C.D.T.A

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية العلوم
Faculté des Sciences

قسم الإعلام الآلي
Département d'Informatique

Mémoire de Projet de Fin d'Études

présenté par

Ounnoughi AbdElKader

&

Yahiaoui AbdElKarim

pour l'obtention du diplôme de Master en Sécurité des Systèmes d'information (SSI)

Thème

Protection des données médicales et de la vie privée des patients à domicile

Promoteur :

M^me^ Narhimene Boustia

Encadre par :

M. Allal Tiberkak

Dr. HENTOUT Abdelfetah

Année Universitaire 2018-2019

C.D.T.A

الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

جامعة سعد دحلب البليدة
University of SAAD DAHLAB BLIDA

كلية العلوم
Faculty of Science

قسم الإعلام الآلي
Computer Science department

Thesis of End of Studies Project

presented by

Ounnoughi AbdElKader

&

Yahiaoui AbdElKarim

For the acquisition of the Master's degree in Security of Information Systems (SSI)

Theme

Protection of medical data and the privacy of patients at home

Promoter:

M^me^ Narhimene Boustia

Supervised by:

M. Allal Tiberkak

Dr. HENTOUT Abdelfetah

2018-2019

Thanks:

We first thank ALLAH for giving us not only the courage but also the strength and the will necessary for the accomplishment of this modest work.

Our heartfelt thanks to our parents, our brothers and sisters for their moral support and encouragement.

We thank our dear friends who are always present and faithful.

We would like to express our deep gratitude and our sincere thanks to our promoter Mrs. Narhimene Boustia and supervisor Mr. Allal Tiberkak and Dr. HENTOUT Abdelfetah for the high quality of supervision, monitoring, availability and advice. Without theme, the realization of this memoir would not have taken place. Once again, thank you so much.

We address our thanks to the jury members who made the honor of evaluating, examining and enriching our modest work.

Our gratitude goes particularly to all the teachers of the Computer science department of SAAD DAHLAB BLIDA.

Resume

The Internet of Things (IoT) opens the way to a multitude of scenarios based on the interconnection between the physical world and the virtual world: home automation, health, smart city, security, etc. One of the main constraints in the healthcare IoT world is the protection of the privet data especially for the patients who need monitoring from their home , for this, the IoT requires effective security mechanisms.

In this thesis, we will be employing a lightweight multi-level access control to enforce a robust security system to handle all the flowing data between IoT devices, server and users.

Keywords: Access control, internet of things, encryption, security, healthcare, Ubiquitous computing.

Résumé

L'Internet des objets (IoT) ouvre la voie à une multitude de scénarios basés sur l'interconnexion entre le monde physique et le monde virtuel : domotique, santé, ville intelligente, sécurité, etc. L'une des principales contraintes du monde de la santé IoT c'est la protection des données confidentielles, en particulier pour les patients qui ont besoin d'une surveillance de leur domicile, pour cela, l'IoT nécessite des mécanismes de sécurité efficaces.

Dans cette thèse, nous allons utiliser un contrôle d'accès multi-niveau léger pour mettre en œuvre un système de sécurité robuste permettant de gérer toutes les données circulant entre les périphériques IoT, le serveur et les utilisateurs.

Mots-clés : Contrôle d'accès, Internet des objets, cryptographie, sécurité, santé, intelligence ambiante.

ملخص

يفتح انترنت الأشياء الطريق أمام العديد من السيناريوهات القائمة على الترابط بين العالم المادي والعالم الافتراضي: أتمتة المنزل، الصحة، المدينة الذكية، الأمن، إلخ. أحد القيود الرئيسية في عالم الرعاية الصحية بأنترنت الأشياء هو حماية البيانات الخاصة، خاصة للمرضى الذين يحتاجون إلى المراقبة من منازلهم، ولهذا فإن إنترنت الأشياء يتطلب آليات أمنية فعالة.

في هذه الأطروحة، سنستخدم نظام مراقبة الدخول متعدد المستويات وخفيف الوزن لفرض نظام أمان قوي للتعامل مع جميع البيانات المتدفقة بين أجهزة إنترنت الأشياء والخادم والمستخدمين.

كلمات المفاتيح: انترنت الأشياء، نظام مراقبة الدخول، التشفير، الأمن، الرعاية الصحية، الحوسبة في كل مكان.

Table of contents

List of Figures

Acronyms and Abbreviations

IoT	Internet of Things
EHR	Electronic Healthcare Record
E-Health	Electronic health
MQTT	Message Queuing Telemetry Transport
ABAC	Attribute based Access Control
MAC	Mandatory Access Control
DAC	Discretionary Access Control
CBAC	Context Based Access Control
RBAC	Role Based Access Control
ABE	Attribute Based Encryption
KP-ABE	Key Policy Attribute Based Encryption
CP-ABE	Cipher Policy Attribute Based Encryption
CHAP	Challenge Handshake Authentication Protocol

General Introduction

We are gradually witnessing a transformation in our daily objects: more and more of them are connected to the Internet and a by phenomena called the "Internet of Things" (IoT), a revolution that lead to a huge increase in Connected Devices, this one has reached 17 billion in 2018 and 7 billion of them are IoT devices [1].

IoT, also known as Web 3.0, represents an extension of the Internet to things and places in the physical world. Each physical object - which can be a person, a computer, a smartphone, a car, a sensor, a smart home, etc. - is associated with a virtual entity that behaves like an active entity in the system. The success of this new paradigm is explained by the variety of equipment (objects) used in our daily life and their development.

Healthcare is such a vast ecosystem, that the applications of the IoT in healthcare seem to be endless: from remote monitoring and personal healthcare to smart sensors and medical device integration, as well as the pharmaceutical industry, healthcare insurance, healthcare building facilities, robotics, smart pills, and even treatments of diseases. It has the potential to not only keep patients safe and healthy, but to improve how doctors deliver care as well.

With such huge personal and sensible data being handled by IoT, it causes a multitude of problems resulting in making some known security methods obsolete so the privacy of user is most endangered from data theft, embezzlement and identity fraud. Therefore new security measures have been popping up to accommodate the growing needs of the market and to ensure the privacy of users, the principal goal of this thesis is to produce an efficient security measures to protect the user's privacy and sensitive data.

The purpose of this thesis is to try to handle such problematics accordingly to the following scenario:

Jaafar a 75-year-old Cardiac patient has been living alone in his home out of the city. Concerned about the possibility of falling or heart attack, his son, bought him an accelerometer device with heartbeat monitor embedded in a smartwatch and set up multiple cameras around the house to monitor Jaafar's activity including smart doors and a smart insulin pump. Jaafar himself then have to choose a group of trusted persons (people appointed by the patient) to take care of him, a personal doctor and a caretaker in case of incapacitation. The caretaker is the person responsible of the patient (parent, son, wife, etc.) and is only active when it is needed, this role can be delegated if required.

The devices continuously stream their collected data and video plus audio, to a nearby personal Gateway (Hub). Upon installing the software that monitors and control these devices, the patient agrees to an End User License Agreement (EULA) that grants consent for the Gateway to dial emergency services on their behalf if his medical condition requires it.
This also allows him to store personal EHR (electronic healthcare record) for ease of access to him and his personal doctor.
It was not long after the Jaafar started using the device when they started showing signs of cardiac attack. The Gateway then recognizes this abnormal signs using the data from the smartwatch and by considering the drop in the heart rates. The Gateway will declare an emergency state, send an emergency message via SMS and send a notification to the mobile app inquiring a response, if they fail to respond a set period of time, the Gateway will then call emergency services.
While the emergency state is active, the trusted persons will acquire elevated privileges that include Door control and Video stream.
The patient, doctor or caretaker are the only ones who can deactivate the emergency state.
In case of false alarm, the patient, doctor or caretaker can remotely deactivate the emergency state.
This thesis is organized in 4 chapters:
The 1st chapter will introduce the concept of IoT and its impact on current life.
The 2nd chapter will study different access control protocols and decide on an approach to use.
The 3rd chapter will be devoted for the study of the requirement in our work.
The 4th is the final chapter that will present the realization of this work.

Finally, we will close with a general conclusion and the future improvement to this work.

Chapitre 1: Internet of Things (IoT)

1.1 Definition

The definition of the Internet of Things (IoT) has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems, and others all contributing to the thriving increase of the overall use these devices across a multitude of practices. Therefore, it is difficult to give it a stable definition for many reasons, one of theme comes from the different visions of IoT that stakeholders (business, researchers, standardization bodies, etc.) have; each of these stakeholder adopts a different description for the IoT depending on their interests, purpose and the context in which the term is used.

Therefor we can cite how Coetzee & Eksteen describes the IoT: “a vision where devices become part of the Internet: where every device is uniquely identified, and accessible to the network, its position and status known, where services and intelligence are added to this expanded Internet, fusing the digital and physical world.”[2]

In addition the Cluster of European research projects on the IoT [41] describes “Things” as: “active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information sensed about the environment, while reacting autonomously to the real/physical world events and influencing it by running processes that trigger actions and create services with or without direct human intervention.”[3]

Figure 1 Indirect connection between devices and the internet [7]

While it is not easy to suggest a common definition, in this thesis we will define the IoT as follows:" IoT is a network of connected devices that can collect, store and transmit data from the physical world to the digital world with a direct (represented in Figure 1) or indirect (in Figure 2) link between theme and the Internet using wireless communication system (RFID chip, Zigbee, Bluetooth or Wi-Fi, etc.)".

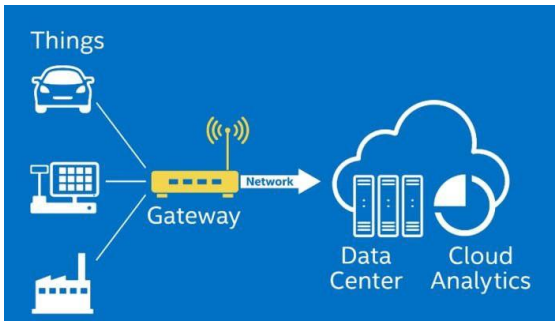


Figure 2 Direct connection between devices and the internet [6]

1.2 The evolution of IoT ecosystem

Kevin Ashton (P & G) came up with the expression "Internet of Things" in 1999 to characterize the connection of RFID chips to the Internet. After some unsuccessful attempts, the first device called "connected" was the Nabaztag -Marketed in 2005 by Violet [8], a Wi-Fi connected rabbit that was able to give the weather prediction or read the e-mails audibly and offered personalized alerts. In 2006, Apple collaborated with the Nike for the launch of

Nike + iPod shoe that sent to an iPod the statistics of your jogging sessions, using a built-in sensor.

We are gradually witnessing a transformation in our daily devices: more and more of them are connected to the Internet, a revolution that lead to a huge increase in Connected Devices, this one has reached 17 billion in 2018 (7 billion of them being IoT devices).

IoT Analytics estimates that 10 billion devices will be connected to the Internet by 2020 and 22 billion by 2025 as shown in Figure 3. This number of IoT devices includes all active connections and does not take into consideration devices that were bought in the past but are not used anymore [8].

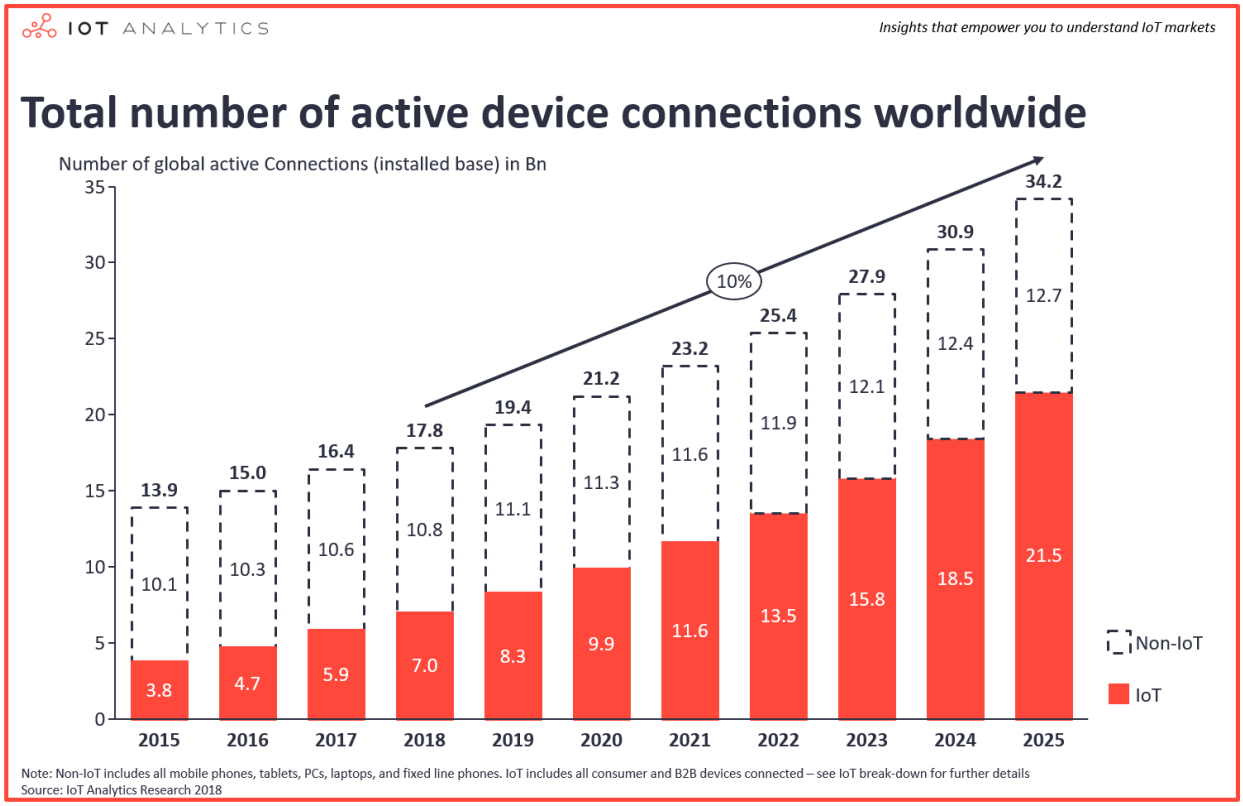


Figure 3 Total number of active device connections worldwide [8]

1.3 IOT Platforms

The real issue of the connected devices lies in the gathered data. Connected sensors record information locally or send it directly to a data storage center. This flow have to be process (in real time or afterwards according to the needs) in order to get all the necessary intelligence for an activity. This is where the connected devices platform comes in. It bridges the gap between IoT and Big Data and offers to the users a remote management of the data-

flow. In reality, its operation is much more complex. Platforms can intervene at the lowest levels of network equipment, servers, embedded systems, software and even at the level of the connected device [9].

These platforms are available as cloud based and standalone, they are available from many companies such as:

- Amazon Web services (AWS),
- IBM Watson Bluemix,
- Microsoft Azure,
- ThingWrox.

1.4 Architecture of IoT

The architecture of connected devices system is composed of several levels that communicate with each other to connect the physical world to the virtual world of data. Not all IoT projects adopt a formally identical architecture; however, it is possible to structure the course of data as shown in Figure 4.

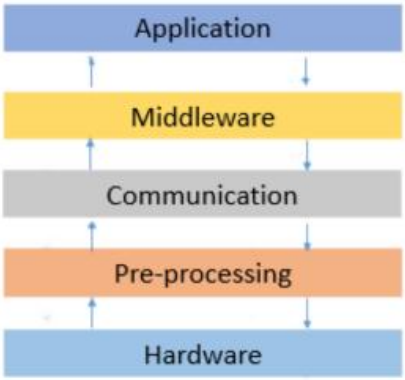


Figure 4 The IoT architecture

1.4.1 Hardware

The hardware consists of the physical devices made of sensors, actuators and embedded communication systems. This layer basically deals with the collection of specific information depending on the sensor type; and passes on the data it collects to the application layer. [4][11]

1.4.2 Pre-processing

Connected devices collect a big amount of data to be stored, processed or/and communicated with other elements, the most commonly used solution to handle these tasks is the use of a cloud (due to its massive data handling, flexibility and scalability). However, sending huge amount of data through the network could overload it, especially in large IoT networks or in a network where bandwidth is very limited; additionally, the frequent communications consumes a substantial amount of energy [11].

A common solution to this problem is fog computing that is commonly used when handling connected devices with the cloud, Cisco defines this concept as a paradigm that extends Cloud computing and services to the edge of the network (router, firewall, server, etc.) as shown in Figure 5, allowing the gateway to pre-process part of the data before sending it to the cloud [12].

