Healing on the cloud: Secure cloud architecture for medical wireless sensor networks

Ahmed Lounis, Abdelkrim Hadjidj, Abdelmadjid Bouabdallah, Yacine Challal

Healing on the Cloud : Secure Cloud Architecture for Medical Wireless sensor networks

Ahmed Lounis, Abdelkrim Hadjidj, Abdelmadjid Bouabdallah and Yacine Challal
{lounisah,ahadjidj,bouabdal,ychallal}@utc.fr
Université de Technologie de Compiègne
HEUDIASYC UMR CNRS 7253
BP 20529, Compiègne Cedex
France

***Abstract*—There has been a host of research works on wireless sensor networks (WSN) for medical applications. However, the major shortcoming of these efforts is a lack of consideration of data management. Indeed, the huge amount of high sensitive data generated and collected by medical sensor networks introduces several challenges that existing architectures cannot solve. These challenges include scalability, availability and security. Furthermore, WSNs for medical applications provide useful and real information about patients' health state. This information should be available for healthcare providers to facilitate response and to improve the rescue process of a patient during emergency. Hence, emergency management is another challenge for medical wireless sensor networks. In this paper, we propose an innovative architecture for collecting and accessing large amount of data generated by medical sensor networks. Our architecture overcomes all the aforementioned challenges and makes easy information sharing between healthcare professionals in normal and emergency situations. Furthermore, we propose an effective and flexible security mechanism that guarantees confidentiality, integrity as well as fine grained access control to outsourced medical data. This mechanism relies on Ciphertext Policy Attribute-based Encryption (CP-ABE) to achieve high flexibility and performance. Finally, we carry out extensive simulations that allow showing that our scheme provides an efficient, fine-grained and scalable access control in normal and emergency situations.**

Keywords: wireless sensor networks, healthcare, cloud computing, attribute based encryption, emergency access control.

I. Introduction

Recent advances in medical sensors, wireless technologies and Micro-Electro-Mechanical systems have enabled the development of senor nodes capable of sensing, processing and communicating several physiological signs. These lightweight miniaturized nodes collaborate to form a wireless sensor network (WSN) that simplify the supervision of patients' health. The major breakthrough of this technology is providing continuous remote patient supervision both in and out of hospital conditions. Consequently, it reduces health cost and improves the quality of life of patients as well as the treatment efficiency.

There has been a host of research works on medical WSN for patient supervision [1]. Proposed solutions have adopted a common architecture with three main components as described in figure 1: Body Area Networks (BAN), gateways, and remote monitoring system. The BAN is a set of sensor nodes carried by the patient to collect different health information. It sends collected data via wireless communication channel to the gateway which serves as a relay node to the monitoring system through a backbone network (ADSL, WiFi, or satellite). The remote monitoring system, usually a server hosted by the healthcare provider, is the heart of the architecture at which the collected data is stored, processed and accessed.

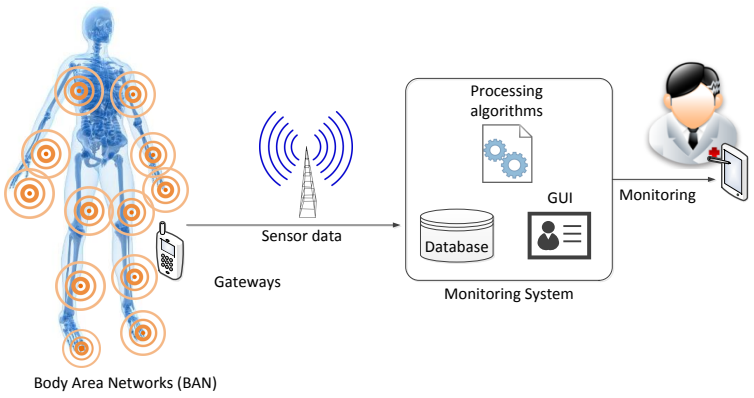

Fig. 1. Remote monitoring system architecture

Scalability is a challenge that WSNs for medical applications should tackle. Indeed, the sampling of medical sensors is performed at high frequency which increases the amount of collected data. In addition, the frequency of sensor sampling is often increased if the condition of patients being monitored gets worse. The important size and heterogeneity of data leads to a need for an increasing storage and processing capacities. Besides scalability issues, medical data could be life saving and must be accessible at any time and from everywhere. Existing solutions rely on a centralized paradigm to store and process sensed data, they cannot tackle the aforementioned challenges. We definitely need new innovative solutions to meet the great challenges of handling the exponential growth in data generated by sensors.

Considering social, ethical and legal aspects of medical systems, data collected by sensor networks is highly sensitive and should be managed properly to guarantee patients' privacy.

Therefore, it is essential to ensure security of collected data during transmission as well as during storage. Access to patient information must be strictly limited to authorized users in order to guarantee the confidentiality. Since data is vital for medical diagnosis, data integrity should be verified to prevent wrong treatments because of malicious or erroneous modifications. Access to medical data is often governed by complex policies that distinguish between each part of the data and each user privileges. Therefore, providing fine-grained access control that supports dynamic and complex organizational policies is a very hard challenge. Practical issues, such as security management, overhead and scalability of the access control with the number of users, also need to be considered. While a lot of research works have been carried out in medical wireless sensor networks, only few studies have been achieved regarding security and existing solutions are far from mature [2].

