Implementing Policy Rules in Attributes Based Access Control with XACML within a Cloud-Enabled IoT Environment

Fatima Sifou
LRIT Laboratory, Faculty of Sciences, Mohammed V University, Rabat, Morocco
Email: fatimasifou@gmail.com

Feda AlShahwan[1], Mbarek Marwan[2], Adra Hammoud[3], and Ahmed Hammouch[3]
[1]Electronic Engineering Departments, Public Authority for Applied Education and Training, Kuwait
[2]LTI Laboratory, ENSA, Chouaib Doukkali University, El Jadida, Morocco
[3]LRIT Laboratory, Faculty of Sciences, Mohammed V University, Rabat, Morocco
Email: fa.alshahwan@gmail.com; marwan.mbarek@gmail.com; adrahammoud@gmail.com; hammouch_a@yahoo.com

***Abstract*—**The Internet of Things (IoT) extends internet connectivity to a wide range of smart devices. However, battery autonomy, computational capability and storage capacity are major technology challenges that hinder increased implementation and adoption. Although the integration of the Internet of Things (IoT) with Cloud Computing is considered as a highly promising solution in overcoming these bottlenecks, it raises security concerns, especially access control. Recently, a variety of access control models have been developed to help protect confidential information and restrict access to sensitive data. Because of its flexibility and scalability, the consensus is that the Attribute Based Access Control (ABAC) is the most appropriate model in a dynamic environment. In the context of IoT, the ABAC model has the ability to enforce data privacy and ensure a secure connection between IoT devices and cloud providers. One of the core components of the ABAC model is access policies, these are used to deny or allow user' requests. To achieve that, an access policy language is required to implement policy rules in ABAC model. In this study, we propose a method based on EXtensible Access Control Markup Language (XACML) to prevent all unauthorized access to remote resources. This policy language is a particularly efficient and appropriate technique within a context of IoT due to its compatibility with heterogonous platforms.

***Index Terms*—**cloud computing, internet of things, cloud, ABAC model, XACML language, security policy tool

I. Introduction

The Internet of Things (IoT) is a technology trend based on a software-powered objects interconnected with each other and to the internet in order to reach a common goal within a global network. It refers to "a world-wide network of interconnected objects uniquely addressable based on standard communication protocols" whose point of convergence is the Internet.

IoT's interoperability and heterogeneity has the potential to change our lifestyle and business for the better, because of its interoperability and heterogeneity. However, this new concept still suffers from limitations in technologies such as storage capacity, processing power etc. Mitigating these challenges has made the integrating of IoT devices with Could Computing a necessity. Cloud Computing is the most efficient technology to bridge the gap created by these IoT issues.

Over the past decade, Cloud Computing has become an integral part of the internet ecosystem. It enables access to digital resources from anywhere and at any time. Moreover, Cloud Computing has conveniently scalable virtually unlimited storage and processing power [1], this has led researchers to the obvious conclusion that the combined use of these two new paradigms is the best way to overcome IoT limitations.

The integration of Cloud Computing and the Internet of Things (IoT) known as CEIoT in our previous work [2] brought unlimited benefits, but it also brought many security and privacy issues such as access control. In general, access control refers to an approach or a security technique that means system administrators can grant, limit or deny access to system resources. A significant volume of research has been proposed in access control models such as DAC, MAC, RBAC and ABAC.

Technically, each model has its related limitations regarding security demands. However, it is important to realize that ABAC is the most appropriate access model in terms of flexibility, reliability, and support scalability [3]. Attribute Based Access Control (ABAC), as inferred by its name uses the attributes as building blocks in a structured language that defines access control rules and describes access requests [4]. It is worth noting that, one of the essential elements to any access control model is access control policies. These policies can be implemented using an access policy language Extensible Access Control Markup Language (XACML) as an example.