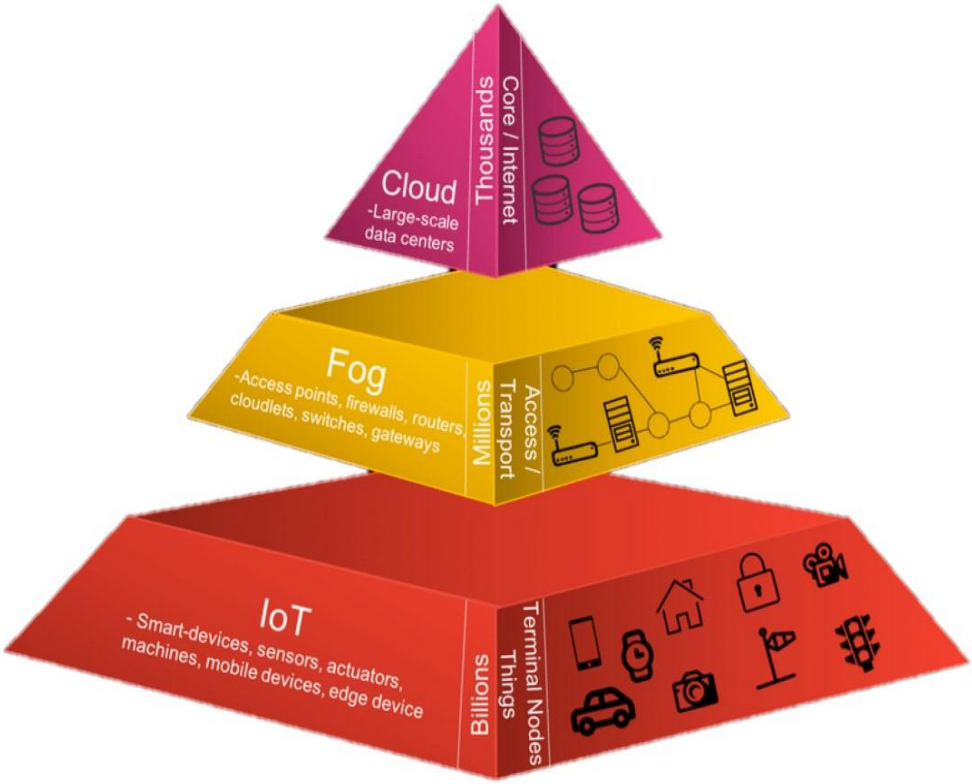


Figure 5 Simple representation for IoT-fog-cloud ecosystem [16].

Another important problem is that the collected data can be full of noise (i.e. receiving corrupted information that can harm or changes negatively the results).

However, it is undeniable that discovering irregularities is essential so we have to accept all collected data. Therefore, depending on the situation: noise, anomalies and outliers need to be cleaned or not. Accordingly, it is vital to adopt data preprocessing algorithm, so accuracy of data would be improved and beneficial [10].

1.4.3 Communication

This layer performs the communication between connected devices and remote servers. Because of their constrained nature (high number of connected devices, low power devices, high data flow…) the communication protocols and equipment must deal with five main challenges: addressing, identification, low power communication, efficient communication patterns and a high-speed communication [11].

The leading communication technologies to tackle these challenges in the IoT world are classified as the following network layers:

1- *Physical Layer*: IEEE 802.15.4 is one of the most used standard in the IoT field; it is the basis for the Zigbee, ISA100.11a, WirelessHART, MiWi, 6LoWPAN, Thread and SNAP specifications. This standard defines the operation of low-rate wireless personal area networks and it focuses on low-cost, low-speed ubiquitous communication between devices.

2- *Adaptation Layer*: 6LoWPAN, an acronym for IPv6 over low power wireless personal area networks, it is a very popular standard for wireless communication enabling communication using IPv6 over the IEEE 802.15.4 protocol. This standard defines an adaptation layer between the 802.15.4 link layer and the transport layer. 6LoWPAN devices can communicate with all other IP based devices on the Internet.

3- *Network Layer*: RPL (Routing Protocol for Low-Power and Lossy Networks) is a routing protocol for wireless networks with low power consumption and generally susceptible to packet loss. A proactive protocol based on distance vectors that operates on IEEE 802.15.4 and support a wide variety of link layers, including those with limitations , potential losses or in devices with limited resources. This protocol can quickly create network routes, share routing knowledge and adapt the topology in an efficient way.

4- *Transport Layer*: there is a tendency of using UDP over TCP in the IoT field because it has low overhead, which mean lower power consumption.

5- *Application Layer*: Many alternate protocols have been developed for IoT environments such as CoAP (Constrained Application Protocol) and MQTT (Message Queue Telemetry Transport).

- MQTT: is a lightweight publish/subscribe messaging protocol, where every sensor is a client (subscriber) that connects to the broker (MQTT server) over TCP. MQTT Clients can subscribe to multiple topics (which is an address dedicated to a service) to receive every message published to the topic. Publisher publishes the messages to the MQTT broker who in return forwards the messages to the listening subscriber, so there is no direct connection from a client to another [13].
- CoAP: is the second most used protocol for IoTs that uses the concept of RESTful API and uses the PUT, GET, POST, DELETE command much like HTTP; but rather than using TCP protocol CoAP use UDP to be more efficient and suitable for IoT devices. With CoAP the communication go directly from the client to the server. [17]

1.4.4 Middleware

This layer is responsible for the service interoperability, because connected devices can implement different types of services; and as consequence, they connect and communicate with only those that implement the same service type.

Middleware, act as a software bridge between devices and applications. It provides the required services (programing abstraction, device discovery and management, etc.) for the developers so that they can focus more on the requirements of applications rather than on interacting with the baseline hardware.

Furthermore it can receive the information from connected devices and store them in a database, performs information processing and global computation, takes automatic decision based on the results and address security and privacy issues, in addition to implementing a user authentication and an access control solution. [4][11].

1.4.5 Application

This layer provides global management of the data processed in the Middleware layer. These applications can be smart health, smart farming, smart home, smart city, intelligent transportation, etc. [4][11]

1.5 E-Health

eHealth is the single-most important revolution in healthcare since the advent of modern medicine, vaccines, or even public health measures like sanitation and clean water, The term e-Health (E-Health, eHealth ...) has been in use since the year 2000 [14].

Nevertheless, as the e-health sector is very broad, and the health care is even broader, healthcare IoTs are just a sub-field of e-Health as shown in Figure 6.

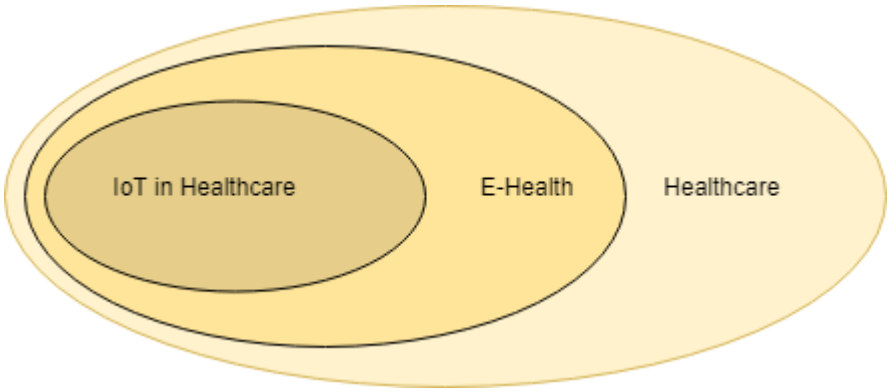


Figure 6 Healthcare sub-fields

1.5.1 Definition

The World Health Organization offers the following definition:

“E-health involves a broad group of activities that use electronic means to deliver health-related information, resources and services: it is the use of information and communication technologies (ICT) for health” [18].

The definition offered by Gunther Eysenbach is the one that we will be adopting since it is the most detailed definition:

“e-health is an emerging field in the intersection of medical informatics, public health and business, referring to health services and information delivered or enhanced through the Internet and related technologies. In a broader sense, the term characterizes not only a technical development, but also a state-of-mind, a way of thinking, an attitude, and a commitment for networked, global thinking, to improve health care locally, regionally, and worldwide by using information and communication technology.” [18].

1.5.2 IoT in E-Health

IoT in the field of health is expanding with the main purpose of collecting the maximum data to guide and improve the medical diagnosis as well as to assist the patient in the treatment of his illness, or help people in need of assistance.
The most emerging field now is the accompaniment of patients through connected devices to monitor their health. [4]

By integrating new IoT sensors and devices, people can be identified and monitored in real time, moreover, it will make it possible to diagnose the conditions of patients to provide real-time information on his health. Consequently, this new technology offer two great benefit in healthcare domain which are:

- Increasing workflow through patient monitoring.
- Optimization of the supply chain (an efficient IoT roadmap means automating managing processes, monitoring day-to-day activities to detect structural bottlenecks, etc.).

1.5.3 IoT in E-Health benefits

Advances in IoT technology has helped push significant advances in healthcare that resulted in many benefits, some of them are [5] [11]:

- Connected devices gathering a huge number of data that could help a good understanding of the disease and making the proper decision for each patient based on evidence and personal data;
- Improving access to effective healthcare by reducing barriers created, for example, by physical location or disability;
- Facilitating patient consent for self-care and health decision making;
- Improving diagnostic accuracy and treatment appropriateness;
- When used with big data analytics, connected devices can examine a patient's health using all information collected about him to estimate future health concerns before the beginning of the disease;
- IoT eHealth can organize medical services to avoid duplicative procedures and surcharges of certain crucial services like the emergency services. It also enables patients to more closely monitor their health to determine if medical attention is needed or not;

1.5.4 IoT in E-Health Challenges

IoT appliances have proven really beneficial in the healthcare domain however creating a large network of connected devices have proven to be a real challenge for the developers, following are the major ones [5]:

- *Design*: when designing IoT platform, a lot of characteristic should be considered, some of them are the heterogeneity of the system (connected devices technologies are not always compatible with each other), the continued evolution of the system, the human factors and the usability of the system…
- *Data management*: dealing with two concerns: large variety and the big volume of the collected data.
- *Scalability*: the deployment of such IoT technology could vary from the scale of a house to a hospital or even a national healthcare system, therefore accessing this system or gathering data becomes more and more difficult as the scalability gets bigger.
- *Interoperability*: The prospect of standardization the manufactures have to adopt raises a number of concerns, one of theme is strict regulation imposed on the devices by health agency that differed from a country to another.
- *Interfaces and human factors*: Many of the IoT systems end users have little or no knowledge of wireless networking or sensor and similar concepts, especially the elderly populations that are the most notable users, so there are a clear need for eHealth systems that can be deployed simply and have a friendly user interfaces.
- *Security and privacy*: privacy is always a concern when collecting personal data, and when this data is related to the healthcare. It could be more dangers, for that there is a huge amount of security solutions. However, due to the constraint imposed on the connected devices (limited memory, computing capabilities, etc.) most of the modern security technologies are not suitable for healthcare applications.

1.6 Security in IoT

Wherever networks are deployed, security concerns will emerge, and IoT networks are not an exception to this rule. In the domain of healthcare security, it is an extremely important factor to guarantee the privacy of the patient and even his own safety due to the possibility that these devices can affect him negatively if compromised (example: change the function of an insulin pump to overdose the patient).

1.6.1 Security-related threats

Connected devices related threats can be grouped together in four categories where attacks can occur the following are some of the well-known attacks:

1. **Devices:**
    - *Sleep Deprivation Attack*: when the devices are powered with batteries they generally follow the sleep routines to extend their lifetime. Sleep Deprivation is a kind of attack that keeps the nodes awake, resulting in more battery consumption, which causes a power failure [36, 37].
    - *Cloning attack*: an attack involving RFIDs tags. The attacker's aim is to obtain the important information on the tag through reverse engineering or directly from its deployment environment [37].

2. **Network:**
    - *Denial of Service (DoS) Attack*: DoS attack is accomplished by submerging the victim with requests, thereby generating a large amount of network traffic [36, 37, 38]. This type of attack can exhaust all available resources, making network resources unavailable to users. Furthermore a distributed denial-of-service attack (DDoS attack) can deploy multiple computers (generally zombie) to launch large-scale attacks on one or more targets [37].
    - *Main-in-The-Middle Attack*: a form of eavesdropping that targets communication channels due to which the unauthorized party can monitor or control all the private communications between the two parties. More so, the unauthorized party can even fake the identity of the victim to gain even more information [36, 37].

- *Sinkhole Attack*: Attackers use a node included in the system or injects a malicious node to attract data flow from nearby nodes [37], the system is fooled and considers the data to have already reached its destination.
- *Sniffing Attack*: Attackers use specific devices or applications to obtain the network traffic and then extract valuable data for the further attacks like passwords, active devices, etc. [36]

3. **Application:**
    - *Malicious Insider attack*: occurs when a trusted person tampers the data for malicious reasons. This person has access to the data that he will extract or alter easily on purpose from the inside [36].
    - *Privilege escalation attack*: when an attacker modify his initial unauthorized access by expanding or elevating his single user privileges till he gains complete administrative privileges (root) [39].
    - *Phishing Attack*: In this type of exploits, an attacker pretends to be a real user or legitimate institution to obtain sensitive information about the users, such as passwords and credit card details [36]. Usually carried out by email or instant messaging, it often directs users to enter personal information at a fake website, which matches the look and feel of the legitimate site.

1.7 Conclusion

In this chapter, we have overhauled in a general way the IoT technology. We have defined what the connected devices are. Then we presented its evolution, an architecture model, and finally we tackled the security problems posed by the use of them in E-health. This chapter has been devoted to the presentation of the IoT, its areas of application as well as its importance in E-health. Also, we have described in detail the security problems related to its deployment. The next chapter will be devoted to the study of some recent work carried out in the context of access control.

Chapitre 2: Access Control

2.1 What is Access Control

An access control system is a security measure that control who or what can use a resource in our environment by granting access when appropriate and denying when inappropriate. Access control tools help accomplish this purpose, as do firewalls, encryption, and intrusion detection, it is a primary and fundamental concept of security [19]. Typically an access control system (ACS) does that by taking users identifiers such as passwords, personal identification numbers (PINs), biometric scans, tokens or others factors and uses them to verify that the users are who or what they claim to be and authenticates them before granting or denying access to the resources accordingly.

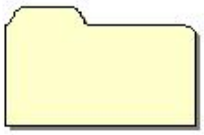
2.2 Types of Access Control

There are multiple types of access control existing currently but the main and the most commonly used ones consist of:

2.2.1 Mandatory Access control (MAC)

A MAC is obligatory that is, it dictates whether an operation should be permitted or denied without letting a user override the policy as is a security strategy that revokes the ability of resource owner to grant or deny access to their resources and it is all handled by the system administrator [19] often used in government and military facilities it works by assigning a classification label to each file system object. Classifications include confidential, secret and top secret. Each user and device on the system is assigned a similar classification and clearance level. When a person or device tries to access a specific resource, the OS or security kernel will check the entity's credentials and the user’s clearance level to determine whether access will be granted as shown in Figure 7.

Therefore, for that reason MAC ends up being not very flexible and with very limited user option, interaction, and intensive on the administrator [19].



User1, Clearance Level 2

Resource 1, Top Secret.

Clearance Level	Classification
Level 5	Top Secret, Secret, Classified, UnClassified
Level 4	Secret, Classified, UnClassified
Level 3	Classified, UnClassified
Level 2	UnClassified

Figure 7 MAC representation

2.2.2 Discretionary Access control (DAC)

DAC is an identity based strategy built-in in most OS that relies on the discretion of the user and the resource owner. In DAC the control is maintained and enforced by an Access Control List (ACL) that is attached to a resource which is defined by its owner to define the access type (grant or deny) to the users of that systems. As it also allows the transfer of authenticated objects or information access to other users by changing or increasing the number of owners of that specified resource. However for these same reasons DAC is very labor intensive as each user must define an access for all users to every resource he owns so its prone to mistakes made by resource owners that are very hard to identify as DAC doesn't scale well and is made to work on small systems where the number of users is manageable to not end ep with an ACL explosion [19].

2.2.3 Role-Based Access Control (RBAC)

RBAC is a commonly used Access control system that uses defined groups in a business or work environment (teacher, student, director …) to restrict access to the protected resource rather than the identities of the users [19], the role-based access protect resources using the predefined privileges and restriction of each role in the system in each object through an ACL, and enforces them on the users depending on their Role membership and a user can have from one to multiple roles. RBAC is great for system with a high turnover of users and

for that reason the roles needs to be well provisioned, defined and maintained as it can result in the possibility of role explosion , moreover RBAC lacks the concept of Context and cannot adapt to real time context.

2.2.4 Attribute-Based Access control (ABAC)

ABAC uses a different approach to access control by using a set of attribute of users, data, environment and a list of policies that allows a high granularity level.

Policies are a set of rules that ABAC uses to grant or deny access to a user to that resource and they are made up three type of attribute , Users to define who or what can access the resource and it can be identity, role , level of the user. Data which determines to what the policy is going to control and protect and Policy also use the environment attribute (time, place, state etc.) to include the concept of context to the policy.

The weakness of ABAC is its complexity compared to other types of access control and is usually hard to adapt to but it compensate that by being able to accommodate to every type of control and its complexity can be negated by how well it can be implemented and simplified [19].

2.2.5 Context-Based Access Control

CBAC is most commonly used to protect traffic through firewalls. Context Based Access Control means that the decision whether a user can access a resource doesn't depend solely on who the user is and which resource it is or even the resource content, but also in the sequence of events that preceded the access attempt[20].