Emergency management is another challenge in medical applications. WSNs for medical applications is a means to detect and provide useful and real time information about patients' health state to the doctors and emergency staff. Moreover, WSNs facilitate response in case of emergency which can save life of patients. In emergency intervention, medical information of victims is required by emergency staff who may not have enough privileges to access this information. Traditional solutions suggest disabling security system in emergency situations in order to allow emergency staff to access full victim's medical information for controlling emergency. Given the sensitivity of WSNs medical information, an access control solution that supports emergency access to some information without disabling security is then required.

In this paper, we address the challenge of data management in wireless sensor networks for patients supervision. We propose a secure and scalable architecture for collecting and accessing large amount of data generated by medical sensor networks. We leverage cloud computing technology to dynamically scale storage resources via on demand provisioning. Furthermore, we propose an innovative security scheme that eliminates potential security threats of medical data outsourcing and guarantees confidentiality, integrity as well as fine grained access control, while guaranteeing a secure emergency access. Our contributions in this work are many folds: first, we propose a new cloud based architecture for medical wireless sensor networks. Second, we show how we guarantee the confidentiality and the integrity of outsourced medical data without involving patients or doctors interventions. Third, we propose an efficient access control which allows implementing complex and dynamic security policies compliant with medical administrative organization while reducing the management and processing overhead. Fourth, we provide emergency management with two options: A) In proactive manner, our solution relies on sensor networks to detect emergency. Thereafter, our system determines responders and give them temporal access. B) Emergency reporting, where our system enables individual (the victim himself, emergency staff,...) to report emergency situations that WSNs cannot detect. Finally, we carried out extensive simulations that allowed showing that our scheme provides an efficient, fine-grained and scalable access control in normal and emergency situations.

The rest of the paper is organized as follows. In section II we review some related works. In section III we present our proposed architecture. In section IV we review attribute based encryption basics necessary to the understanding of our proposed access control scheme. In section V we describe the security services that are ensured by our architecture. In section VI, we present our emergency management solution. In section VII, we analyze the security of our solution. In section VIII, we provide and simulation results and performance evaluation of our scheme compared to representative schemes from literature. In section IX we conclude the paper.

II. Related works

Scalability via on-demand resource provisioning and virtually infinite data storage capacity makes the cloud computing [3] compelling for managing data generated by WSNs. Cloud computing eases storage, processing and sharing of sensor data and provides anywhere/anytime access to supervision applications. Research works on coupling WSN and the cloud are still in their early infancy. A recent paper [4] tried to identify the opportunities and challenges of connecting wireless sensor networks to the Cloud. Also, few papers introduced cloud computing to different WSN applications such as industrial supervision [5], patient data collection [6], energy monitoring [7] and environmental monitoring [8]. However, all these papers described preliminary works and ignored the challenges induced by combining WSN and cloud computing.

Authors in [9] proposed a framework based on a publish/subscribe model which facilitates WSN-Cloud connection. In another paper [10], they used this framework to monitor human activities and to share information among doctors, caregivers, and pharmacies. However, authors did not discuss the security requirements for such a framework. In ESPAC [11], data collected from patients are sent to the hospital server before being stored on the Cloud. Despite taking advantage of the cloud to offer unlimited data storage, the scalability of this scheme is limited. Indeed, the hospital server is a bottleneck (single point of failure) that may crash in the case of flash crowd. In addition, no data storage and data access are possible on the cloud if the hospital server is out of order or inaccessible.

The storage of sensitive data over untrusted servers requires cryptography techniques in order to keep data confidential and preserve patients' privacy. Various solutions, based on symmetric or public cryptography, have been proposed to provide cryptographic access control that allows storage and sharing of data on untrusted servers [12][13][14][15][16]. However, These techniques do not support fine grained access control required by medical applications. Indeed, they are not scalable with the number of users and introduce high complexity in key distribution and management.

Recent works leveraged new cryptography techniques, such as Role Based Access Control (RBAC) and Attribute Based Encryption (ABE), to provide fine-grained access control required by personal medical systems. Ibraimi et al. [17] applied Ciphertext Policy ABE (CP-ABE) to enable patients to securely store and share their health record on external third party servers. Barua et al. [11] used bilinear pairing and ABE to guarantee data confidentiality and integrity as well as user privacy and authentication in cloud-based medical systems. In [18], authors proposed a novel practical framework for fine-grained data access control to medical data in Cloud. To avoid high key management complexity and overhead, they organized the system into multiple security domains where each domain manages a subset of users. Unfortunately, all these works adopted a patient-centric approach where each patient generates his own security keys and distributes them to authorized users. We argue that the patient-centric approach is not applicable to manage data collected by WSN for patient supervision. Indeed, the access policies that govern such systems are often complex to be defined by the patient. In addition, the healthcare organization is the legal owner and the responsible for patient's health data during his hospitalization within the hospital or at home settings. Consequently, security policies must be fixed by the healthcare organization rather than the patient. Finally, the patient being monitored by the sensor network can have serious health problems which make him unable to define a security policy in this reduced-mobility state.