Manuscript received June 15, 2019; revised December 11, 2019.
doi:10.12720/jcm.15.1.107-114

II. Cloud-Enabled Internet of Things

Cloud Computing and IoT are two different technologies. Each has its advantages and limitations. IoT is characterized by widely-distributed devices with limited processing capabilities and data storage; these devices can be connected, utilized remotely. Obviously, the connection to Cloud resources would address some IoT limitations, since these two technologies are complementary [5].

This has led to a natural merging which can partially resolve most of the issues faced by IoT. Hence, CEIoT is changing the current and future environment of internetworking services [6].

Suppose that Cloud' characteristics are A, IoT characteristics are B, and Cloud Enabled characteristics are C, we can write this formula:

$$A \cap B = \emptyset \qquad A \cup B = C$$

A. CEIoT Benefits

The Cloud can also benefit from IoT by developing its limits with real world objects in a more dynamic and distributed way, and providing new services for billions of devices in different real life scenario [5], [6]. Likewise, CEIoT has advantages, and has several benefits:

Economic benefits: computing and storage resources paid for only as used reducing costs and potentially improve earnings.

Safe: critical and important data stored in the Cloud can be accessed by any device or machine and remotely recover them.

Efficient growth: the capacity of resources needed in the Cloud is used.

Elastic and scalable: resources used can be increased and reduced according to the business demand.

Avoiding downtime or delay: in case of a node failing, its load is taken up by the second Cloud node, so the business site is never down.

Disaster recovery: big business can afford additional IT resources for disaster management.

Environmental compatibility: cloud conserves energy and provides efficient technical solutions to the business.

Given those benefits, CEIoT had been applied in several domains as discussed below [7]-[9].

B. Application's Domain

Healthcare: CEIoT improves the quality and effectiveness of service, bringing high value for patients with chronic conditions. It implements smart features into medical devices and related software systems, keeping the domain innovative. The CEIoT provides many services in the healthcare field.

Logistics: CEIoT can help mainstream logistics systems to deal automatically with complexity and changes. It provides new interesting scenarios and allows the easy and automated management of flows of goods between the point of origin and the point of consumption, in order to meet specific requirements expressed in terms of time, cost or means of transport [5]. CEIoT do a remarkable job in terms of reducing core management complexities which are problematic for many companies across sectors.

Energy: cloud computing and the IoT can work together effectively to provide consumers with smart management of energy consumption. CEIoT solutions have come up with a practical processing formula to compensate for the increasing sophistication of the energy distribution networks and achieve real-time visibility of the consumption process (e.g. smart meters, smart appliances, smart lighting, remote infrastructure maintenance, renewable energy resources, smart grid asset monitoring) [10].

Home and buildings: CEIoT applications transform our everyday objects into information appliances by connecting them to internet so as to remotely monitor their behavior (e.g. electrical power distribution, consumption of water, gas emissions, safety systems, lighting, heating and air conditioning).

Mobility and transportation: several existing challenges in this field have been solved thanks to CEIoT applications, which bring business benefits, such as increasing road safety, reducing road congestion, managing traffic and parking, performing warranty analysis and recommending car maintenance or repairs [11].

Media: the CEIoT is an incredible opportunity for Telco'-s operational to reduce costs by applying IoT technology for software defined networking and network function virtualization. Fig. 1 below illustrates the different CEIoT domains of application.

Fig. 1. CEIoT application domains

C. CEIoT Issues

As CEIoT provides several benefits and promotes the improvement of many applications, the merging of those paradigms results in many issues, challenges and research problems.

These challenges are discussed in detail in our previous work [2].

Heterogeneity: by merging Cloud and IoT, a large number of heterogeneous devices, platforms, services and

operating systems can be synchronized used for new applications.

Big data: CEIoT produces a huge amount of data, which then requires particular attention to transfer, storage, access and processing.

Large scale: achieving the required computational capability and storage capacity is becoming difficult in new applications, and the IoT devices have to deal with connectivity issues and latency dynamics.

Performance: CEIoT applications initiate specific performance and Quality of Service (QoS) at different levels including communication, computation and storage aspects.