CBAC specifies which traffic should be allowed on the inside and the outside using the access lists. However, CBAC access lists include IP inspection statements that allow the protocol's inspection to ensure that it’s not tampered with before the protocol goes to the systems behind the firewall.

2.3 ABAC

Though there are many types of access control systems we have decided to dive deeper and chose Attribute-based access control as the main for the following reasons.

Role-based access control and other type control have been around since the 1970 and continue to be a well-accepted standard however with the arrival of attribute-based access control system many companies are shifting their views to take advantage of the granularity level of authorization that it can provide.

In the digital age, data access control is best done with an ABAC model. ABAC can employ user attributes, action attributes, context attributes (such as time, device and location), resource attributes (such as a record's sensitivity), and much more. The fact that we can use multiple attributes (data) that describe the user, the resource and the context makes ABAC multi-dimensional and capable of supporting virtually any access control scenario.

What this means in a healthcare scenario is that the authorization decision process, whether if a user should be granted access to a resource, is a dynamic process that evaluates the context as a whole.

Data access control has evolved to meet the changing data protection challenges organizations face in the age of unlimited data. Evolving from ACLs and RBAC, ABAC is now the standard model for organizations to ensure employees only have access to the information they need under the right circumstances [19].

2.3.1 Introduction to ABAC

Fully understanding ABAC requires understanding of the basic principles of logical access control. The purpose of logical access control is to protect objects (be they data, services, executable applications, network devices, or some other type of information technology) from unauthorized operations. These operations may include discovering, reading, creating, editing, deleting, and executing objects. These objects are owned by an individual or organization (teacher, director, patient …etc.) and have some inherent value that motivates those owners to protect them by using a complex set of a multitude of attributes and a list of precise policies designed by the business to enforce the rules and privileges.

2.3.2 Definition

ABAC is An access control method where subject whether they be users or application requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions[21].

2.3.3 ABAC Components

ABAC is composed of a multitude of components used to enforce and deploy the control system as shown in Figure 8, it is fundamental to understand each and every one of them to better comprehend how ABAC functions and these components are [21]:

Attributes: are characteristics of the subject, object, or environment conditions. Attributes contain information given by a name-value pair such as (Age: "20").

Subject: is a human user or non-person entity (NPE), such as a device that issues access requests to perform operations on objects. Subjects are assigned one or more attributes. For the purpose of simplicity, assume that subject and user are synonymous.

Object: is a system resource for which access is managed by the ABAC system, such as devices, files, records, tables, processes, programs, networks, or domains containing or receiving information. It can be the resource or requested entity, as well as anything upon which an operation may be performed by a subject including data, applications, services, devices, and networks.

Operation: is the execution of a function at the request of a subject upon an object. Operations include read, write, edit, delete, copy, execute, and modify.

Policy: is the representation of rules or relationships that makes it possible to determine if a requested access should be allowed, given the values of the attributes of the subject, object, and possibly environment conditions.

Environment conditions: operational or situational context in which access requests occur. Environment conditions are detectable environmental characteristics. Environment characteristics are independent of subject or object, and may include the current time, day of the week, location of a user, or the current threat level.

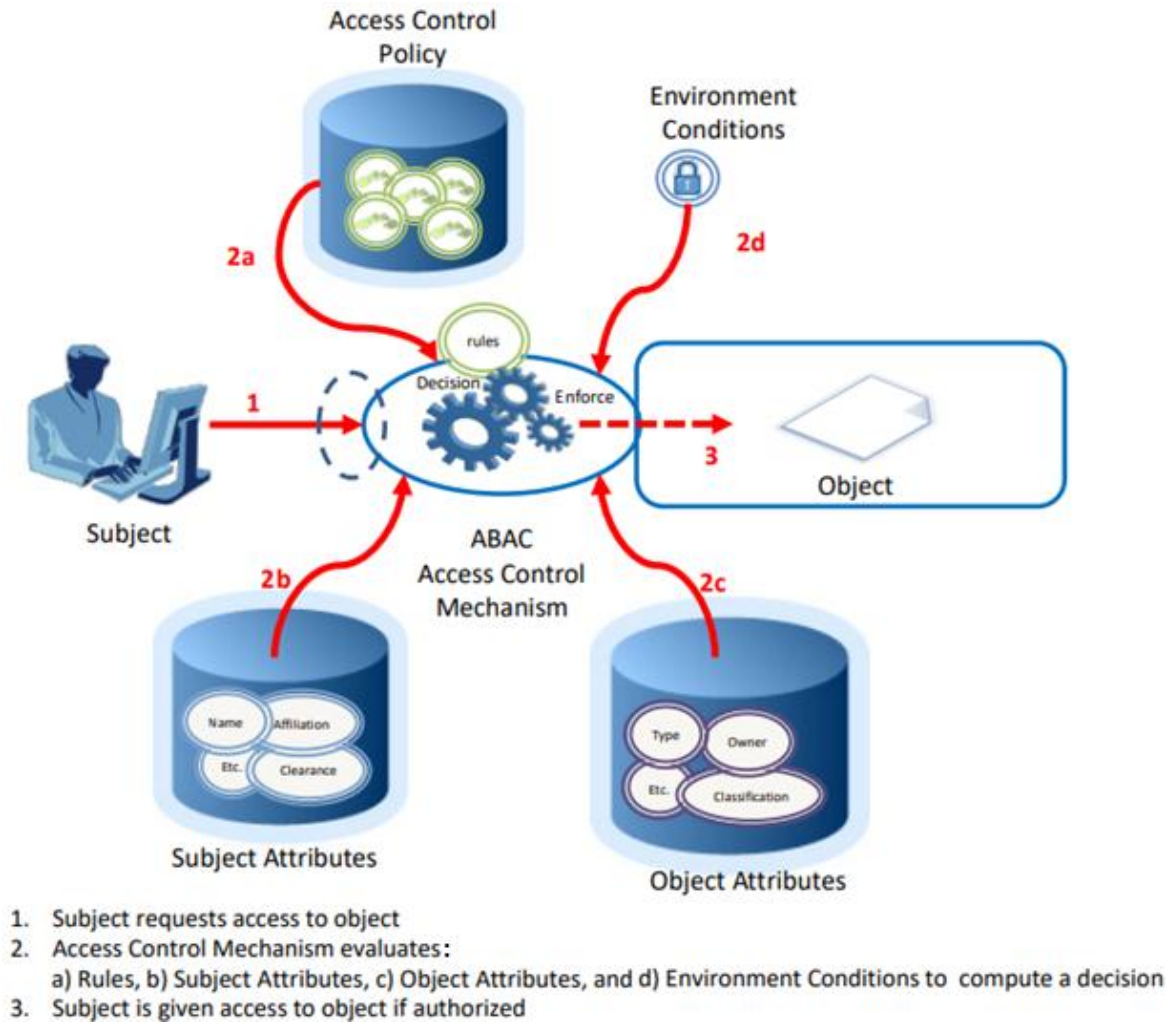


Figure 8 Attribute based access control structure.

2.3.4 Concept

ABAC relies upon the evaluation of attributes of the subject; attributes of the object, environment conditions, and the formal relationship or access control rule or policy defining the allowable operations for subject-object attribute combinations as shown in Figure 9.

ABAC relies upon the assignment of attributes to subjects and objects, and the development of policy that contains the access rules.

Each object within the system must be assigned specific object attributes that characterize the object. Some attributes pertain to the entire instance of an object, such as the owner. Other attributes may only apply to parts of the object.

Each subject that uses the system must be assigned specific attributes. This subject may have a name, a role, and an organization affiliation.

Every object within the system must have at least one policy that defines the access rules for the allowable subjects to the object according to operations, and environment conditions.

The objects selected to be shared and protected by the ABAC solution will vary based upon organizational requirements. Each object or class of object must be identified and the policy or rules protecting each must be documented. A set of business processes need to be established to identify, class, and assign policy to each new object created within the scope of the ABAC implementation [21].

The rules that bind subject and object attributes indirectly specify privileges (i.e., which subjects can perform which operations on which objects). Allowable operation rules can be expressed through many forms of computational language such as:

- A Boolean combination of attributes and conditions that satisfy the authorization for specific operation.
- A set of relations associating subject and object attributes and allowable operations.

And though ABAC is sufficient by itself it still isn't enough to protect users data in this day and age for lack of encryption system in it and for that purpose Attribute based encryption was created.

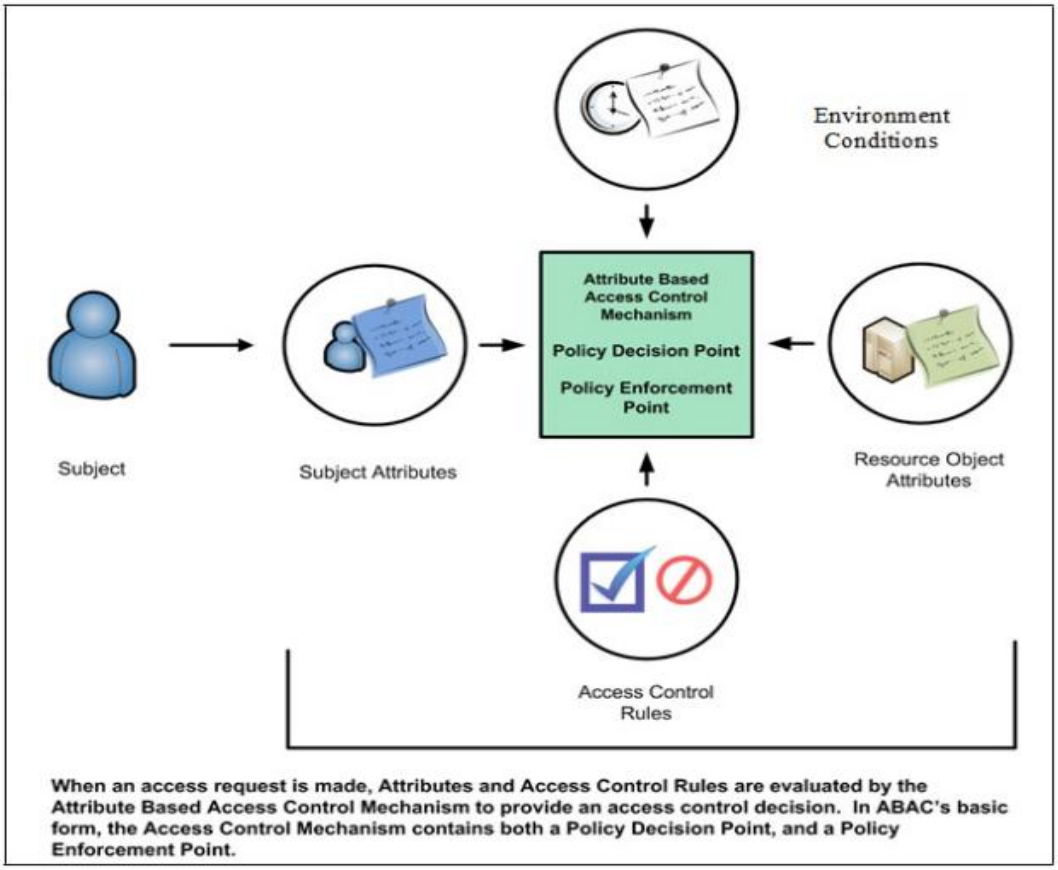


Figure 9 ABAC Mechanics.

2.4 Attribute-based encryption

Attribute-based encryption (ABE) is an approach that redefines the concept of public-key (PUK) cryptography that is most commonly known, in the traditional way PK cryptography has always operated, a message is encrypted using the recipient's Public Key (PUK a random big number), which then can only be decrypted using their private key (PRK). While identity-based encryption (IBE) has altered the traditional understanding of this method by allowing the Public Key to be an arbitrary string that designates the identity of the person [22]. In that same way, the Attribute based encryption added the possibility that the message encrypted with the chosen public key could be decrypted with several Private keys issued by a trusted authority.

2.4.1 Definition

ABE is an asymmetric encryption primitive in which the encryption key of a user depends on attributes that are unique to him, for example: his identification code, level of clearance, name or his role in the system as shown in Figure 10. In this type of encryption, the decryption of the secret message is possible only if the attributes of the recipient's key correspond to the attributes of the encrypted message. In our days, ABE is dominantly used in the protection of personal medical data for its ability to make an object able to be decrypted by several different keys using policies defined by system like ABAC; in addition, this approach allows imposing access rights at a high level of granularity due to the way attribute r so intricate and detailed.

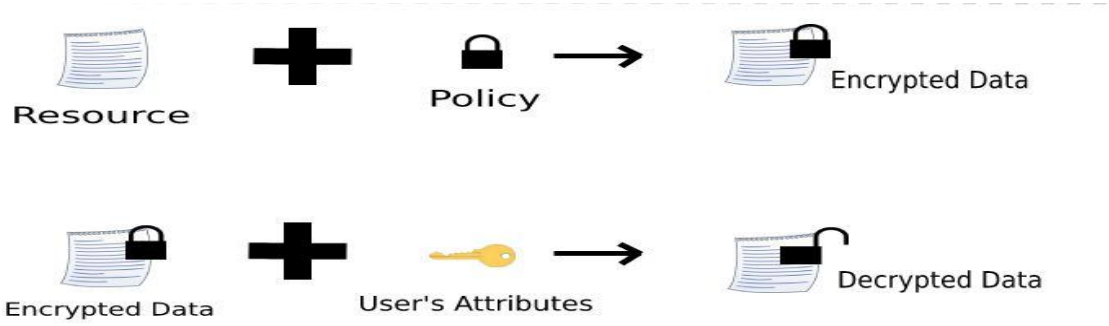


Figure 10 Basic ABE mechanic.

2.4.2 Types of ABE

The concept of Attribute-based encryption appeared in the year 2005 under the name of Fuzzy Identity-Based Encryption developed by *A. Sahai and B. Waters*, and several research aimed at improving it, which had concluded the development of two major versions, Key-Policy ABE (KP-ABE) followed by Ciphertext-Policy ABE (CP-ABE).

2.4.2.1 Fuzzy Identity-Based Encryption

The first version of ABE consists of Encrypting a text with a set $\boldsymbol{\alpha}$ of attributes, and the one who wants to decipher the text must have a key that contains a set $\boldsymbol{\beta}$ of attribute such as that $|\boldsymbol{\alpha} \cap \boldsymbol{\beta}| \geq \boldsymbol{d}$, $\boldsymbol{d}$ is a threshold defined by the user [23]. This method while theoretically is plausible but actually, it have problems including a major one the confusion problem:

For example, if a Person implement a rule stating that the medical information can only be accessed by their owner identified by his ID or a cardiologist who works at Mustapha Hospital, for example assuming we have a set of medical information with the attributes { user = "001", specialty = cardiologist, Hospital = Mustapha } and two users Omar with the attribute { user = "001"} and Ali with the attribute { specialty = cardiologist, Hospital = Dely-Ibrahim}.

In this rule, the threshold $\boldsymbol{d}$ must be equal to "2" to verify the second part of the rule yet the first part requires one attribute so $\boldsymbol{d}$ should be equal to "1" and that is where a confusion occurs.

If the threshold is set to "1" and the person requesting access is Ali the system will consider the rule valid and will accept the request, even if Ali doesn't work in Hospital A. In addition, if the threshold is set to "2" and the person requesting access is Omar he will not get access.

One possible solution is to transfer the second rule to doctors by assigning the keys of all patients to each cardiologist at Hospital A and recalculate the keys for each new patient, but this remains a very taxing and difficult solution to manage.

2.4.2.2 Key-Policy ABE

While Subsequently, *Goyal and the others.* Have proposed KP-ABE in which the Attributes of any user are constructed in a tree-like policy way that's used to define the user's key ,The leaves of the tree are associated with the attributes and the non-leaf nodes are logical operations, such as "and" and "or" Figure 11. The data owner associates his encrypted text

with a set of attributes. If the associated attributes satisfy the user's specific key policy, the user can decipher the encryption. However, the data owner needs to know all users keys before encrypting the data, so that the encrypted text can be appropriately associated with the corresponding attributes. These KP-ABE requirements are not suitable for our scenario, in which the data owner cannot necessarily remember the keys of everyone who has access to his system so he can choose the attributes of his data.

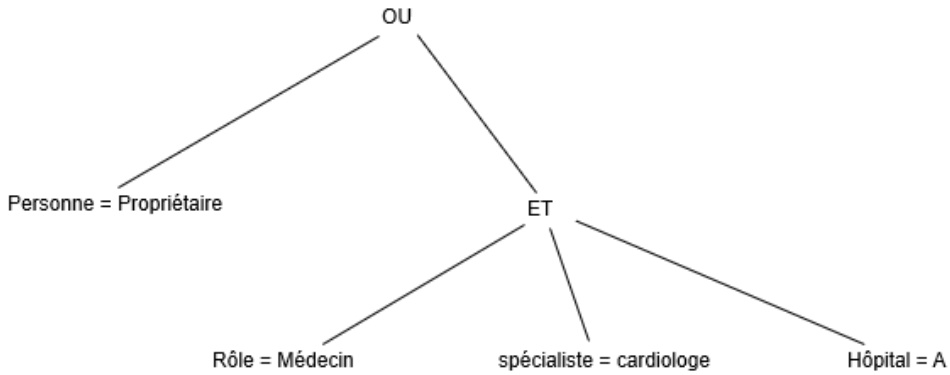


Figure 11 Scheme representing an example of a Tree policy.