III. Our architecture

In this section, we describe our architecture which enables a healthcare institution, such as a hospital or a clinic, to manage data collected by WSN for patient supervision. The proposed architecture is scalable and able to store the large amount of data generated by sensors. Since these data are highly sensitive, we propose a new security mechanism to guarantee data confidentiality, data integrity and fine grained access control. Unlike existing patient-centric systems, security configuration and key management in our solution are totally transparent to users (patients and doctors) and do not require their intervention.

In order to achieve the aforementioned objectives, we propose the architecture described in figure 2. This architecture considers two categories of users, healthcare professionals and patients, and is composed of the following components: (1) the WSN which collects health information from patients, (2) the monitoring applications which allow healthcare professionals to access the stored data, (3) the Healthcare Authority (HA) which specifies and enforces the security policies of the healthcare institution and (4) the cloud servers which ensure data storage. By storing data on the cloud, our architecture offers virtually infinite storage capacity and high scalability. Indeed, the architecture increases its storage capacity, through on-demand provisioning feature of the cloud, whenever it is necessary. In addition, it offers enormous convenience to the

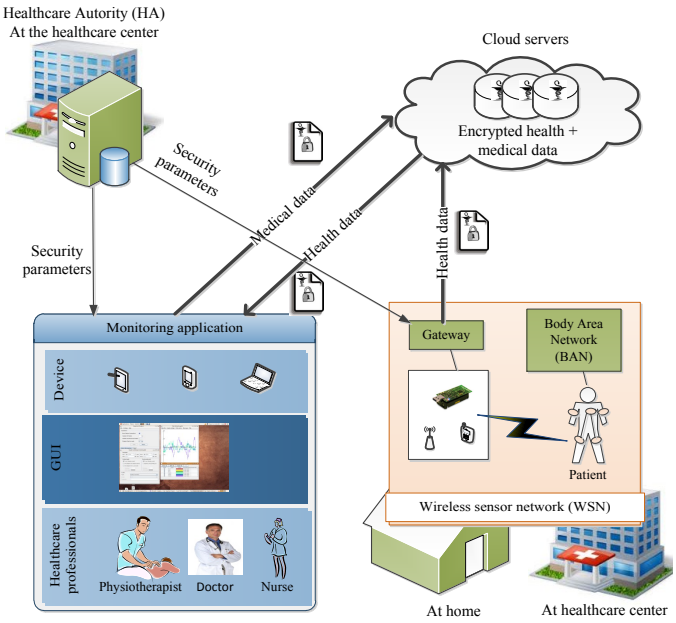

Fig. 2. The proposed architecture

healthcare institution since it does not have to care about the complexity of servers' management.

To achieve fine-grained access control, we can use attribute based encryption (ABE) to encrypt data before storing them on the cloud. However, integrating ABE into medical systems is a real challenge. In ABE, data are encrypted with an access structure which is the logical expression of the access policy (eg: the data can be accessed by physician in cardiology division or by nurses). The cyphertext (encrypted data) can be decrypted by any user if his secret key has attributes that satisfy the access policy. The power of ABE is that we do not need to rely on the storage server for avoiding unauthorized data access since the access policy is embedded in the cyphertext itself. However, this characteristic becomes an inconvenient when the access policy changes. Indeed, to apply a new access policy to a file, we must download it, re-encrypt it with a new access structure and upload it again to the cloud. The second challenge faced with the integration of ABE is keys and access structures management. Indeed, the questions of who should generate the access structure that govern the security policy and who should generate and distribute keys necessary to access to the data are a real challenge in medical systems. To answer these questions, existing ABE-based systems adopted a patient centric approach that we showed unsuitable for our application.

To tackle the first challenge of ABE integration, we propose to use both symmetric cryptography and ABE to encrypt data. More specifically, we propose to encrypt each file with a randomly generated symmetric key (RSK) and encrypt the RSK with ABE. Both the encrypted file and the encrypted RSK are sent to the cloud for storage to allow fine grained data sharing with authorized users. Indeed, if a user has a secret key that satisfies the ABE access policy, he will be able to decrypt

the RSK and hence to decrypt the file. Furthermore, if the file access policy changes, we should download and re-encrypt the RSK rather than the whole file. This lead to a significant gain in data communication and encryption operations. Finally, our solution has less encryption overhead compared to the naif utilization of ABE to encrypt the whole file. In fact, ABE consumes much more processing power than symmetric cryptography when we use complex access policies [19] like the ones used in medical systems.

To tackle the second challenge, which is mastering the complexity of security management, we introduce an entity that we call ***Healthcare Authority (HA)***. The HA specifies and enforces the security policies of the healthcare institution. It is used by the administrators of the healthcare institutions to define rules as "who can access to what". Based on these rules, the HA generates and sends to each user his ABE security parameters which are a pair of ***access structure*** and ***secret key***. The secret key is generated from the user attributes set which represents the user privileges. This information is required to decrypt data that the user is allowed to access. The access structure represents the access policy that protects the user data. When a user encrypts the random symmetric key (RSK) that protects his data using this structure, he can be sure that only authorized users (who have the correct attributes) can decrypt and access his data. Introducing the HA releases users from creating and distributing access structures and secret keys. Consequently, it improves the system usability since a patient has no action to do to secure his data. Also, the healthcare professionals transparently access to data falling under their scope. All the details of security operations are given in section V.