Monitoring: CEIoT applications are impacted by volume, variety and velocity characteristics of IoT.

Privacy and security: because of their distributed architecture, CEIoT systems are exposed to several potential attacks and have common vulnerabilities. Another important issue that has not yet been resolved is how to provide appropriate authorization rules and policies while ensuring that only authorized users have access to the sensitive data. This challenge is called access control and it will be discussed in more details in the following section. Fig. 2 below presented the CEIoT issues discussed above.

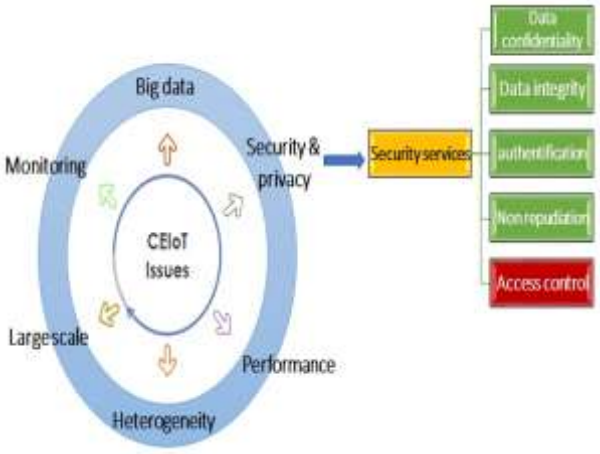

Fig. 2. CEIoT issues

In this work, we propose the implementation of Policy Rules in Attributes Based Access using XACML as the best choice to build access control policies into CEIoT in order to make access decisions. The remainder of this paper is structured as follows. Section 2 presents an overview of Cloud–Enabled Internet of Things (CEIoT). Section 3 discusses an access control within CEIoT. In section 4 the proposed Policy Based on ABAC model is presented. In section 5 we end this study with some concluding remarks. Your goal is to simulate the usual appearance of papers in the. We are requesting that you follow these guidelines as closely as possible.

III. Access Control in Cloud-Enabled IOT

Access Control is an important component to take into account in terms of security and privacy within a dynamic environment such Cloud-Enabled IoT. It makes sure that only trusted users can access the resource. Thus, it governs the actions that maybe taken on objects. In the context of CEIoT, the Cloud provider will offer access to resources such as file, record, data and so forth, to ensure that only the correct subject (IoT devices or device's owner) can access its services. The access control system consists of four elements (e.g. Subject, Object, Action and Permission). These elements are described below.

Subject: the entity that requests data access (e.g. Users or IoT devices).

Object: the entity can be accessed (e.g. File, Computer, Database…).

Action: the operation to be performed on the object (e.g. Write, Delete, Update…).

Environment: the context in which the action will be performed on an object (e.g. time, location…).

Authentication and Authorization are the main aspects in access control. Authentication verifies who will be given access to the resource, while Authorization defines the rights and privileges of the user identified and what actions they are authorized to undertake [12]. In this regard, researchers and technologists have proposed various access control mechanisms. Ouaddah *et al*. [13] presented a survey of access control models in IoT. An overview of access control mechanisms was proposed in our previous work [3]. According to the research, ABAC also named Policy Based Access Control is the most appropriate model in a dynamic environment due to its flexibility, scalability and many other characteristics. Making a decision in favor of the ABAC model is based on evaluating policy rules against user 'attributes. Our contribution is implementing these rules using XACML language. The following section will offer more details.

A. Design a Security Policy Using XACML

In this section, we discuss the motivation of our proposed method and an overview of some basic concepts used in this method.

Extensible Access Control Markup Language (XACML) is a standard language widely adopted in the field of information security especially access control. It uses XML as the internal format. It describes an access control policy language (e.g. ABAC), as well as access control decisions (request/response) language [14]. XACML provides a high level of granularity and the adaptability necessary to express and enforce security policy. Practically, an XACML policy, it is composed of several top-level elements such as PolicySets, Policies, and Rules. PolicySets is the highest element of an XACML policy hierarchy. It can contain any number of policies or policy set. Policies in turn can that contain a set of rules. The rules define the desired effect, either of Permit or Deny [15].

