrono, L., Vilei, A.: Evolution of wireless sensor networks towards the internet of things: a survey. In: 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), IEEE (2011)

Cryptology and Network Security

14th International Conference, CANS 2015, Marrakesh,
Morocco, December 10-12, 2015, Proceedings

Reiter, M.; Naccache, D. (Eds.)

2015, X, 257 p. 41 illus. in color., Softcover

ISBN: 978-3-319-26822-4