In our architecture, each patient has a personal WSN composed of a set of lightweight/small sensor nodes and a gateway. A WSN enables unobtrusive and continuous health supervision of the patient at the hospital and at home settings. Sensor nodes are carried by the patient to collect different ***health data*** such as heart beats, motion and physiological signals. Each sensor node sends the collected information via a wireless communication channel to the gateway. The gateway aggregates the different health data into a file and encrypts it using the RSK. Thereafter, it sends to the cloud the encrypted file along with the RSK which is encrypted using the access structure obtained from the HA.

The monitoring application allows healthcare professionals to supervise their patients and enables them to access to a patient's data anytime and from everywhere. The monitoring application downloads the required data from the cloud and decrypts it using its secret key. In addition, it allows the healthcare professionals to add ***medical data***, such as reports, diagnostics and prescriptions, to the patient's information. The medical data is also encrypted and stored on the cloud along with the patient's health data. Similarly to the gateway, the monitoring application encrypts the medical data using a RSK and the access structure obtained from the HA.

IV. Background: Attribute-based encryption

Attribute-based encryption (ABE) is a promising cryptographic method proposed by Sahai and Waters in 2005 [20]. The ABE technique extends the identity-based encryption to enable expressive access policies and fine-grained access to encrypted data. With ABE, the access control decision is based on a set of attributes and the concept of access structure described as follows :

- **Universal attributes set (U):** is the set of all attributes that describe data properties, user properties and environment properties.
- **Access structure:** is an access policy that designates who can access to what. It is built from an access tree (T) which can be seen as a logical expression combining several attributes through AND, OR or other operators (figure 3). Each non-leaf node of the tree represents a threshold gate, described by its children and the threshold gate value (AND, OR or other operators). Each leaf node of the tree is described by an attribute from U and a value.

In figure 3, we give an example of an access tree which is derived from the following logical expression: ((speciality=physician AND (division=cardiology OR division=pulmonary) OR (division=gerontology AND (speciality=nurse OR speciality=physician))). This expression means that data can be accessed by all physicians working in cardiology, pulmonary or gerontology divisions, as well as all nurses working in gerontology division have access.

Key-Policy Attribute-Based Encryption (KP-ABE) [21] and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [19] are the two main variants of ABE. KP-ABE assigns to each file a set of attributes to be encrypted, and assigns to each user an access structure, that represents his access scope, for data decryption. On the contrary, CP-ABE assigns to each file an access structure to be encrypted and uses a set of attributes to generate the user's key for data decryption. In medical systems, healthcare professionals are assigned particular roles (eg. general practitioner, nurse), and depending on their role, they get permissions to access to particular data or not. Implementing these policies is easier and more efficient using CP-ABE than using KP-ABE. Indeed, we can describe the role of each healthcare professional by assigning him a combination of attributes. At the same time, we encrypt each file by an access structure that expresses the access policy. In what follows, we present the basics of CP-ABE necessary for understanding of our architecture. More extensive description of CP-ABE is available in [19].

A CP-ABE scheme consists of four fundamental algorithms: setup, encrypt, key generation, and decrypt.

Setup: defines the universal attributes set (U) and computes the public key (PK) and the master key (MK). The public key (PK) is used by encryption and decryption algorithms. The master key (MK) is needed to generate secret keys by the Key generation algorithm.

Encryption (PK, M, A): it takes as input the public key PK, a message M, and an access structure A built over the universal

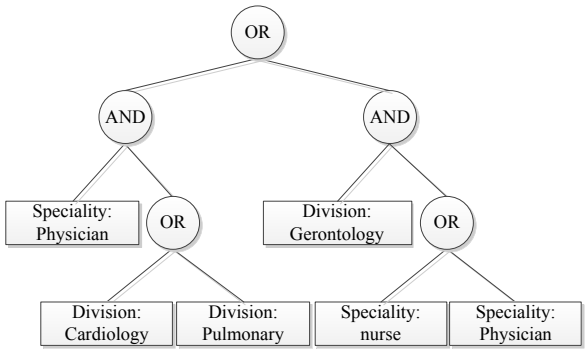

Fig. 3. An access tree T

attributes set (U). This algorithms encrypts the message M according to the access policy that is defined by the access structure A, and gives as output the ciphertext CT. Only users having a set of attributes corresponding to the access structure A can decrypt the ciphertext (CT).

Key generation (MK, S): this algorithm takes as input a master key MK and the user set of attributes S and generates the user's secret key SK.

Decryption (PK, CT, SK): it takes as input the public key PK, the ciphertext CT and a secret key SK. It returns a message M that is plaintext of CT if the set of attributes corresponding to SK satisfies the access structure A of CT.

V. Security services implementation

In this section, we give the security model and define the services that are ensured by our architecture. Then, we details the implementation of these security services.