2.4.2.3 Ciphertext-Policy

Therefore, Bethencourt and the others, Proposed the CP-ABE model, conceptually closer to traditional access control methods, such as RBAC, The CP-ABE scheme attaches the access policy into the encrypted text and not in the user attributes. It is more intuitive that the owner of the data specifies his policy when he encrypts his data [24]. For users, to decipher correctly the encrypted text they must have enough attributes issued by credit authorities, example in Figure 12. Generally, we use several authorities for the generation of keys but in our case, we will have a maximum of thirty users because the system is implemented directly in the hub of the patient's home, which means that the users will simply be the patient him-self, his treating doctor, and the people that he allowed into the system.

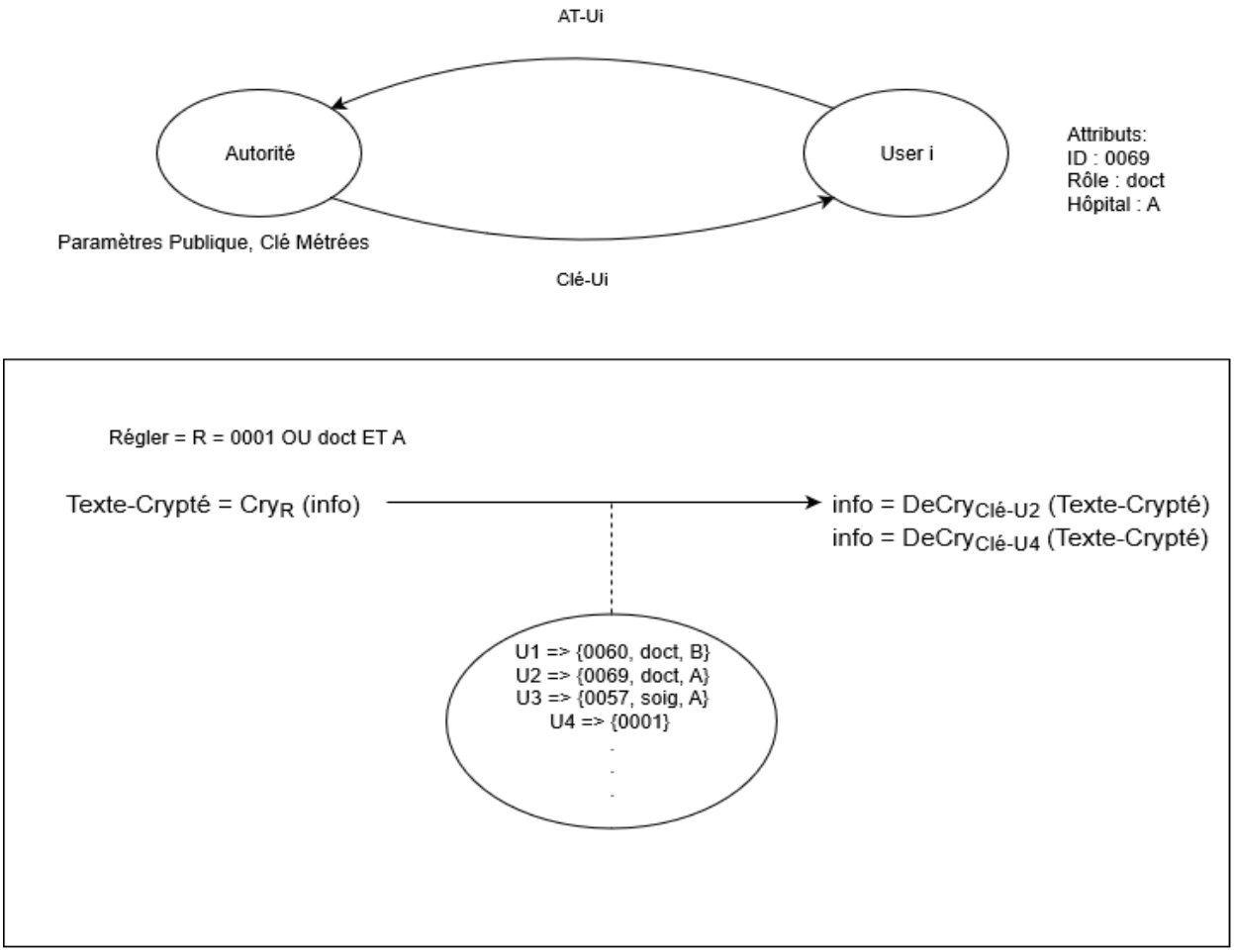


Figure 12 Scheme representing an example of a CP-ABE.

2.5 Similar work

In this section we'll be talking about similar projects that employ the same principles, In May 2018, Usama Salama et al [27]. achieve a multi-level access control solution by extending

Public Key Infrastructure (PKI) for secure authentication and utilizing Attribute-Based Access Control (ABAC) for authorization. The proposed access control system regulates access to healthcare data by defining policy attributes over healthcare professional groups and data classes classifications [27]. And in August 2018, Yang YANG et al [26]. they proposed a lightweight break-glass access control (LiBAC) system that supports two ways for accessing encrypted medical files: attribute-based access (ABE) and break-glass access. In normal situations, a medical worker with an attribute set satisfying the access policy of a

medical file can decrypt and access the data. In emergent situations, the break-glass access mechanism bypasses the access policy of the medical file using a universal key distributed from a trusted authority to allow timely access to the data by emergency medical care or rescue workers. [26]

2.6 Conclusion

To conclude this chapter, access control is the most important aspect of data security, a lot of work has been done in order to have a performant model, which ensures a high level of safety and granularity.

In this chapter, we presented a state of the art on access control models. We have studied different type of models that will be useful later for the improvement of a specific solution that will be discussed in the next chapter.

Chapitre 3: Requirements Analysis

3.1 Introduction

Based on the study conducted in the previous chapter, we begin the development of our security system, which consists of the protection of medical data and private life of patients at home. To do this, we will follow a software engineering approach by conducting needs analysis, design, realization and deployment of our system. To concretize the communication between the different system components, we will use some communication protocols dedicated to connected devices systems. In addition, the use of a development platform for the management of some aspects such as data storage and encryption is of great importance.
This chapter presents the first phase of the development of our security system. It aims to study and analyze functional and non-functional requirements as well as the designation of the equipment necessary for the implementation of such a system. Thus, an overview of the overall functioning of the system is then given, describing the "what" to do.

3.2 Problems and Objective

A home automation medical system based on IoT is used to intelligently monitor and store patient's medical data in their homes. The first problem is to ensure the privacy of patient knowing that cloud service providers are prone to mass data theft and secondly a lack of a reliable constant internet connection in our environment. This system is accessed by different users daily and for different reasons and it contains a multitude of sensitive information about the patient, so a robust flexible access control that can handle most situations is essential. The patient's data will also have to be sent outside the system (hospital or doctor etc.) so a strong encryption method is needed while also ensuring that system is simple enough to be used and managed by the patient.

The objective of this project is to create a localized context aware system that can handle most types of access and to overall secure the patients data inside and outside the system.

3.3 Requirements

The requirements that our system need to meet are divided into Functional and non-Functional features as follows:

3.3.1 Functional Requirements

The system should be able to establish the following features:

a) User Authentication.
b) IoT Authentication.
c) Data Consultation.
d) Data Encryption and Decryption.
e) Remote Control of IoT.
f) Pertinent Data Stocking.
g) User account management.

3.3.2 Non-Functional Requirements

It includes:

a) Contextual Access Control Awareness.
b) Ergonomics.
c) Users and IoT Security.

3.4 Proposed Solution

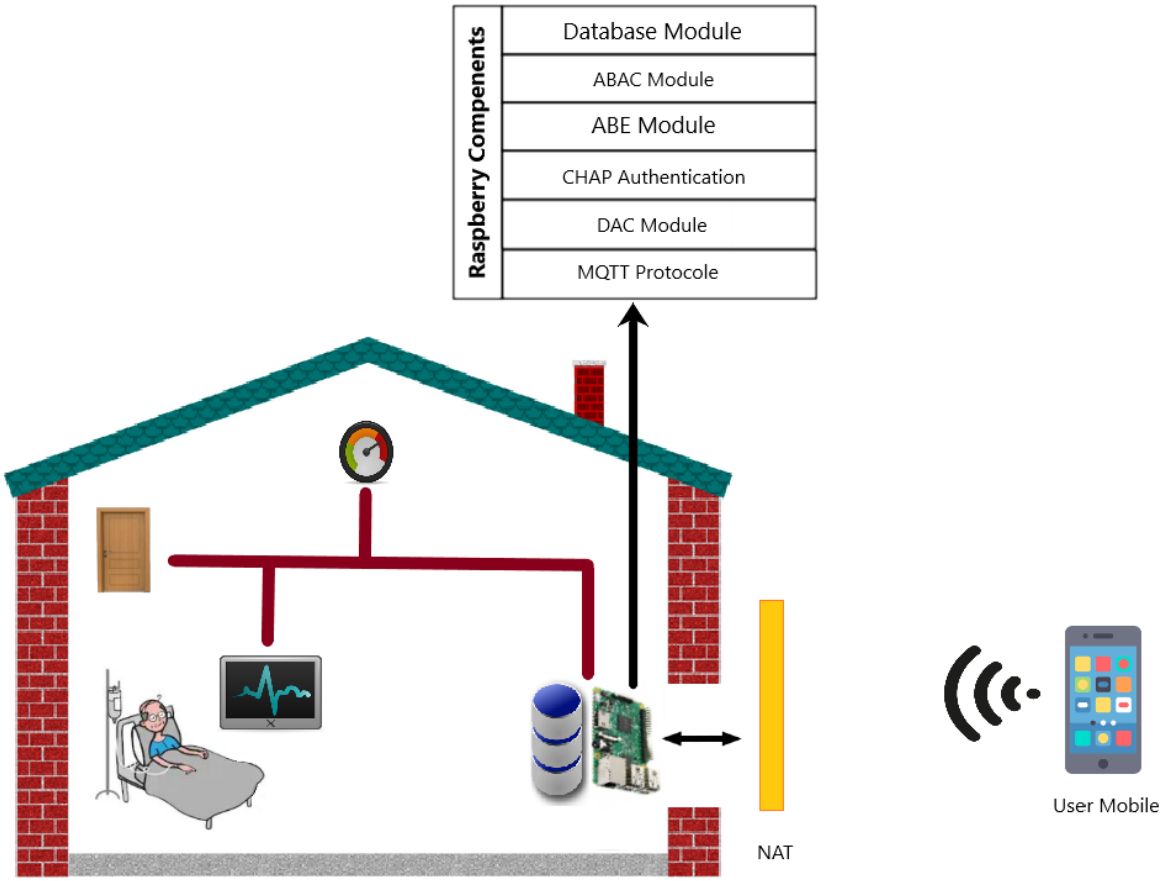


Figure 13 Proposed solution framework.

As shown in Figure 13 this proposition is based on the use of a multi-level access control between ABAC, DAC and an encryption system ABE this solution will also handle the authentication of users with CHAP and connected devices with MQTT.

The Multi-level access Control will use:

1. DAC: It’s the first level of access that will provide the owner (Responsible person) an easy and comprehensible way to manage simple access in the system based on their identity only by an ACL of all users.
2. ABAC:

   The second level of access control that manages the contextual type of access based on a multitude of attribute provided from the user, system and environment in conjunction with Policies.
3. ABE:

   The Third level of access control that primarily works on EHR (Electronic health record) files for its high level of security by encrypting those using predetermined policies and decrypting them using user attributes.

For Authentication protocols:

1. MQTT: it is used to authenticate and smooth connection between connected devices and the server
2. CHAP: it is used to authenticate users.

3.5 Static view of the system

We have employed the Use case diagram and Class Diagram to represent the static view of our system.

3.5.1 Class Diagram

The class diagram represents the conceptual architecture of the system; it generally expresses the internal structure of the system and shows the relationships between the classes making up the system. In this part, we will propose a class diagram according to our project.

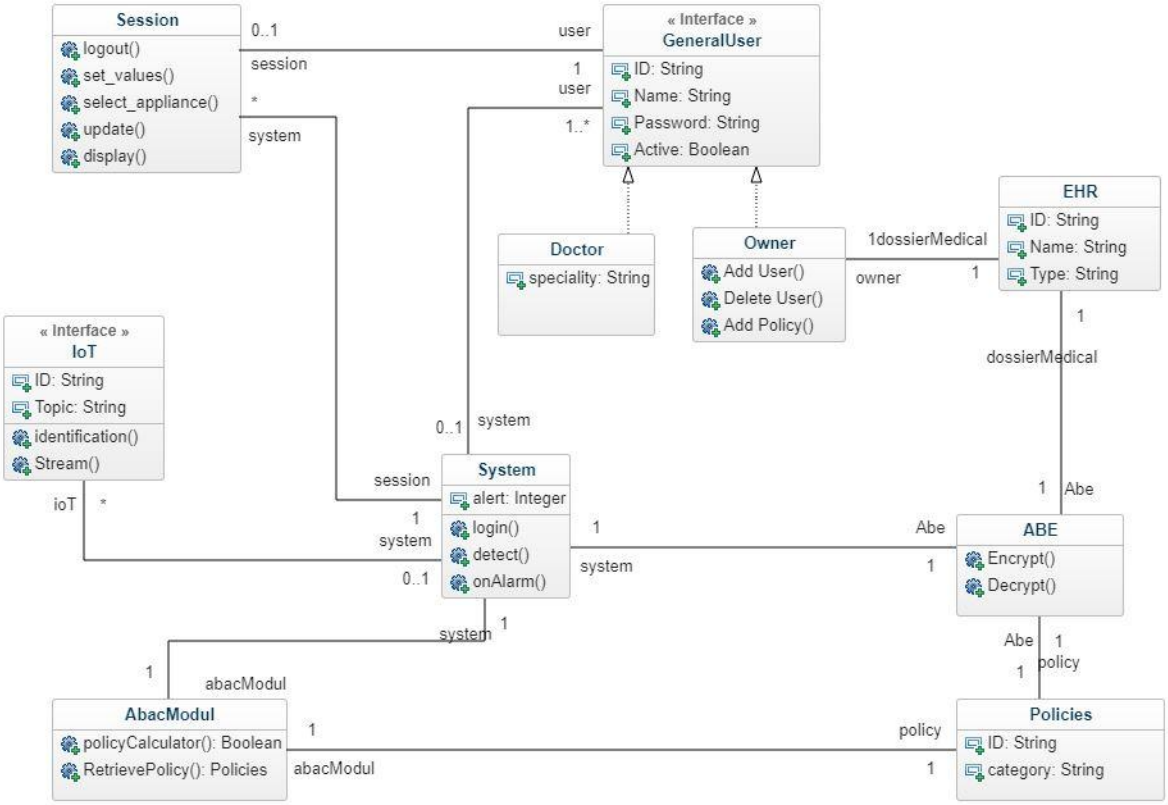


Figure 14 System Class Diagram.

3.5.2 Use Case Diagram

The static view of the functional behavior of the system is represented by the use case diagram.