A typical XACML engine is built on top of six basic components [16], [17]:

- Policy Administration Point (PAP): creates, stores, and manages policy or a policy set in order to make them available for the PDP.
- Policy Enforcement Point (PEP): performs access control request by enforcing authorization decisions made by PDP.

- Policy Information Point (PIP): retrieves attributes values about subject, object, action, and environment. This makes these attributes available for the PDP.
- The Context Handler: converts request and responses between native formats and the XACML canonical representation and coordinates, with the PIPs, gathering of the required attributes values.
- Policy Decision Point (PDP): Evaluates the request against the applicable access control policies and returns de final decision to the PEP.

The functioning of these policies is further detailed in Fig. 3.

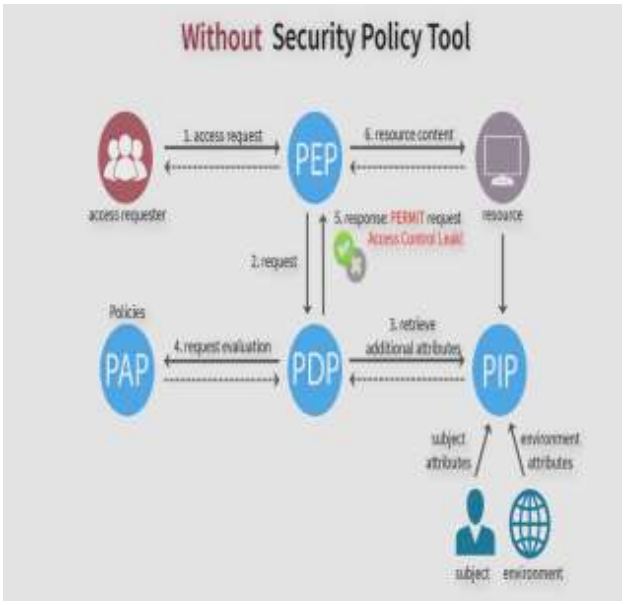

Fig. 3. Policy architecture in XACML [16]

B. Security Policy Tool

Actually, there are several tools to design XACML policy like ALFA, Security Policy Tool (SPT), etc. In this work, we opt for SPT as a tool for editing, modeling, testing, and verifying security policies to prevent access control leakage. It allows the user to verify the XACML rules in terms of security before operations, test the policies to help the user prevent access control leaks as a result of these errors and save time and cost in development of XACML policies. In this regard, it automatically converts any access control model into XACML policies. Thanks to this tool we can ensure that the access control policies are fully safe before implementing them into an Access Control System [19].

IV. Implementing the Proposed Policy Based on ABAC Model

The ABAC model is the most appropriate model within a dynamic environment such as CEIoT due to its flexibility and scalability. In case of access requests, the ABAC engine's decision is based on users, resource, environment attributes and set of policies. Our contribution is implementing ABAC rules. To achieve this aim we chose XACML as a language to evaluate requests for resources according to rules defined in policies, because it is one of most popular methods for managing data access.

A. Used Tool

In this study, we use SPT to implement the proposed security policy. In the SPT tool, the access permissions are granted to a request using a set of policies. In our implementation, we used this tool to evaluate, test and validate the access control policies, in order to ensure data privacy in a system. Besides, it is an efficient access control for testing and verifying security policies. More importantly, it provides an interface to design XACML access control policies more efficiently [19] [20], as shown in Fig. 4.