A. Security Model

Our system is composed of the following parts: users (patients and healthcare professionals), cloud servers and the healthcare authority (HA) server. We assume that communication channels between users, the HA and cloud servers are secured by a security protocol such as SSL. Even if SSL guarantees data confidentiality and integrity during transfer, we should encrypt data at the user level because we consider that cloud servers are untrusted. Indeed, data are stored on clouds operated by companies that may disclose personal information to third parties. For legal and ethical concerns, the cloud provider should neither be able to access to patients' data nor perform data mining or patients profiling. Furthermore, we consider that cloud servers might collude with some malicious or revoked users for illegal data access. Similarly, users might collude together to illegitimately access to file contents. The Healthcare Authority (HA) assures keys and access policies management. We consider that the HA is trusted and secured. Finally, we assume that each party has a public/private key pair and the public key can be easily obtained by other parties through a Public Key Infrastructure (PKI).

Notation	Description
PK, MK	System public key and master key
$Priv_P$, Pub_P	Patient private key and public key
$Priv_{HP}$, Pub_{HP}	Healthcare professional private key and public key
$Priv_{cloud}$, Pub_{cloud}	Cloud private key and public key
SK_P	Patient secret key
SK_U	User secret key
RSK	Symmetric random secret key
AR_P	Patient access structure for read access
AW_P	Patient access structure for write access
AR_{HP}	Healthcare professional access structure for read access
AW_{HP}	Healthcare professional access structure for write access
ID	Unique file identifier which is a structure allowing to find the file we need.
PASS	Password

Fig. 4. Notation used in our solution

B. Security services

Our architecture guarantees the following security services.

Fine-grained Access control: our solution ensures healthcare information confidentiality and scalable fine-grained access control to data stored on the cloud. Furthermore, it ensures access control in multi-writers access mode on medical data.

Integrity and authenticity: our solution ensures message integrity during transfer between two parties. Furthermore, it ensures data integrity during storage on cloud servers. Also, each party authenticates the origin of each message received from other parties.

Availability and scalability: our solution ensures availability of service for legitimate users when they need it, and it is resilient when a large group of legitimate users' access at the same time (flash crowd).

Collusion resistance: our solution enforces the access control system and guarantees that users (patients/healthcare staff) cannot collude together to get illegitimate access to medical data. Our architecture resists against the collusion attacks to avoid any unauthorized access to medical data.

C. Security Implementation

1) System initialization: at the initialization of our architecture, the HA creates the universal attributes set and calls the ABE setup algorithm to generate the master key (MK) and the public key (PK). The MK must remain secret while the PK must be known to all users since they need it to encrypt and decrypt data. To share the PK, the HA signs it with its private key and sends it, along with the signature, to cloud servers. Once the PK on the cloud, users can download it and check its authenticity thanks to the signature.

2) Adding a new user: When a new patient is admitted to the hospital, the Healthcare Authority gives him a secret key and an access structure. The access structure allows him to encrypt his data before uploading it on the cloud and ensures that only authorized users can access to it. The secret key allows him to access to medical data on which he has right.

The following steps are performed each time a new patient P joins the system :

1) The PKI generates a couple of private/public keys ($Priv_P$, Pub_P) for the patient P.
2) The HA calls the key generation algorithm of CP-ABE to generate the secret key SK_P. Furthermore, it builds the access structure AR_P that the patient P will use to encrypt his health data.
3) The HA asks the cloud to add the patient P to the users lists.
4) Upon receiving the patient addition request, the cloud adds the patient P and his public key Pub_P to the users list (LU).
5) When the patient's gateway establishes a connection to the HA for the first time, it receives the corresponding secret key SK_P, access structure AR_P and private key $Priv_P$.

The difference between the security parameters of a patient and a healthcare professional comes from the fact that a patient needs to encrypt *health data* which can be only read while a healthcare professional needs to encrypt *medical data* which can be both read and modified. The read access policy and the write access policy which govern a medical data may be different. For example, a nurse can only read a report while a doctor can read and modify it to add comments. Consequently, the healthcare professional should obtain two access structures for read and for write policies. The following steps are performed each time a new healthcare professional HP joins the system :

1) The PKI generates a couple of private/public keys ($Priv_{HP}$, Pub_{HP}) for the HP.
2) The HA calls the key generation algorithm of CP-ABE to generate the secret key SK_{HP}. Furthermore, it builds an access structure AR_{HP} that the HP will use to encrypt the medical data. Also, it builds another access structure AW_{HP} for protecting the write mode. We will explain how the AW_{HP} is used in the medical data management subsection.
3) The HA asks the cloud to add the HP to the users list.
4) Upon receiving the HP addition request, the cloud adds the HP and his public key Pub_{HP} to the users list (LU).
5) When the HP establishes a connection to the HA for the first time, its application receives the corresponding secret key SK_{HP}, private key $Priv_{HP}$ and access structures AR_{HP}, AW_{HP}.