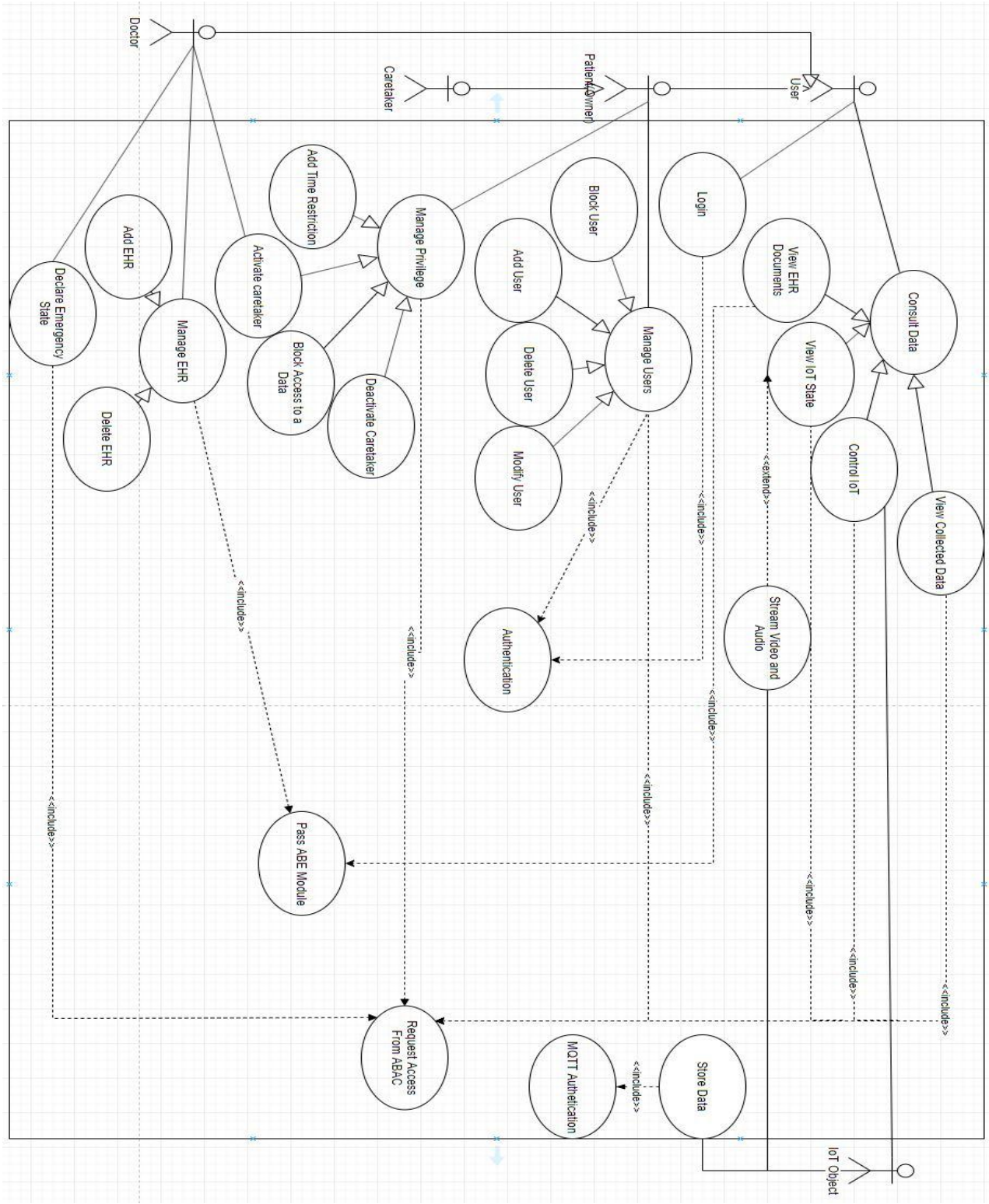


Figure 15 Use Diagram of the system.

As shown in Figure 15, the caretaker is inheriting all of the patient's (owner) functions to signify that once the caretaker role (when the owner is incapacitated or not able to take care of himself) is active it has the same level of authority as the owner.

3.6 Dynamic View of the system

The diagrams presented in this section show the dynamic view of our system and the different interactions.

3.6.1 Sequence Diagram of <<Login>>

We use the textual description of this use case to specify the interactions of system actors.

Title: Login

Résumé: It allows the user to login into the system

Actor:

- User (Mobile APP)
- Hub

Precondition:

- None

Nominal Scenario:

Alternatives Actions:

- Verify(userName) sends false : request credentials again

Post Condition:

- User session is created and ready to use

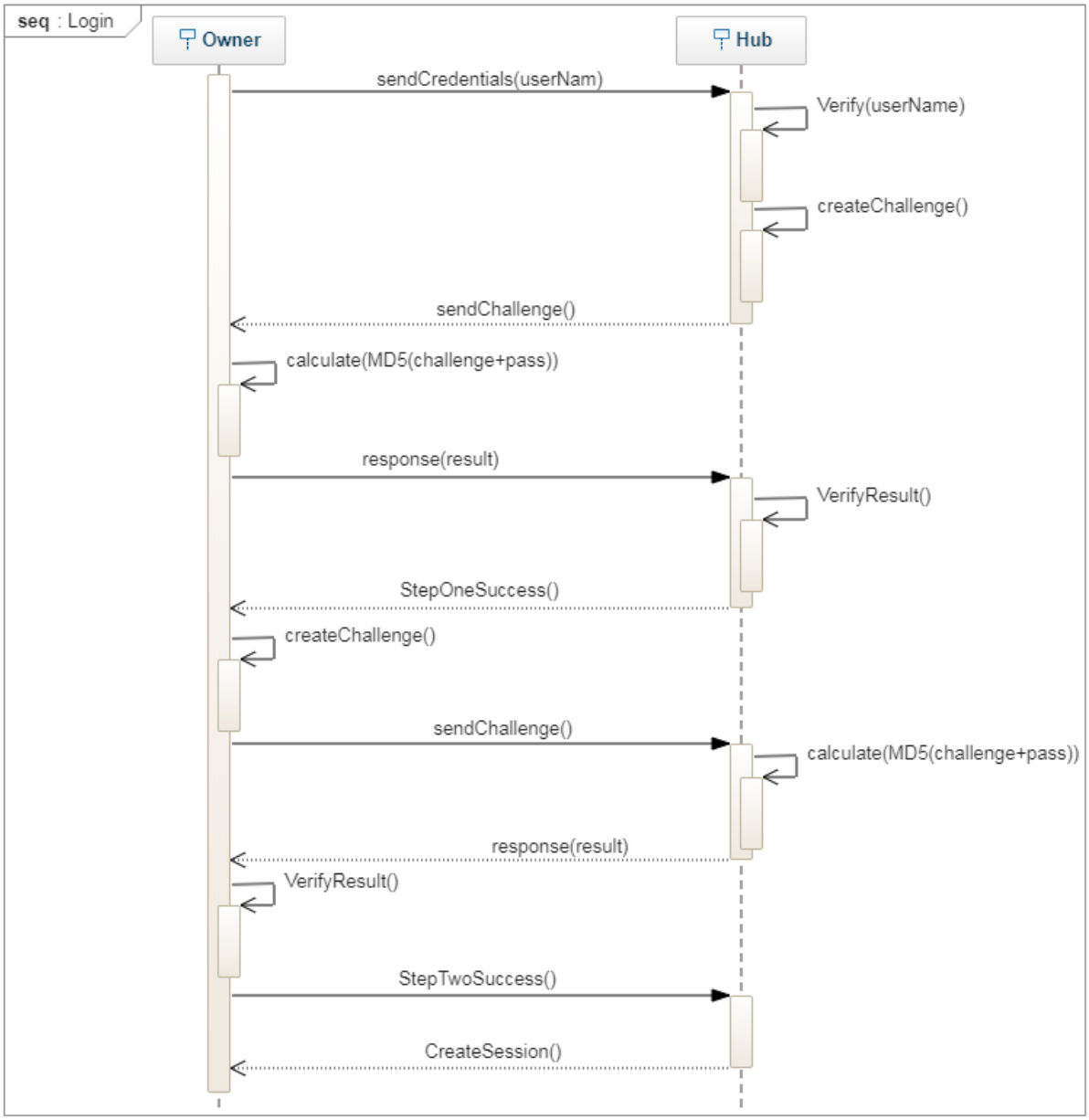


Figure 16 Sequence Diagram of Login.

3.6.2 Sequence Diagram of <<Permission Verification>>

We use the textual description of this use case to specify the interactions of system actors.

Title: Permission verification

Résumé: Consults ABAC core to check whether the user have the right to do that Action.

Actor:

- Hub
- ABAC Core
- SGBD

Precondition:

- Login
- Action Requested

Nominal Scenario:

Alternatives Actions:

- CheckDacRight() Fails : stops process and sends Deny.

Post Condition:

- Sends a Deny or Grant response.

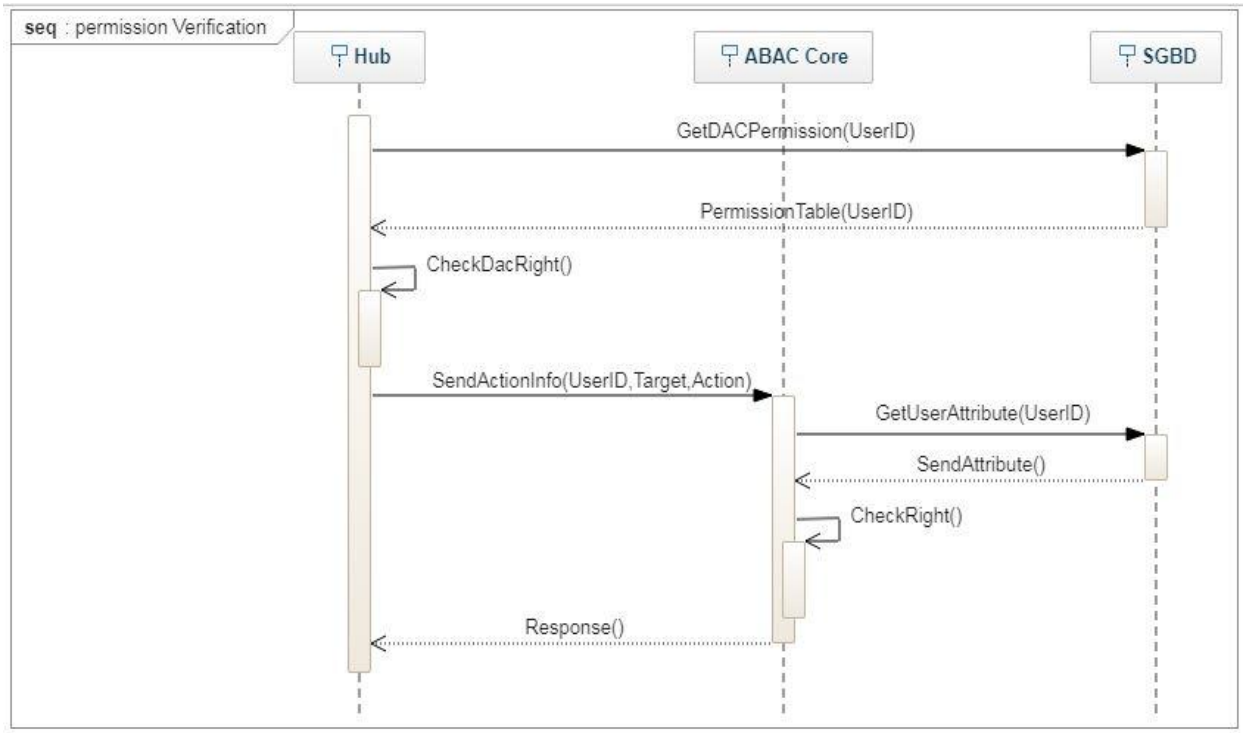


Figure 17 Sequence Diagram of Permission Verification.

3.6.3 Sequence Diagram of <<Control connected IoT>>

We use the textual description of this use case to specify the interactions of system actors.

Title: Control connected devices.

Résumé: it allows the user to send specific command to the connected device to control it accordingly.

Actor:

- User (Mobile APP)
- Hub

- ABAC Core
- SGBD
- Connected device

Precondition:

- Login

Nominal Scenario:

Alternatives Actions:

- Permission Denied: Denies the command and stops the process.

Post Condition:

- IoT stats changed according to the command

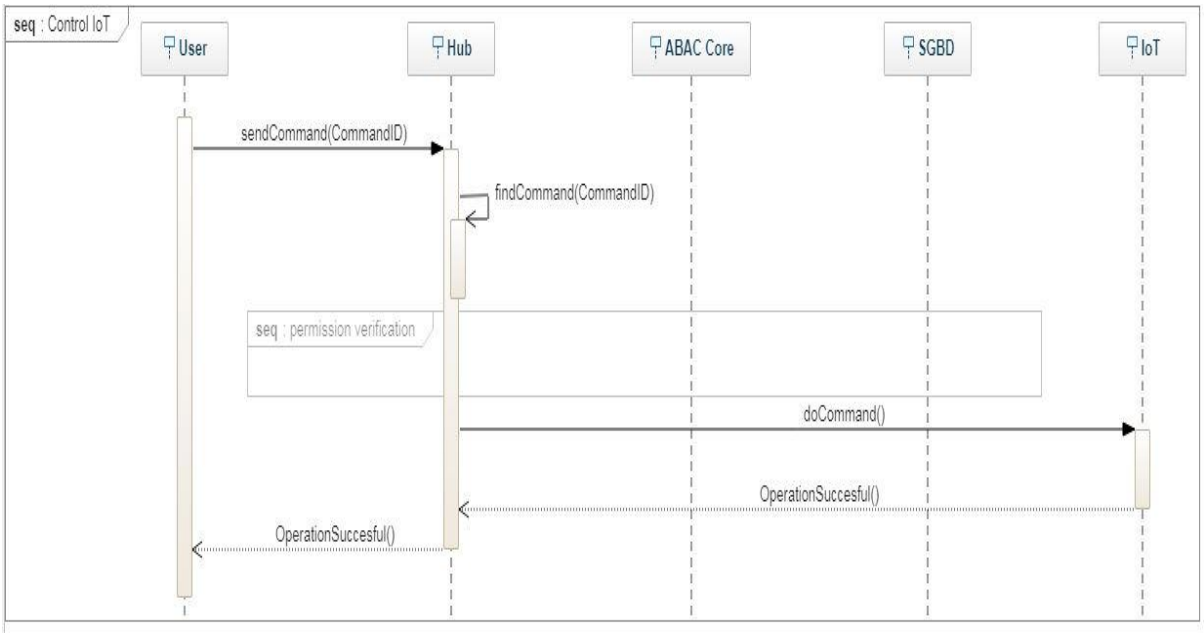


Figure 18 Sequence Diagram of Control IoT.

3.6.4 Sequence Diagram of <<Add User>>

We use the textual description of this use case to specify the interactions of system actors.

Title: Add User

Résumé: it allows the Owner to add a user to the system

Actor:

- Owner (Mobile APP)
- HUB
- SGBD
- ABAC Core

Precondition:

- Logged as Owner.
- Logged as Caretaker.

Nominal Scenario:

Alternatives Actions:

- Confirm(Password) failed : request again.

Post Condition:

- User added to the system

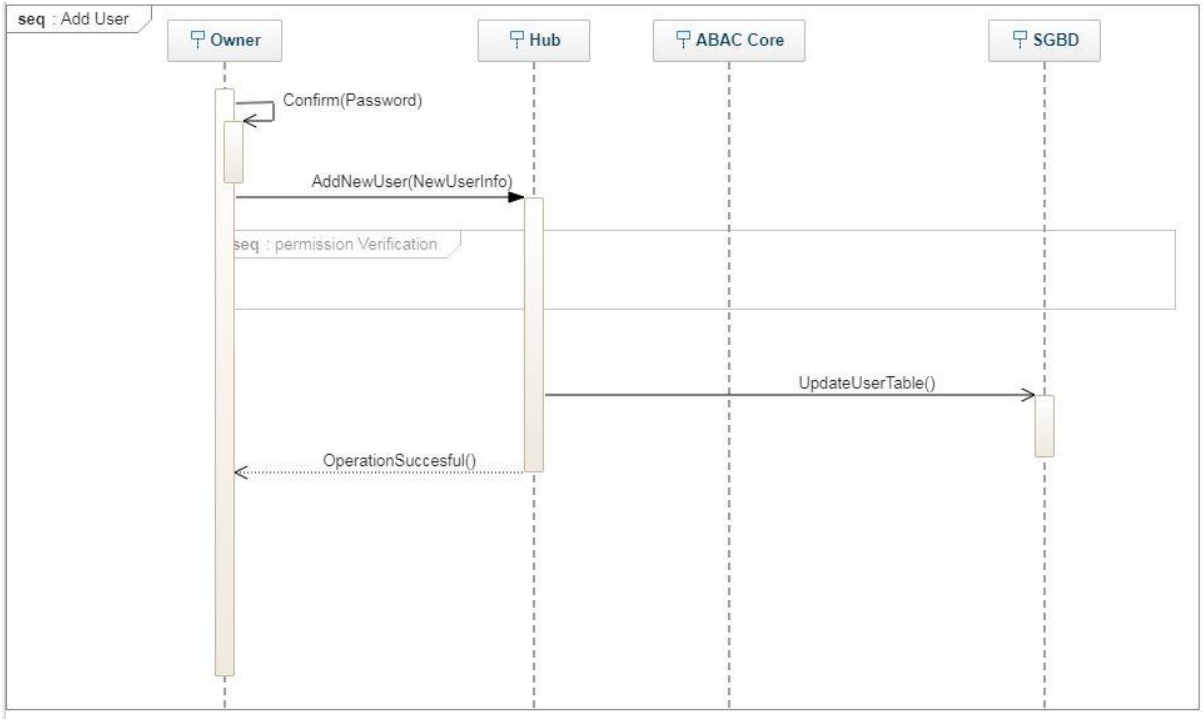


Figure 19 Sequence Diagram of Adding a User.

3.6.5 Sequence Diagram of << Add Time Restriction>>

We use the textual description of this use case to specify the interactions of system actors.

Title: Add Time Restriction

Résumé: it allows the Owner to add time restriction to the system

Actor:

- Owner (Mobile APP)
- HUB
- SGBD
- ABAC Core

Precondition:

- Logged as Owner.
- Logged as Caretaker.

Nominal Scenario:

Alternatives Actions:

- UpdateActivePolicy() fail: send already exist message and abort operation.