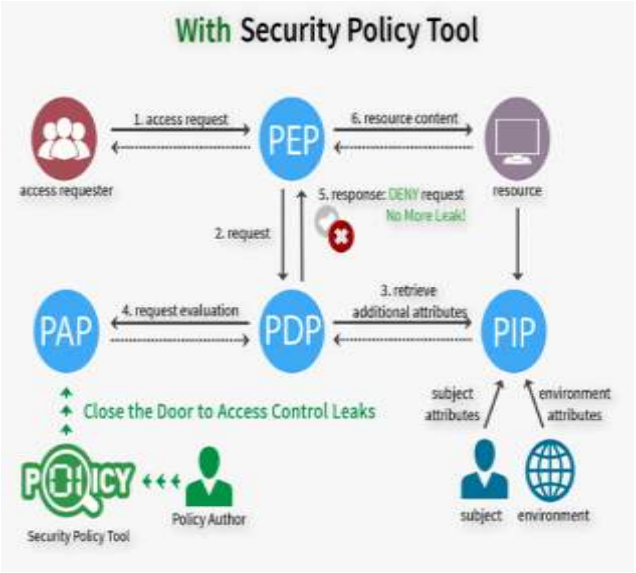

Fig. 4. Policy architecture in XACML with security policy tool [16]

B. Simulation and Results

This section is intended for simulation and results. We simulate a University Library as a case study.

Fig. 5. Security policy life cycle

Test Case (Hospital Test Case): In our implementation, we treat scenario, a Doctor, Manager, and Patient as subjects who request access to Patient's Medical Records, Personal Information, and Private Notes as objects in order to execute the View an Add actions. These actions are limited by an environment and condition attribute in our study. We suppose time as an attribute, allowing the subject access to resources for a limited, predefined time under an emergency condition. Implementing policy rules in a security policy tool requires four steps as depicted in Fig. 5[21].

1) Setting up the policies

The Hospital Test case contains three policies (Doctor Policy, Manager Policy, and Patient Policy). Each policy includes several attributes values. The Doctor Policy attributes are: Dentist, Gynecologist, and Psychologist. The Manager Policy has one attribute which is Hospital Manager. The Patient Policy has two attributes which are PatientA and PatientB. After defining different attributes and attributes values the setting up of these policies is shown as below in Fig. 6.

Fig. 6. Test case Interface

2) Modeling the policies

After the setting up of the policies, we can model them by entering the rules. Each policy has various rules as illustrated in the list below.

DoctorPolicy:

(Subject = Any Value & Doctor = Dentist, View&Add, PatientA_MedicalRecords, Time= 08:00:00 am to 05:00:00 pm) →Permit

(Subject = Any Value & Doctor = Dentist, Any Action, PatientB_MedicalRecords, Any Value) → Deny

(Subject = Any Value & Doctor = Dentist, Any Action, PatientA_PersonalInfo&PatientB_PersonalInfo, Any Value) →Deny

(Subject = Any Value & Doctor = Dentist, Any Action, PatientA_PersonalInfo, Time= 08:00:00 am to 05:00:00 pm, Emergency=True) →Permit

(Subject = Any Value & Doctor = Dentist, Any Action, PatientA_PrivateNotes, Any Value) →Permit

(Subject = Any Value & Doctor = Dentist, Any Action, PatientB_PrivateNotes, Any Value) →Deny

(Subject = Any Value & Doctor = Gynecologist, View&Add, PatientB_MedicalRecords, Time= 08:00:00 am to 05:00:00 pm) →Permit

(Subject = Any Value & Doctor = Gynecologist, Any Action, PatientA_MedicalRecords, Any Value) → Deny

(Subject = Any Value & Doctor = Gynecologist, Any Action, PatientB_PersonalInfo&PatientA_PersonalInfo, Any Value) →Deny

(Subject = Any Value & Doctor = Gynecologist, Any Action, PatientB_PersonalInfo, Time= 08:00:00 am to 05:00:00 pm, Emergency=True) →Permit

(Subject = Any Value & Doctor = Gynecologist, Any Action, PatienB_PrivateNotes, Any Value) →Permit

(Subject = Any Value & Doctor = Gynecologist, Any Action, PatientA_PrivateNotes, Any Value) →Deny