3) health data management: Health data files are information collected by the WSN and can be accessed only in reading mode. The gateway continuously receives information collected by sensor nodes and executes the following algorithm when this data is ready to be uploaded to the cloud:

1) Assign an unique identifier ID to the health data file F. it is a structure allowing to find the file we need.
2) Generates a random secret key RSK for a symmetric cryptography algorithm
3) Computes H the hash value of the file F
4) Uses RSK to encrypt the concatenation of the file F and the hash value H
5) Encrypts RSK with CP-ABE encryption algorithm according to the access structure AR_P
6) Sends to the cloud the following data :

ID	$\{RSK\}_{AR_P}$	$\{(Data + H)\}_{RSK}$

Once stored on the cloud, the health data can be used by healthcare professionals to remotely supervise the patient or by the patient himself. When a user U wants to access a health data file, he starts by downloading this file from the cloud server. After, he decrypts the RSK field of the file using ABE and his secret key SK_U. If he has the right to access this file (his secret key corresponds to the access structure of the patient P), he gets the correct RSK and hence decrypts the file. After the decryption, the user checks the integrity of the content thanks to the hash value. If he detects that the data file was altered he signals it to the Healthcare Authority. Figure 5 shows the different steps performed from adding a new patient until its supervision.

4) Medical data management: The medical data (such as reports, diagnostics and prescriptions) are created by healthcare professionals and can be modified by other authorized users. The read access to medical data is similar to health data management. However, to control medical files updates, we assign to each file a password given to only authorized entities (users or cloud) to allow them to modify the file. To allow a user to upload a new version of a file F, the cloud asks him for the file password. If the user provides the correct password, the new file version is accepted. When a healthcare professional HP creates a new medical file F, he performs the following actions:

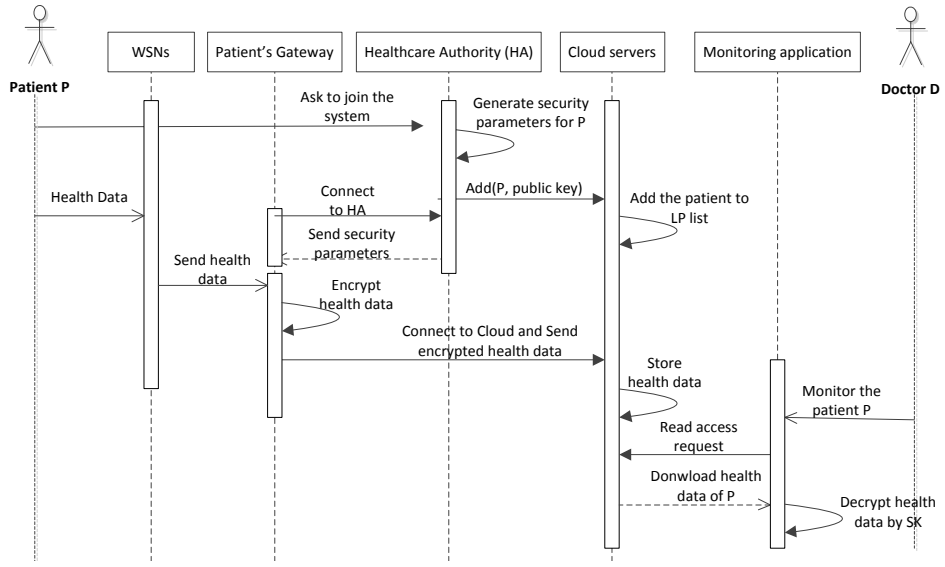

Fig. 5. Example of patient supervision

1) Assigns a unique identifier ID to the medical data file F
2) Generates a random secret key RSK for a symmetric cryptography algorithm
3) Generates a random password $PASS$ for protecting controlling the write access
4) Computes H the hash value of the file F
5) Uses RSK to encrypt the concatenation of the file F and the hash value H
6) Encrypts RSK with CP-ABE encryption algorithm using the read access structure AR_{HP}
7) Encrypts $PASS$ with CP-ABE encryption algorithm using the write access structure AW_{HP}
8) Encrypts $PASS$ with the public key of the cloud
9) Sends to the cloud the following data :

ID	$\{RSK\}_{AR_{HP}}$	$\{PASS\}_{AW_{HP}}$	$\{PASS\}_{Pub_{Cloud}}$
$\{(Data + H)\}_{RSK}$			

To read the content of a medical file, a user U performs the same actions described in the last section (access to health file). However, to modify a medical file he performs the following actions:

1) Downloads the medical file
2) Updates the file content and computes the new hash value of the file;
3) Encrypt the medical content along with the new hash value using RSK;
4) Decrypt the password with ABE and Sk_U
5) Sends to the cloud an update request containing the new file along with computed password
6) Upon receiving the update request, the cloud decrypts the password of the original file using his private key $Priv_{cloud}$. The new version of the file is accepted if and only if the password computed by the cloud is equal to the password in the update request.

5) Data health deletion: this operation can be performed only by the file owner. To delete a file, the owner signs and sends a delete request to the cloud. Upon receiving this request, the cloud checks if the sender is the real owner of the file based on the signature and proceeds to the deletion.