Post Condition:

- Added new time restriction to active rules

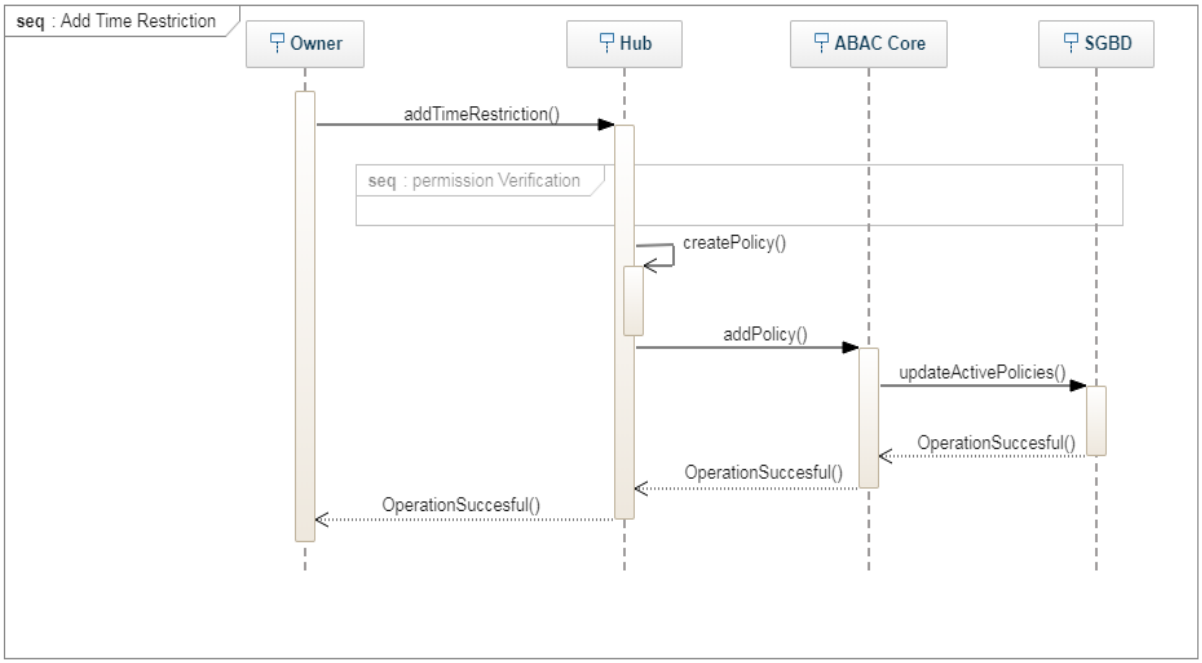


Figure 20 Sequence Diagram of Adding a Time Restriction.

3.6.6 Sequence Diagram of << Add EHR>>

We use the textual description of this use case to specify the interactions of system actors.

Title: Add EHR

Résumé: it allows adding an Electronic Healthcare Record to the system

Actor:

- Owner or Doctor
- HUB
- SGBD
- ABAC Core
- ABE Module

Precondition:

- Logged in as Doctor or Owner

Nominal Scenario:

Alternatives Actions:

- ValidateInfo() Fails : Request another file

Post Condition:

- EHR is added to the system

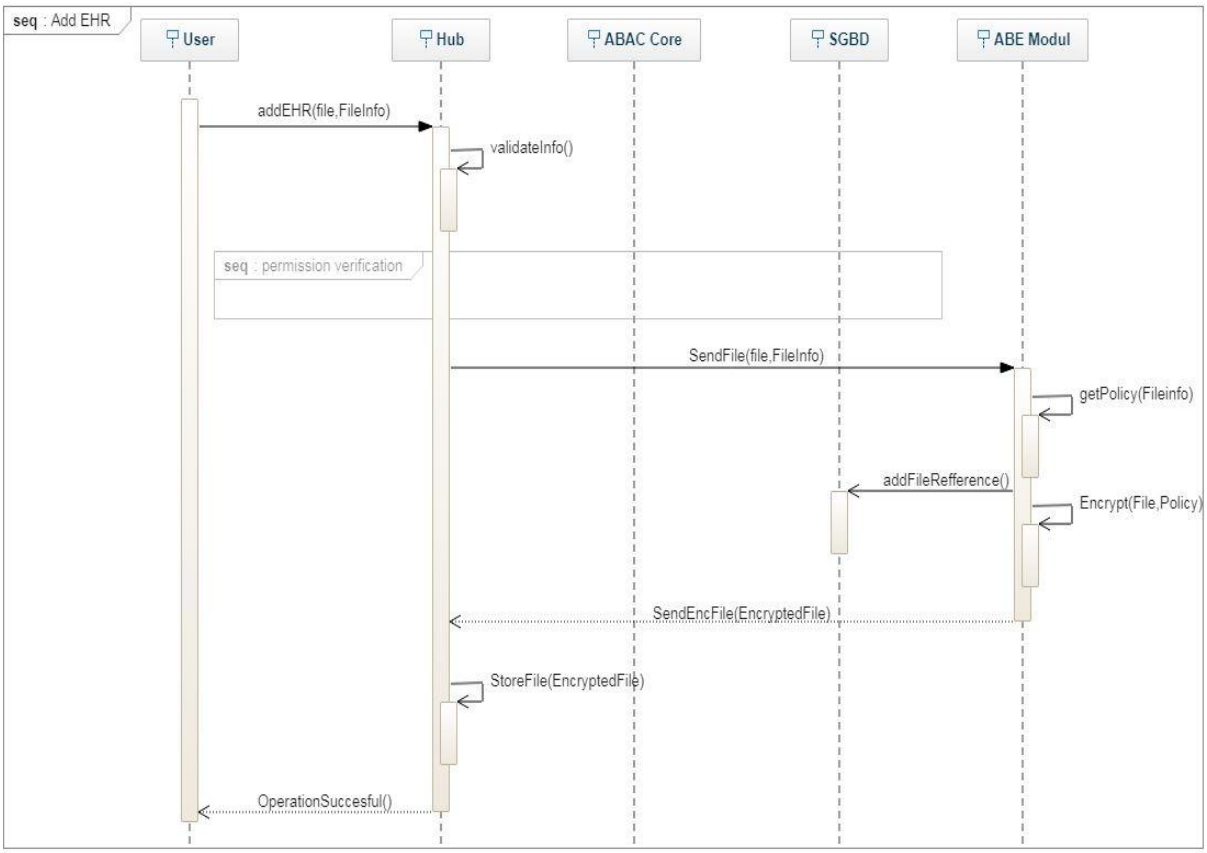


Figure 21 Sequence Diagram of Adding an HER.

3.6.7 Sequence Diagram of << IoT Authentication and Storing Data>>

We use the textual description of this use case to specify the interactions of system actors.

Title: IoT Authentication and Storing Data

Résumé: this sequence describe the process of authenticating an IoT and also how it stores data

Actor:

- HUB
- IoT

- MQTT
- SGBD

Precondition:

- None

Nominal Scenario:

Alternatives Actions:

- Validate(Id,PWD) fails : request again.

Post Condition:

- IoT Authorized
- IoT is storing Data

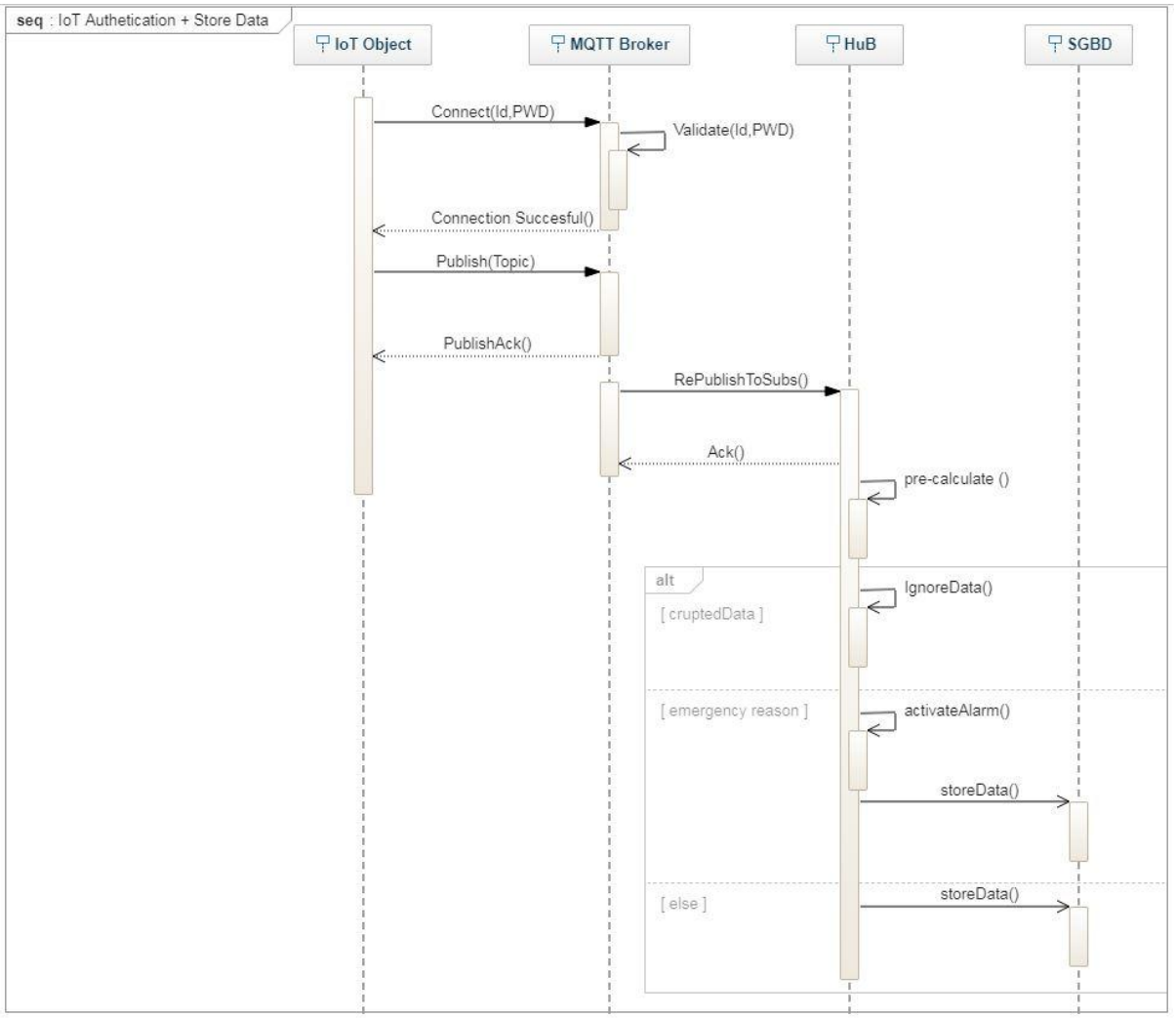


Figure 22 Sequence Diagram of IoT Authentication and Storing Data.

3.6.8 Sequence Diagram of << Emergency >>

We use the textual description of this use case to specify the interactions of system actors.

Title: Emergency

Résumé: this sequence describes how the emergency mechanism works.

Actor:

- HUB
- IoT
- User

Precondition:

- IoT Authenticated
- IoT is Publishing Data

Nominal Scenario:

Alternatives Actions:

Post Condition:

- Emergency Activated
- User Notified

Figure 23 Sequence Diagram of Emergency.

3.7 Conclusion

This chapter aims to present a new theoretical model for access control, authentication and data sharing in Healthcare IoT. Our model theoretical consists of four major security controllers summarized schematically in Figure 24 and repeated over in each iteration.

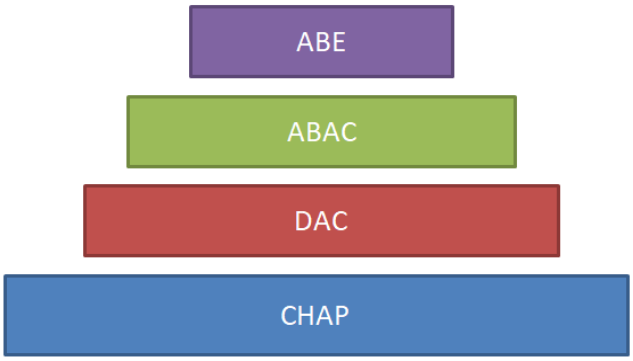


Figure 24 Proposed access control architecture.

The next chapter deals with the implementation of our theoretical framework and presents the results of the experiments, which are intended to validate the different concepts of our model.

Chapitre 4: Realization of the system

4.1 Introduction

After having realized the appropriate design for our project, we will in this chapter describe the process of the realization of our system, and that's by specifying the development environment, and a presentation of some important algorithms that describe its mechanism.

4.2 Tools and platforms

This part concerns the presentation of the tools, programming language and platforms used in this project:

4.2.1 Java

Java is a programming language and an IT platform created by Sun Microsystems in 1995. It is the underlying technology that enables the execution of state-of-the-art programs, including utilities, games, and business applications. Java is used on more than 850 million desktop computers and one billion devices worldwide, including mobile devices and TV broadcasting systems. [28]

4.2.2 Eclipse

Eclipse is a powerful development environment, that allows a smooth accurate develop in Java and C++, but offers many other possibilities thanks to a complex system of plugins,

It is a project organized into a set of software development subprojects aimed at developing a free software production environment that is extensible, universal and versatile, relying mainly on Java. [29]

Its purpose is to produce and provide tools for software development, encompassing programming activities. Its IDE, part of the project, aims to support any programming language like Microsoft Visual Studio. [29]

4.2.3 Android Studio

It is a development environment to develop Android applications. It is based on IntelliJ IDEA. (IntelliJ IDEA is a Java commercial IDE developed by JetBrains). Android Studio mainly allows you to edit Java files and configuration files for an Android application. Among other things, it offers tools to manage the development of multilingual applications and allows you to view the layout of screens on screens of various resolutions simultaneously. [30]

4.2.4 Raspberry Pi 3

The Raspberry Pi is a single-chip, ARM processor-sized Nano-computer that is intended to encourage learning of computer programming; it allows the execution of several variants of the GNU / Linux free operating system, including Debian, and compatible software. However, it also works with the Microsoft Windows OS: Windows 10 IoT Core3 and that of Google, Android Pi.

The Raspberry hardware has gone through a number of variations in terms of peripheral device support and memory capacity. Every new addition comes with a little improvement in terms of design where advance features are added in the device so it can do as many function as a regular computer. However, it has a leg over desktop PC when it comes to cost, size and power consumption. [31]

4.2.5 SQLite

SQLite is a library written in C language that offers a relational database engine accessible by the SQL language; it implements much of the SQL-92 standard and ACID properties.

Its particularity is not to reproduce the usual client-server schema but to be directly integrated into the programs. The entire database (declarations, tables, indexes and data) is stored in a file independent of the platform.

It is the most used database engine in the world, thanks to its use in many consumer software like Firefox, Skype, Google Gears, it is also very popular on embedded systems, especially on most modern smartphones and tablets. IOS, Android and Symbian mobile operating systems use it also as embedded database system.[32]

4.2.6 Challenge handshake authentication protocol (CHAP)

CHAP is a protocol used to authenticate both ends of a communication link. Such authentication protocol is intended for use primarily by hosts and routers that connect to a PPP (point-to-point protocol) network server, but might be applied to dedicated links as well. It uses a challenge response mechanism based on a nonce (a sort of random number used only once) and a shared secret. This secret is only known to the entities that want to authenticate each other. Mutual authentication can be achieved by running the protocol in both directions. In this case, the shared secret can be the same for both directions. [33]

4.2.7 Cp-ABE

This software is a Java realization for "ciphertext-policy attribute based encryption" (CP-ABE). That provides a set of programs implementing a ciphertext-policy attribute-based encryption scheme. It uses the jPBC library for the algebraic operations. [34]

4.2.8 MQTT

MQTT stands for MQ Telemetry Transport. It is a simple and lightweight publish/subscribe messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimize network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the emerging “machine-to-machine” (M2M) or “Internet of Things” world of connected devices, and for mobile applications where bandwidth and battery power are at a premium. [35]

4.2.9 Node Red

IBM Watson uses the Node-Red visual tool to program the IOT application and for connecting hardware devices, APIs, and online services as part of the IoT.

It provides a browser-based editor that makes it easier to connect streams using the wide range of nodes in the palette that can be deployed at its runtime with one click. Streams created in Node-RED are stored using JSON that can be easily imported and exported for sharing with others. JavaScript functions can be created in the editor using a rich text editor. An integrated library allows you to save useful functions, templates, or streams for reuse [40].

4.3 Description of implemented Access Control system

In this section, we are going to describe and explain how the most of the important functions work

4.3.1 Policies

In our solution, we have divided the policies into 3 different categories that are targeted for ABAC, ABE and a list of models to integrate after the system has started.