(Subject = Any Value & Doctor = Psychologist, View&Add, PatientA_MedicalRecords&PatientB_MedicalRecords, Time= 08:00:00 am to 05:00:00 pm) →Permit

(Subject = Any Value & Doctor = Psychologist, View, PatientA_PersonalInfo&PatientB_PersonalInfo, Time= 08:00:00 am to 05:00:00 pm) →Permit

(Subject = Any Value & Doctor = Psychologist, Any Action, PatientA_PrivateNotes&PatientB_PrivateNotes, Time= 08:00:00 am to 05:00:00 pm) →Permit

ManagerPolicy:

(Subject = Any Value &Manager = HospitalManager, Any Action,PatientA_MedicalRecords&PatientB_MedicalRecords)→Deny

(Subject = Any Value & Manager = HospitalManager, View, PatientA_PersonalInfo&PatientB_PersonalInfo, Time= 08:00:00 am to 05:00:00 pm)→Permit

(Subject = Any Value & Manager = HospitalManager, Any Action, PatientA_PrivateNotes&PatientB_PrivateNotes)→Deny

PatientPolicy:

(Subject = Any Value & Patient = PatientA, View, PatienA_MedicalRecords, Any Value)→Permit

(Subject = Any Value &Patient = PatientA, Add, PatientA_MedicalRecords)→Deny

(Subject = Any Value & Patient = PatientA, View&Add, PatientA_PersonalInfo Any Value)→Permit

(Subject = Any Value & Patient = PatientA, Any Action, PatientA_PrivateNotes, Any Value)→Deny

(Subject = Any Value & Patient = PatientB, View, PatientB_MedicalRecords, Any Value)→Permit

(Subject = Any Value &Patient = PatientB, Add, PatientB_MedicalRecords)→Deny

(Subject = Any Value & Patient = PatientB, View&Add, PatientB_PersonalInfo, Any Value)→Permit

(Subject = Any Value & Patient = PatientB, Any Action, PatientB_PrivateNotes, Any Value)→Deny

The modeling of the Doctor Policy, Manager Policy, and Patient Policy after entering their rules is depicted in Fig. 7, Fig. 8, and Fig. 9.

Figure 7. Doctor policy

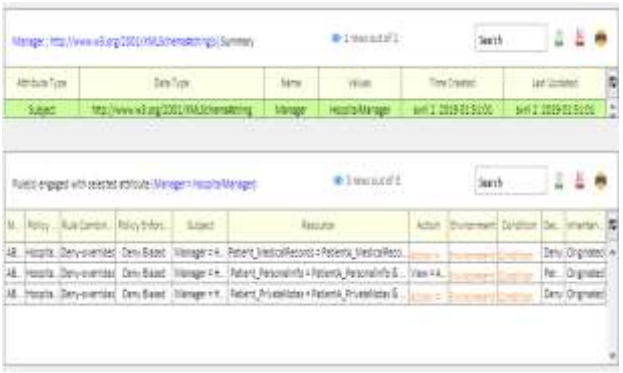

Fig. 8. Manager policy

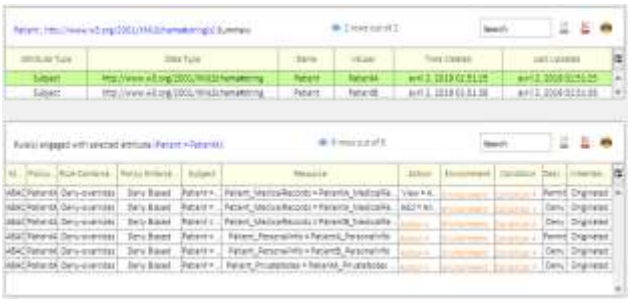

Fig. 9. Patient policy

Fig. 10. Policy rules converted on XACML language

Due to the difficulty of using XACML, we rely on the Security Policy Tool to implement the proposed policy rules. Subsequently, we use this tool to convert these policies into XACML language. As an illustration, Fig. 10 presents XACML policy for Doctor with Dentist attribute.