6) Revocation: there are two types of revocation. The first one consists of limiting access to data through modifying the access policy. To change a data access policy, we should create a new data access structure and re-encrypt the desired data. The second one consists of revocation of attributes that are associated to a user to limit his access scope. To do so, the system should determine the set of attributes which must be updated. Then, it regenerates new system master key and public key (MK and PK) of ABE. Finally, according to the updated attributes set, some user's secret keys must be also updated and some files must be re-encrypted. However, these operations induce high computational overhead for key management and distribution. So then, the user attribute revocation is not scalable. In [22], to allow scalable revocation S. Yu et al. rely on dynamic scalability of the cloud by delegating most of

laborious revocation tasks such as user secret key update and file re-encryption to the cloud without disclosing file contents or user access privilege information. This solution implements the proxy re-encryption technique on the cloud. The goal of re-encryption proxy is securely to enable the re-encryption of ciphertexts from one secret key to another, without relying on trusted parties. Our scheme supports the implementation of this solution. However, this solution does not reduce computational overhead, it only enables the cloud deals with some revocation tasks. In our solution, to allow an efficient revocation and solve this challenge, we add an expiration time attribute to each user's key. This expiration time indicates until when the key is considered valid. Indeed, to avoid revocation tasks user access privileges are temporary allocated. After expiration of user key, he needs a new key to allow him continuing to access patient's medical data.

VI. Emergency management

In previous section, we have described our architecture which provides a solution to healthcare institution for patient supervision. This proposed architecture is scalable and able to store the large amount of data, and it integrates security mechanisms to ensure high level security of sensitive medical data. In this architecture, the security mechanisms are totally transparent which improves its usability to users. However, there is a major challenge that the proposed architecture does not consider: emergency management. Indeed, the healthcare providers must support emergency management policies and procedures that proactively and continually identify emergency situations and should be able to respond effectively and appropriately to an emergency situation.

In [23] Ming Li et al. proposed a patient-centric framework. Using ABE, they implemented secure, scalable and fine-grained access control to Personal Healthcare Records (PHRs) stored in the cloud. Ming Li et al. considered emergency access by providing break-glass access for extending a person's access rights in emergency cases. In [23] break-glass access is managed by an authority called emergency department (ED). Each patient delegates his emergency key to ED which will give it to medical staff in emergency situation after identifying and verifying enquirer. This solution is simple and allows exceptional access to victim's PHRs when emergency happens. However, the separation between break-glass access and regular access makes this kind of solution suffering from duplication issues of PHRs storage, where PHRs are stored in two forms: PHRs encrypted with ABE system and PHRs encrypted with emergency key. Also, despite the introduction of ED which is in charge of key management, after each emergency situation, the victim should be online to revoke emergency access. In [24] A D.Brucker et al. have proposed a fine-grained break-glass access which is constructed by integrating break glass access concept into a system for end-to-end secure data sharing based on ABE. In [25], K. Venkatasubramanian et al. proposed a criticality aware access control for emergency (criticality) management in smart-infrastructure. This solution can be applied in several emergency management applications, such as the case where a patient needs urgent medical assistance. In emergency situation, this solution becomes more proactive, where the system evaluates emergency situation to identify the response actions that need to be taken and enables them, and allow chosen responders (subject) to access the system with set of privileges for emergency management. However, this solution does not provide encryption service of medical data available in emergency. As medical data is highly sensitive and the risk of unauthorized disclosure of this data when it is unecrypted and stored on the cloud is very high since the cloud is untrusted and there is no totally transparency and control on data when it outsourced on the cloud.

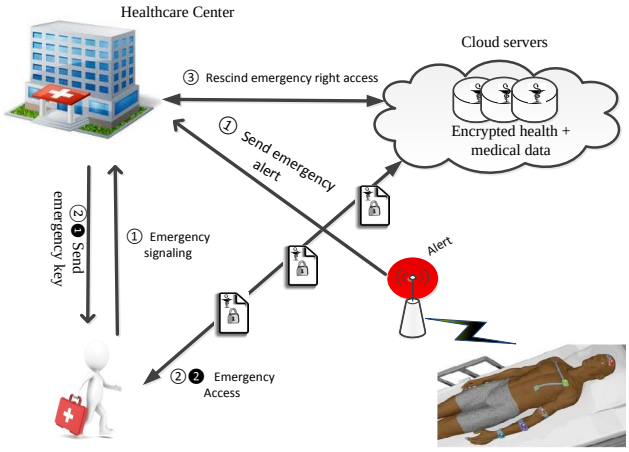

Fig. 6. Example of Emergency intervention

In what follows, we present our emergency management solution. Where we show how an emergency case can be supported by our cloud-based medical WSNs architecture. Then, we present implementation of our emergency access control.

A. Emergency scenario and requirements

Our architecture allows the patient to be remotely supervised by healthcare staff based on collected physiological data which is continuously processed and transmitted to cloud servers. The healthcare staff access to this information through the cloud and can use it to treat the patients. Also, promoting timely intervention of healthcare staff as and when required. Indeed, medical WSN could detect some emergency situations by analyzing collected information. Then, it alerts the healthcare staff for timely intervention. Cardiovascular diseases which include heart attacks, heart failure, stroke, coronary artery disease are an example of emergency intervention. When an emergency situation is not detected by WSN infrastructure, the emergency may be reported by emergency responders who

need access to patient's medical data, or by any person which is not medical staff (patient, patient's family): for example, a patient who is victim of a road accident.Therefore, we distinguish two possible emergency scenarios:

1) **Proactive scenario**. In this scenario, our access control deals with emergency in proactive manner. Namely, the system can detect emergency situation when it happens thanks to analyzing health data collected by WSN, and determines a set of responders (emergency staff, patient's doctor) and access rights which enable them to access victim's medical data needed in emergency. In our architecture, as shown in figure 6, the healthcare authority is alerted by the gateway which informs that an emergency case happened with a patient. Then, the healthcare authority finds responders and gives them access privileges (emergency key) of victim's medical data according to emergency case.
2) **Passive scenario**. In the first scenario, the emergency detection is done thanks to WSN, which alerts Healthcare center to rescue the victim. So then, the healthcare center determines responders and gives them temporary access to victim's medical data. However, not all emergency situations can be detected by WSNs. In this case, first aiders and doctors who deal with the victim until his returning to stable state request for getting temporary access victim's medical data when this is needed. We call this case a passive scenario.