4.3.1.1 ABAC Policies

We have chosen the XML approach to define our policies since it allows us to recreate it into an arbitrary tree that makes calculating the outcome efficient; as seen in the Figure 25 the policy is split into categories(Camera, Door etc.) and sub categories(READ,WRITE) to define each access type to every IoT object.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<policy>
    <camera>
        <read>
            <and>
                <or>
                    <and>
                        <equal>
                            <att>utilisateurs.role</att>
                            <val>doctor</val>
                        </equal>
                        <supequal>
                            <att>system.alert.1</att>
                            <val>1</val>
                        </supequal>
                    </and>
                    <and type="Permit">
                        <equal>
                            <att>utilisateurs.ID</att>
                            <val>231649</val>
                        </equal>
                        <subequal>
                            <att>system.time</att>
                            <val>2000-01-01 12:12:12</val>
                        </subequal>
                        <infequal>
                            <att>.datetime('now')</att>
                            <val>2001-01-01 12:12:12</val>
                        </infequal>
                    </and>
                </or>
            </and>
        </read>
    </camera>
```

Figure 25 Policy structure.

Even though policies are good at expressing access right, they are hard to later on update to add additional right or deny such right so we defined our policies in such a way that it allows future changes or updates to them as seen in Figure 26 and that's by sockets where to add a DENY or PERMIT statement as shown in Figure 27:

Rule = ((permit-policy)OR ,deny-policy)AND

This also gives a priority to DENY rules for better security.

```
    <door>
        <read>
            <and>
                <or>
                </or>
                <and type="deny">
                    <equal>
                        <att>utilisateurs.role</att>
                        <val>user</val>
                    </equal>
                </and>
            </and>
        </read>
    </door>
</policy>
```

Figure 26 Policy model.

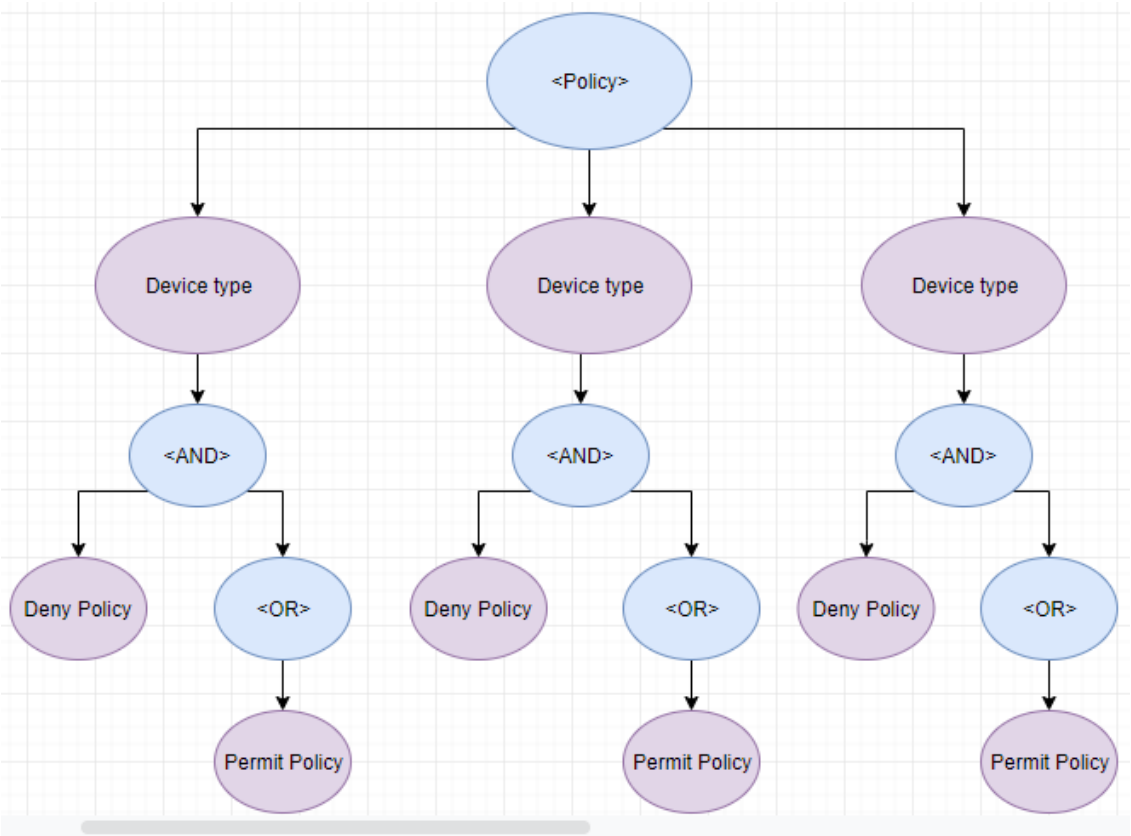


Figure 27 Policy.xml architecture

4.3.1.2 Models List

This is a list containing pre-determined empty policies that allows the user to add additional rules to the policies file as seen in Figure 30 such as time restriction, locking an IoT object or granting access as seen in Figure 31 as example of what type of rules can be added.

These rules can also be deleted by storing them as Active rules in the SGBD and deleting once expired as seen in Figure 29 or just outright deleting them as seen in Figure 28.

```
public void removePolicy(Map<String, String> m) throws Exception {
    Statement stat = connection.createStatement();
    stat.executeUpdate("DELETE FROM activerules WHERE map = \""
            + warppMap(m) + "\"");
    DOMUtils.deleteXML(mainDoc.getFirstChild(), DOMUtils.creatXMLNode(m));
    activeRules.remove(m);
    stat.close();
    DOMUtils.xMLPrinter(mainDoc);
}
```

Figure 28 Remove policy code.

```
public void CleanExpired() throws Exception {
    SimpleDateFormat dateFormat =new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    Date date = new Date(),dateMap;
    ArrayList<Map<String, String>> deletItem = new ArrayList<Map<String, String>>();
    Statement stat = connection.createStatement();
    Node node;
    for (Map <String, String> map : activeRules) {
        dateMap = dateFormat.parse(map.get("ETIME"));
        if(dateMap.before(date)) {
            node = DOMUtils.creatXMLNode(map);
            stat.executeUpdate("DELETE FROM activerules WHERE map = \""
                    + warppMap(map) + "\"");
            deletItem.add(map);
            DOMUtils.deleteXML(mainDoc.getFirstChild(), node);
        }
    }
    activeRules.removeAll(deletItem);
    stat.close();
    DOMUtils.xMLPrinter(mainDoc);
}
```

Figure 29 Removing expired rules.

```
//add pri-defined polices to policies.xml                (we don't need the object ID but the class ID)
public void addPolicy(Map<String, String> m) throws Exception {
    Node newNode = DOMUtils.creatXMLNode(m);
    DOMUtils.trimEmptyTextNodes(newNode);
    NodeList nodes = mainDoc.getFirstChild().getChildNodes();
    for(int i=0;i<nodes.getLength();i++) {
        if(nodes.item(i).getNodeName().equals(getObjectType(m.get("OBJECT")))) {
            NodeList nodes1 = nodes.item(i).getChildNodes();
            for(int j=0;j<nodes1.getLength();j++) {
                if(nodes1.item(j).getNodeName().equals(m.get("R_W"))) {
                    NodeList nodes2 = nodes1.item(j).getChildNodes();
                    for(int k=0;k<nodes2.getLength();k++) {
                        if(nodes2.item(k).getNodeName().equals("and")) {
                            if(((Element)newNode).getAttribute("type").equals("deny")) {
                                Node n = mainDoc.importNode(newNode, true);
                                nodes2.item(k).appendChild(n);
                            }else if(((Element)newNode).getAttribute("type").equals("Permit")) {
                                NodeList nodes3 = nodes2.item(k).getChildNodes();
                                for(int l=0;l<nodes3.getLength();l++) {
                                    if(nodes3.item(l).getNodeName().equals("or")) {
                                        Node n = mainDoc.importNode(newNode, true);
                                        nodes3.item(l).appendChild(n);
                                    }
                                }
                            }else throw new Exception("invalid Policy Model type");

                        }
                    }
                }
            }
        }
    }
    Statement statement = connection.createStatement();
    statement.executeUpdate("INSERT INTO main.activerules (map) VALUES (\""+warppMap(m)+"\")");
    DOMUtils.xMLPrinter(mainDoc);
    activeRules.add(m);
    statement.close();
}
```

Figure 30 Adding Policy according to a selected model.

```
<?xml version="1.0" encoding="UTF-8"?>
<Model>
    <TimeRes>
        <and type="Permit">
            <equal>
                <att>utilisateurs.ID</att>
                <val>IDLOCATION</val>
            </equal>
            <subequal>
                <att>system.time</att>
                <val>STIME</val>
            </subequal>
            <infequal>
                <att>.datetime('now')</att>
                <val>ETIME</val>
            </infequal>
        </and>
    </TimeRes>
    <Lock>
        <and type="deny">
            <equal>
                <att>utilisateurs.role</att>
                <val>USERROLE</val>
            </equal>
        </and>
    </Lock>
</Model>
```

Figure 31 Model types.

4.3.1.3 Alarm management policies

To easily maintain and update our alarm conditions and to also make them easy to read and handle we have opted to put them in an XML format as shown in Figure 32, as it allows better parsing and fast calculating.

Therefore, the file is divided into different level (in this case 3) and each level holds the condition to activate; only one can be active at a time with higher level holding more significance.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Alarm>
    <l3>
        <or>
            <equal>
                <att>heart.state.281640</att>
                <val>0</val>
            </equal>
        </or>
    </l3>
    <l2>
        <or>
            <infequal>
                <att>heart.state.281640</att>
                <val>50</val>
            </infequal>
            <supequal>
                <att>heart.state.281640</att>
                <val>170</val>
            </supequal>
        </or>
    </l2>
    <l1>
        <or>
            <infequal>
                <att>firesensor.temp.251640</att>
                <val>5</val>
            </infequal>
            <supequal>
                <att>firesensor.temp.251640</att>
                <val>45</val>
            </supequal>
        </or>
    </l1>
</Alarm>
```

Figure 32 Alarm policy.

4.3.1.4 CP-ABE Policies

We used a different approach to handle ABE policies unlike ABAC policies since their main purpose is to ensure high level of encrypted access, which makes the policies more straightforward and doesn't allow modification, so all ABE policies are pre-determined and stored in the SGBD as shown in Figure 33.

Rules are expressed as such:

Rule = (Attribute = Value, Attribute2= Value …. AttributeN=Value) (N-x)of(N)

If (N-x) parameter is Right the answer is right where n > x > 1.

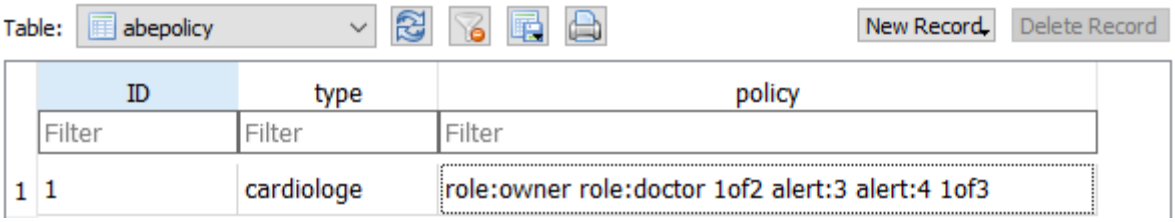


Figure 33 ABE Policy example.

4.3.2 Decision-making

In this section we will be describing how the decision making process is done starting from DAC to ABAC and ABE.

4.3.2.1 DAC

It's the first level of decision making process that makes sure the user have the right to "demand" such action before going further and checking if he has the right to "perform" such actions as seen in Figure 34. These rights are not mandated by Policies but by a DAC table defining each user's basic rights that can be taken away or given by the Owner as seen Figure 35, and these rights cannot be delegated.

```
public Boolean dacCalculator(String IDUser, String IDObject) throws Exception {

    Statement statement = connection.createStatement();
    ResultSet res = statement.executeQuery("SELECT "+IDObject+" FROM dac WHERE utilisateurs = "+IDUser);
    if (res.isClosed()) {
        throw new Exception("User ID or object ID incorrect");
    }
    Boolean boll = res.getBoolean(1);
    statement.close();
    return boll;
}
```

Figure 34 DAC layer.

```
//the function that change a user permission in DAC
public boolean denyOrAcceptDAC(String IDUser, String IDObject, boolean decision) throws Exception {

    try {
        Statement stat = connection.createStatement();
        stat.executeUpdate("UPDATE dac SET \""+IDObject+"\"= "+((decision == true)? "1" : "0" )+" WHERE \"_rowid_\"="+IDUser);
    } catch (SQLException e) {
        throw new Exception("Error user "+IDUser+" or object "+IDObject+" not fund");
    }return false;
}
```

Figure 35 DAC layer control.

4.3.2.2 ABAC

The second level of decision making that makes sure the user have right to "perform" the demanded Action, it works in conjunction with ABAC Policies file, collected environment attributes, User Attribute, Targeted Object Attribute as seen in Figure 36.

```
public Boolean Permission(String IDUser, String IDObject, String r_w) throws Exception {
    if(!dacCalculator(IDUser, IDObject)) {
        return false;
    }
    Boolean res;
    NodeList nodes = mainDoc.getFirstChild().getChildNodes();
    for(int i=0;i<nodes.getLength();i++) {
        if(nodes.item(i).getNodeName().equals(getObjectType(IDObject))) {
            NodeList nodes1 = nodes.item(i).getChildNodes();
            for(int j=0;j<nodes1.getLength();j++) {
                if(nodes1.item(j).getNodeName().equals(r_w)) {
                    NodeList nodes2 = nodes1.item(j).getChildNodes();
                    for(int k=0;k<nodes2.getLength();k++) {
                        if(nodes2.item(k).getNodeType() == Node.ELEMENT_NODE) {
                            res = policyCalculator(nodes2.item(k), IDUser, IDObject);
                            if(res != null) return res;
                            throw new Exception("inconue exeption !!!");
                        }
                    }
                }
            }
        }
    }
    return true;
}
```

Figure 36 Permission decision code.

It does that by using a policy calculator after parsing the Policies file and acquiring the arbitrary tree for the targeted object then it starts recursively going through each node (and, or, values, equals etc.) of the tee and apply the proper treatment and give the final result of True or False Statement. As seen in Figure 37.

```
//Recursively loop through and calculate out one policy from the XML file
public Boolean policyCalculator(Node node, String IDUser, String IDObject) throws Exception{
        if(node.getNodeType() == Node.ELEMENT_NODE){
            if(node.getNodeName().equals("and")) {
                NodeList nl=node.getChildNodes();
                Boolean fRes = true;
                int j;
                for(j=0;j<nl.getLength();j++) {
                    Boolean res = policyCalculator(nl.item(j), IDUser, IDObject);
                    if(res != null) {
                        fRes = fRes && res;
                        System.out.print(" and ");
                    }

                }
                if(j == 0) throw new Exception("invalid policy: invalide \"and\" statement");
                else return fRes;
```

Figure 37 Policy outcome code.

4.3.2.3 CP-ABE

The third level of decision making that only intervenes when a user is try to Write and read an Electronic Health Record (EHR) and that's by storing every HER in an encrypted manner depending on the type of ABE policy chosen when it was added as seen in Figure 38.

```
//the map => {(docname, type)}
public void addEHR(byte [] file, Map<String, String> m) throws Exception {
    Statement s = connection.createStatement();
    //check that the doctor specialty exist
    ResultSet res = s.executeQuery("SELECT count(*) FROM abepolicy WHERE type = \""
            +m.get("docSpeciality")+"\"");
    if(res.getInt(1) != 1) throw new Exception("invalid doctor speciality");
    //add the new EHR
    s.executeUpdate("INSERT INTO ehr(\"docName\",\"docSpeciality\") VALUES (\""
            +m.get("docname")+"\" , \""+m.get("docSpeciality")+"\")");
    res = s.executeQuery("SELECT last_insert_rowid()");
    String name = "Resource/"+res.getString(1);
    //Crypte and save the file
    res = s.executeQuery("SELECT policy FROM abepolicy WHERE type = \""+m.get("docSpeciality")+"\"");
    cpabe.enc(pubfile, res.getString(1), file, name);

}
```

Figure 38 Adding EHR to the system.

When a user tries to read an EHR they will attempt to decrypt it using their own attribute as seen in Figure 39 and depending on what they are the file can either be decrypted or not.

```
public byte[] getEHR(String encfile, String att) throws Exception {
    // add system attribute to att before creating a key
    byte[] privetKey = cpabe.keygen(pubfile, mskfile, att);
    return cpabe.dec(pubfile, privetKey, encfile);
}
```

Figure 39 Acquiring EHR.