3) Creating individual security requirement

This phase is meant for testing the rules. So, it is necessary to create an individual security policy for this purpose. The Fig. 11 below is shown the security requirement of the Manager Policy after entering the following rule.

(Manager=HospitalManager)&(Action=AnyValue)&(Environment=AnyValue)&(Patient_PrivateNotes=PatientA_PrivateNotes&PatinetB_PivateNotes)→ decision=Deny

(Manager=HospitalManager)&(Action=View)&(Environment=AnyValue)&(Condition=Emergency)&(Patient_MedicalRecords=PatientA_MedicalRecords&PatinetB_MedicalRecords)→ decision=Permit

Fig. 11. Security requirement for HospitalManager

4) Verification of the policies

In this phase the SPT can test one or multiple policies against specific security requirements. A security requirement represents an access request (e.g., a HospitalManager requests any action to a Patient_MedicalRecords at any time and in case of emergency) as shown in the figure below. For instance, we evaluate access control decision for a Manager who tries to get access Patient_MedicalRecords as shown in Figure 12 below.

Fig. 12. Verification of HospitalManager within manager policy

V. Conclusion

The integration of Cloud and IoT has emerged as an innovative solution to address the scarcity of IoT

resources. In fact, Cloud provides large scale storage systems and powerful tools. However, despite the benefits of this model, it still faces several issues regarding security and privacy. In this regard, we suggest an efficient XACML basic access control to protect data collected form IoT devices. To this aim, we use SPT to easily design security policies. The simulation results prove that our proposed solution can prevent unauthorized access to confidential data. To illustrate the importance of the proposed methodology, we design and implement a policy that ensures the management of the hospital in secure manner.

CONFLICT OF INTEREST

Authors declare that they have no conflict of interest. And they do not have any financial relationship with any organization.

AUTHOR CONTRIBUTIONS

Fatima Sifou conceived of the presented idea. She wrote the manuscript with support from Marwan M.

Feda AlShahwan critically reviewed the study proposal and reviewed the final manuscript.

Adra Hammoud contributed to design and performed experiments, and helped in developing the theoretical formalism.

Mbarek Marwan provided critical feedback and helped shape the research, analysis, and manuscript.

Ahmed Hammouch provided the final approval of the version to publish, and ensured that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

All authors discussed the results, commented on the manuscrip and approved the final version.

REFERENCES

[1] M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing," *Journal of Network and Computer Applications, Elsevier*, pp. 99-117, 2016.

[2] F. Sifou, M. Marwan, and A. Hammouch, "Applying OM-AM reference to an ABAC model for securing cloud-enabled internet of things," in *Proc. 3th International Conference on System Reliability and Safety*, Barcelona, Spain, November 24-26, 2018.

[3] F. Sifou, A. Kartit, and A. Hammouch, "Different access control mechanisms for data security in cloud computing," in *Proc. International Conference on Cloud and Big Data Computing, London, United Kingdom*, 2017, pp. 40-44.

[4] Attribute-based access control. [Online]. Available: https://www.axiomatics.com/attribute-based-access-control/

[5] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: A survey," *Futur. Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016.

[6] S. M. Babu, A. J. Lakshmi, and B. T. Rao, "A study on cloud based Internet of Things: CloudIoT," in *Proc. Global Conference on Communication Technologies (GCCT)*, 2015, pp. 60-65.

[7] M. Marwan, A. Kartit, and H. Ouahmane, "A framework to secure medical image storage in cloud computing environment," *Journal of Electronic Commerce in Organizations*, vol. 16, no. 1, pp. 1-16, 2018.

[8] M. Marwan, A. Kartit, and H. Ouahmane, "Secure cloud-based medical image storage using secret share scheme," in *Proc. International Conference on Multimedia Computing and Systems (ICMCS)*, 2016, pp. 366–371.

[9] M. Marwan, A. Kartit and H. Ouahmane, "Applying secure multi-party computation to improve collaboration in healthcare cloud," in *Proc. Third International Conference on Systems of Collaboration (SysCo)*, 2016, pp. 1-6.

[10] H. F. Atlam, A. Alenezi, A. Alharthi, R.J. Walters, and G.B. Wills, "Integration of cloud computing with internet of things: challenges and open issues," in *Proc. IEEE International Conference on Internet of Things (iThings*), UK, June 21-23, 2017.

[11] W. He, G. Yan, and L. D. Xu, "Developing vehicular data cloud services in the IoT environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587-1595, 2014.

[12] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, "Twenty security considerations for cloud-supported Internet of Things," *IEEE, Internet of Things Journal*, 2015.

[13] A. Ouaddah, H. Mousannif, A. Abou Elkalam, and A. Ait Ouahman, "Access control in the Internet of Things: big challenges and new opportunities," *Computer Networks, Elsevier*, pp. 237-262, 2017.

[14] A. Abdl-Aziz, and K. Arputharaj, "A comprehensive presentation to XACML," in *Proc. 3th International Conference on Computational Intelligence and Information Technology*, India, 2013.

[15] Xacml [Online]. Available: https://www.axiomatics.com/blog/understanding-xacml-combining algorithms/

[16] M. Drozdowicz, M. Ganzha, and M. Paprzycki, "Semantically enriched data access policies in eHealth," *Journal of Medical Systems*, vol. 40, no. 238, 2016.

[17] Y. Keleta, J. H. P. Eloff, and H.S. Venter, "Proposing a secure XACML architecture ensuring privacy and trust," January 2005.

[18] Security Policy Tool [Online]. Available: https://securitypolicytool.com/

[19] The utilization of Security Policy Tool. [Online]. Available: https://securitypolicytool.com/help,

[20] *Security Policy Tool User Manual, InfoBeyond Technology LLC*, August 2017.

[21] *Healthcare policy test cases.* InfoBeyond Technology LLC, 2017.

Sifou F. received the Bachelor's degree in Mathematics and Computer Sciences in 2009 and the Master degree in software development "software quality" in 2011, from the faculty of sciences. Since 2017 she is a predoctoral researcher in the Department of computer sciences at Mohammed V University where she is pursuing a PhD degree. Her main research interests are Cyber security in IoT, Cloud Computing and their applications.

AlShahwan F. is the chair of the second IEEE GCC SYP Congress. Feda is currently an Assistant Professor at the Electronic Engineering Department/ Computer Section of the College of Technological Studies in the Public Authority for Applied Education & Training. She has diverse research interests in Mobile Web Services, IoT, cloud computing and their applications Born in Kuwait, obtained B.Sc., M.Sc. in

Computer Engineer from Kuwait University 1992, 2004 respectively. She had got her Ph.D.

Hammoud A. was born in Meknès, Morocco, in1988. She received the Bachelor's degree in Computer Sciences from Moulay Ismail University, Meknès in 2009 and the Master degree in software development "software quality" from Mohammed V University, Rabat, in 2011. Her research interests include security and privacy in IoT and Cloud Computing based on Access Control Model.

Marwan M. received the Engineer degreein computer science. He had got his Ph.D. "Security in Cloud Services" in the Laboratory of Information Technology (LTI) at National School of Applied Sciences (ENSA), El Jadida, Morocco. Marwan has held senior management level positions in several IT projects. His area of research covers security aspects in the cloud computing, Big Data, IoT in Healthcare domain.

Hammouch A. received the M.S. degree and the Ph.D. degree in automatic, electrical, and electronic by the Haute Alsace University, Mulhouse, France, in 1993, and the Ph.D. degree in signal and image processing by the Mohammed V University, Rabat, Morocco, in 2004. From 1993 to 2013, he was a Professor in the Mohammed V University, Rabat, Morocco. Since 2009 he manages the Research Laboratory in Electronic Engineering. He is an author of several papers in international journals and conferences. His domains of interest include multimedia data processing and telecommunications. He is with National Center for Scientific and Technical Research, Rabat, Morocco.