4.4 Node-Red

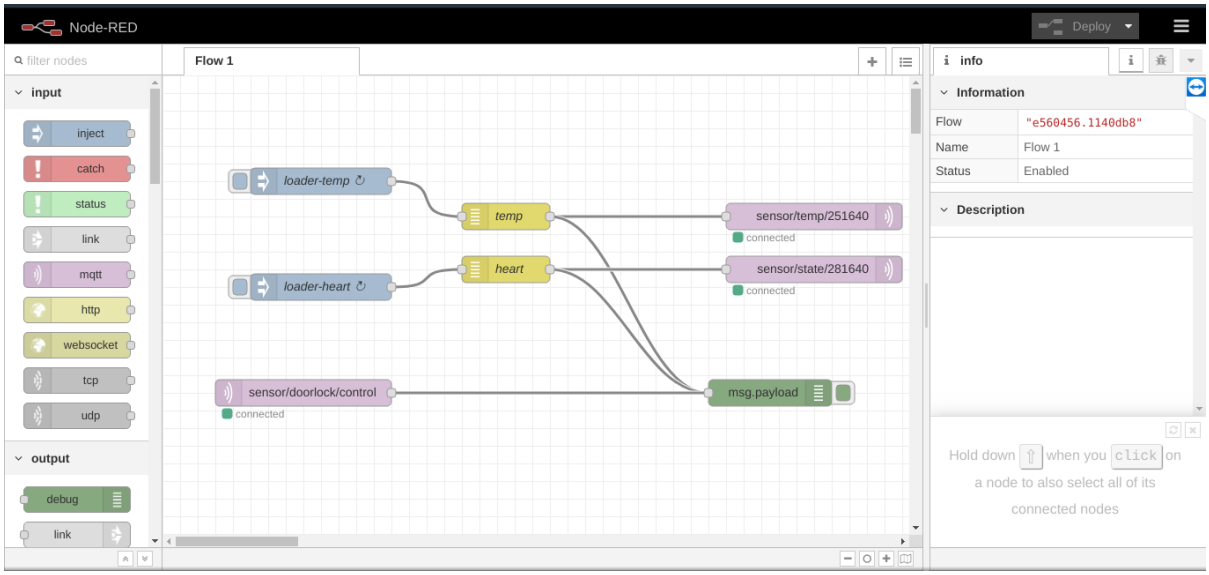


Figure 40 Device simulation with Node-Red.

As we see in Figure 40, With Node-RED we managed to simulate the different connected devices required using MQTT communication protocol with added password authentication to test our lightweight multi-level Access Control system.

4.5 System files security

To ensure that the files used by our system such as policy files are not accessed or modified maliciously, we have opted to encrypt the partition in raspberry that holds the system files.

4.6 Mobile application

In this section, we will present the main pages that consist our mobile app:

4.6.1 Home Page

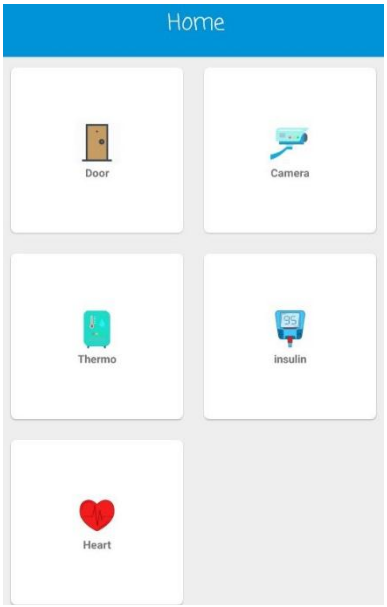


Figure 41 Home Page.

After authenticating, the user will be welcomed with main page of the mobile app as presented in Figure 41,

It contains the different IoT devices monitored by the system, the user can demand access to each one of them and depending on his attributes, he will either get access and receive live feed of the collected data or get an "Access Denied" toast message to show that he doesn't have the rights for it.

The response may differ in an emergency state to allow access to more users

4.6.2 HER page

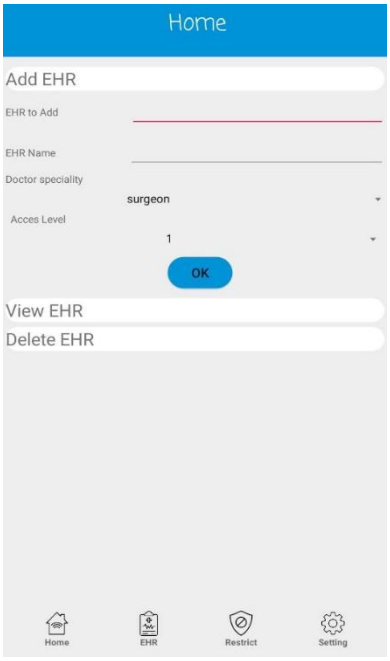
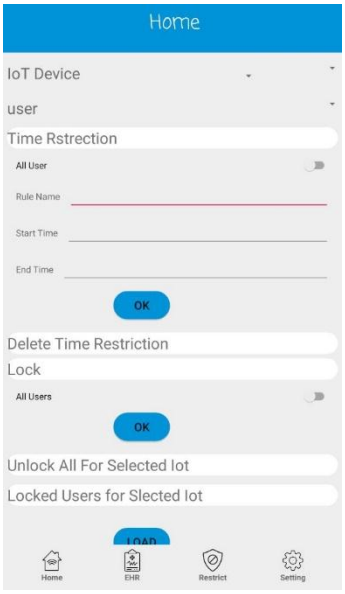


Figure 42 EHR Page.

As shown in Figure 42, this is the page where the user with enough rights can add or delete HER documents as well as view them, which work on the ABE level.

The response may differ in an emergency state to allow viewing access to more users.

4.6.3 Restriction page

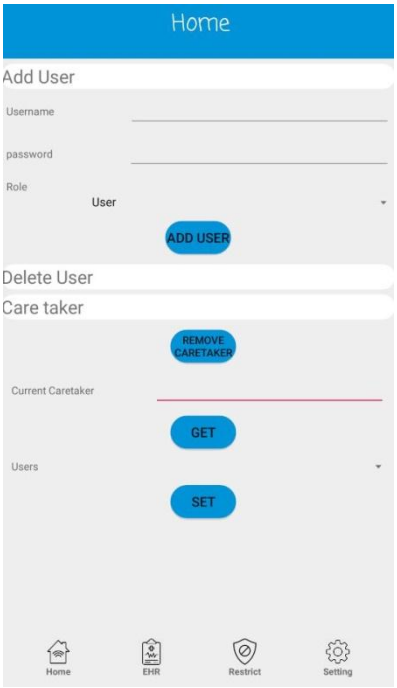


The Figure 43 presents the page where the owner of the system or a user with enough rights can add or retract different type of restrictions, namely time restrictions which work on the policies level.

He can also lock or unlock access to a chosen type of IoT which work on the DAC level.

Figure 43 Restriction Page.

4.6.4 Setting page



As we see in Figure 44, the setting page is where the owner or a user with high enough rights has the ability to either add or delete users, and also to view the current caretaker and to remove or chose a different one.

Figure 44 Setting page.

4.6.5 SQL injection countermeasure

In each of the previous pages, we are trimming all input fields that sends text fields to our server to avoid any type of SQL injection.

4.7 Result

After realization, implementation, and multiple tests we end up with the following results in Figure 45.

	RBAC	ABAC	Multi-Level Access	LiBAC	Our solution
Role-based permission assignment	O	O	O	O	O
Flexible permission assignment	O	O	O	O	O
Context Awareness	X	O	O	O	O
Granularity	X	O	O	X	O
Cryptic Data	X	X	X	O	O
Authentication	X	X	O	X	O
Flexibility to add more control	X	X	X	X	O
Group-level access control	X	O	O	O	O
Lightweight	X	X	X	O	O
Cloud ready	X	X	O	O	X
Easy to implement	O	X	X	X	X

Figure 45 Comparison table.

We must also emphasize that the tests and result are only valid in the intended work environment that is a Localized healthcare smart home.

4.8 Conclusion

Without a strong security the use of IoT in any field of application would have undesirable consequences. Our approach is a way to improve the security of the protocol presented in the article [25]. The main contribution is the creation of a lightweight version of ABAC that ensure the granularity and the context awareness of the system as a whole, because ABAC is not suitable for all type of data we add ABE cryptography for the most sensitive data (EHR). Also establishing secure access includes setting up a good authentication protocol that will ensure the identity of the connecting user by using CHAP.

In contrast, our approach is not meant to be used in the cloud or any environment with high number of users because of the high intensity traffic that would hinder our system for working efficiently as its intended for a low traffic, low energy environment.

This solution is also meant to be used in a none de-centralized structure because the ABE key management is done within the system rather than using an external trusted authority.

General Conclusion

The emergence of the IoT offers great potential for the development of new services and applications connecting the physical world to the virtual world especially in the Healthcare world and monitoring patients from their home.

The problem addressed in this work is how to offer a secure data management system for the collected health information from IoT and also the personal EHR of the patients in their home. To do this, we have proposed a lightweight multi-level Access Control system relaying on four main security system; a lightweight version of ABAC that work as the core of the system, a ABE module for enhancing the security level of the HER data, a DAC module to enforce the owner roles, and finely the CHAP module to ensure the true identity of the user. This model is intended to be used in a localized data system on the patient's home .We then summarize our achievements and future research.

Future Improvement

Designing an effective security system for healthcare smart home is still a more open area of research, the perspective of our work is to further improve and optimize our access control system to be even more practical and fast and more secure by improving the notion of attributes and acquiring a more efficient ABE encryption algorithm.

Even though our system uses CHAP as its authentication system, we intend to upgrade it into a more secure and efficient version that supports Single-Sing-On Protocol like Kerberos.

Moreover our system lacks the ability to auto identify every model of IoT device for poor compatibility between all the different models so our goal is to make a system that can identify and connect with each different model.

In addition, as a final improvement it is ideal to create a platform to hold all the owners of these systems in an easy to access database for different Doctors to allow and facilitate the discovery and connection to these homes.

Bibliography

[1] KNUD LASSE LUETH "STATE OF THE IOT 2018: NUMBER OF IOT DEVICES NOW AT 7B – MARKET ACCELERATING" IN: HTTPS://IOT-ANALYTICS.COM/AUTHOR/KNUD-LASSE-LUETH/, PAGE CONSULTED ON 24/02/2019.

[2] COETZEE, & EKSTEEN, "THE INTERNET OF THINGS-PROMISE FOR THE FUTURE?", PAPER PRESENTED AT THE IST-AFRICA CONFERENCE PROCEEDINGS, 2011.

[3] H. SUNDMAEKER, ET AL., "VISION AND CHALLENGES FOR REALISING THE INTERNET OF THINGS", CERP-IOT, 2010.

[4] J. GUBBI ET AL., "INTERNET OF THINGS (IOT): A VISION, ARCHITECTURAL ELEMENTS, AND FUTURE DIRECTIONS", ELSEVIER B.V, 2013.

[5] F. FIROUZI, ET AL., "FROM EDA TO IOT EHEALTH: PROMISE, CHALLENGES, AND SOLUTIONS", IEEE, 2018, P: 2965 - 2978.

[6] DARSHITA PATEL, "INTERNET OF THINGS AND POWERFUL IOT GATEWAYS" IN: https://medium.com/@darshipatel/internet-of-things-and-powerful-iot-gateways-a1673cba6cb9, PAGE CONSULTED ON 08/03/2019.

[7] MOUSER ELECTRONICS, "ALL THINGS IOT ", IN: HTTP://WWW.MOUSER.COM/EMPOWERING-INNOVATION/ALL-THINGS-IOT, PAGE CONSULTED ON 08/03/2019.

[8] KNUD LASSE LUETH "STATE OF THE IOT 2018: NUMBER OF IOT DEVICES NOW AT 7B – MARKET ACCELERATING", IN: https://iot-analytics.com/author/knud-lasse-lueth/, PAGE CONSULTED ON 11/03/2019.

[9] OBJETCONNECTE, COMPARATIF DES PLATEFORMES IOT, IN: HTTPS://WWW.OBJETCONNECTE.COM/COMPARATIF-PLATEFORME-IOT/, PAGE CONSULTED ON 11/03/2019.

[10] Feifei Shi et al, "A Survey of Data Semantization in Internet of Things", MDPI

2018.

[11] Pallavi Sethi and Smruti R. Sarangi "Internet of Things: Architectures, Protocols, and Applications", Journal of Electrical and Computer Engineering, 2017.

[12] Cisco, "Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are", Cisco, 2015.

[13] Sebastian Raff," MQTT Wiki", IN: https://github.com/mqtt/mqtt.github.io/wiki, page consulted on 16/03/2019.

[14] England I, Stewart D, Walker S,"Information technology adoption in health care: when organisations and technology collide", Aust Health Rev, 2000.

[15] Claudia Pagliari, "What Is ehealth (4): A Scoping Exercise to Map the Field", Medical Internet Research, 2005.

[16] Ashkan Yousefpour, "All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete Survey", Elsevier B.V, 2018.

[17] Z. Shelby, "The Constrained Application Protocol (CoAP): RFC 7252", IETF, 2014.

[18] World health organization regional office for Europe "http://www.euro.who.int/en/health-topics/Health-systems/e-health ", page consulted on 22/03/2019.

[19] Camelot, "Differentiating Between Access Control Terms", 2001.

[20] cisco, "Context-Based Access Control (CBAC): Introduction", cisco, 2008.

[21] VINCENT C. HU AL, "GUIDE TO ATTRIBUTE BASED ACCESS CONTROL (ABAC) DEFINITION AND CONSIDERATIONS", NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, 2014.

[22] DAN BONEH, MATTHEW FRANKLIN,"IDENTITY BASED ENCRYPTION FROM THE WEIL PAIRING", PROCEEDINGS OF CRYPTO, 2001.

[23] A. SAHAI AND B. WATERS, "FUZZY IDENTITY-BASED ENCRYPTION", EUROCRYPT'05 PROCEEDINGS OF THE 24TH ANNUAL INTERNATIONAL CONFERENCE ON THEORY AND APPLICATIONS OF CRYPTOGRAPHIC TECHNIQUES, 2005, P: 457-473.

[24] HONGYING ZHENG AND AL,"MODIFIED CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION SCHEME WITH EFFICIENT REVOCATION FOR PHR SYSTEM" , MATHEMATICAL PROBLEMS IN ENGINEERING, 2017.

[25] ALLAL TIBERKAK ET AL. "A NOVEL APPROACH FOR GENERIC HOME EMERGENCY MANAGEMENT AND REMOTE MONITORING", JOHN WILEY & SONS, 2017.

[26] YANG, YANG; LIU, XIMENG; AND DENG, ROBERT H,"LIGHTWEIGHT BREAK-GLASS ACCESS CONTROL SYSTEM FOR HEALTHCARE INTERNET-OF-THINGS", IEEE , 2018, P: 3610 - 3617.

[27] USAMA SALAMA, LINA YAO AND HYE-YOUNG PAIK. "AN INTERNET OF THINGS BASED MULTI-LEVEL PRIVACY-PRESERVING ACCESS CONTROL FOR SMART LIVING", INFORMATICS, 2018.

[28] "JAVA INTRODUCTION", IN: HTTPS://WWW.W3SCHOOLS.COM/JAVA/JAVA_INTRO.ASP, PAGE CONSULTED ON 02/04/2019.

[29] G.C. MURPHY ET AL. "HOW ARE JAVA SOFTWARE DEVELOPERS USING THE ELIPSE IDE? ", IEEE SOFTWARE, 2006.

[30] "ANDROID STUDIO INTRODUCTION", IN: HTTPS://DEVELOPER.ANDROID.COM/STUDIO/INTRO/ , PAGE CONSULTED ON 02/04/2019.

[31] ADNAN AQEEL, "INTRODUCTION TO RASPBERRY PI 3", https://www .theengineeringprojects.com/2018/07/introduction-to-raspberry-pi-3-b-plus.html, PAGE CONSULTED ON 02/04/2019.

[32] SQL LITE OFFICIAL WEBSITE, IN: "HTTPS://WWW.SQLITE.ORG/ABOUT.HTML", PAGE CONSULTED ON 02/04/2019.

[33] GUY LEDUC: "VERIFICATION OF TWO VERSIONS OF THE CHALLENGE HANDSHAKE AUTHENTICATION PROTOCOL (CHAP)", ANNALS OF TELECOMMUNICATIONS, 2000.

[34] JUNWEI WANG, "JAVA REALIZATION FOR CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION", HTTPS://GITHUB.COM/JUNWEI-WANG/CPABE/, PAGE CONSULTED ON 10/05/2019.

[35] ANDREW BANKS ET AL. "MQTT VERSION 5.0 OASIS STANDARD", OASIS, 2019

[36] M. FAROOQ, M. WASEEM, A. KHAIRI, S. MAZHAR, "A CRITICAL ANALYSIS ON THE SECURITY CONCERNS OF INTERNET OF THINGS (IOT)", PERCEPTION, 2015.

[37] SASTRY AS, SULTHANA S, VAGDEVI S "SECURITY THREATS IN WIRELESS SENSOR NETWORKS IN EACH LAYER", ADVANCED NETWORKING AND APPLICATIONS, 2013

[38] KEJUN CHEN ET AL. "INTERNET-OF-THINGS SECURITY AND VULNERABILITIES: TAXONOMY, CHALLENGES, AND PRACTICE", JOURNAL OF HARDWARE AND SYSTEMS SECURITY, 2017

[39] SVEN BUGIEL ET AL. "TOWARDS TAMING PRIVILEGE-ESCALATION ATTACKS ON ANDROID", NDSS SYMPOSIUM, 2012

[40] NODE-RED, IN: HTTPS://NODERED.ORG/, PAGE CONSULTED ON 02/04/2019.

[41] EUROPEAN RESEARCH CLUSTER ON THE INTERNET OF THINGS, IN: http://www.internet-of-things-research.eu/ , PAGE CONSULTED ON 22/03/2019.