The both described scenarios consist of three phases:

1) **Emergency detection:** this phase is responsible for identification of emergency situation when it is happens.
2) **Response:** after identification of emergency, the system gives access rights to responders. To improve response time of access to victim's data in emergency situation our access control should give the priority to emergency access while ensuring bounded waiting time for other requests.
3) **Mitigation:** in this phase the time allowed for emergency is over, so access rights will be revoked.

Emergency management should:

- Allow responders to access victim's medical data in emergency situation while preserving patient's privacy. Indeed, responders need a temporary access to a part of the victim's medical data to ensure timely intervention.
- Preserve the fine-grained access property of our solution. Hence, allow access to data according to complex policies in emergency situations.
- Preserve the scalability of our access control while considering requests which due to emergency situations.

In what follows, we present our solution which allows managing emergency situation which considers the above requirements.

B. Emergency management: Implementation

To extend our access control solution to support emergency access, the healthcare authority generates different access

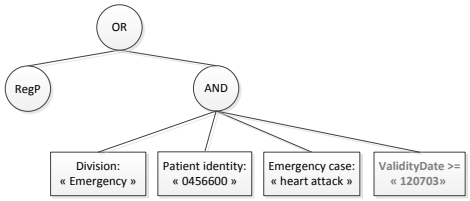

Fig. 7. An example of access structure for break-glass access

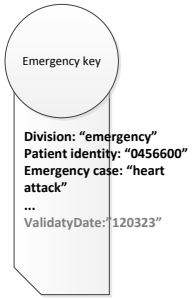

Fig. 8. Emergency key

structures from emergency policies in ABE form. Then, instead of sending only regular access structures to users, as in initial architecture, it sends access structures that are disjunction of emergency and regular access structures, as shown in figure 7. So, authorized users (emergency staff, patient's doctor,...) can decrypt medical data if his secret key satisfies regular access policy or if his emergency key satisfies emergency access policy, and the used key is still valid. We can use attributes which are used in regular policies to construct an emergency access structure(such as division attribute to define a particular policy which is applied for a specific division in hospital, function attribute to define a particular policy which is applied for specific medical function). In addition, we need other attributes for emergency management such as Emergency Case (EC) which allows the identification of required medical data to ensure emergency response. Also, to avoid unauthorized access to medical data of other patients who are not concerned with the current emergency case we need to indicate the patient identity (PI) in emergency policies and emergency key. In order to accelerate emergency response, we suggest that emergency keys are prebuilt and stored (in HA or patient device such as mobile phone) in a secure manner.

Emergency access is temporary and it should be disabled after the end of the time granted to emergency response. To revoke access rights given in emergency situation, we need to revoke emergency keys. However, revocation is a very difficult issue in attribute based encryption schemes and may include high overhead. To handle the revocation problem of

emergency key in our scheme, we provide a temporal access to patient's medical data by using integer values and integer comparisons proposed in Bethencourt et al. [19] scheme. To do so, we introduce a numerical attribute which has a date value to express validity date *VD* of emergency key in the format *VD*=*YYYYMMDD* (Y: year, M: mounth, D: day), and each medical data is encrypted according to access structure which contains numerical comparison of validity date attribute as $VD \geq 20100624$. Consequently, the user can decrypt medical data with his emergency key expiring on VD only if access structure comparison ($VD \geq$ YYYYMMDD) is verified and the rest of the emergency policy matches the user's emergency attributes. When the time allowed for emergency response comes to end, the available patient's medical data in emergency should be re-encrypted with the current new date. Note that to revoke medical data access, we need only to re-encrypt the random secret keys RSKs of concerned files.

VII. Security and performance analysis

A. Security analysis

Our solution guarantees message integrity, authenticity and confidentiality during data transfer through SSL protocol. Furthermore, it ensures a secure and fine grained access control to data files stored on the cloud. Indeed, data files are encrypted by a randomly generated symmetric key, and this key is encrypted by CP-ABE. The CP-ABE scheme has been proved secure in [19]. Especially, The CP-ABE scheme has been proved resistant against collusion attacks and ensuring that encrypted data cannot be accessed by unauthorized users. From this, we deduce that the random symmetric key is confidential and can be accessed only by authorized users. Consequently, the data confidentiality is guaranteed by the standard symmetric encryption security.

Since our scheme enables scalable and fine-grained access control, the HA is able to define and enforce expressive and personalized access structure for each user. These access structures enable us to select with fine granularity which users can access to the symmetric key of a given file. Since accessing the symmetric key is necessary to access the file, we deduce that these access structures enable us to select with fine granularity which user can access a file contents. Finally, by using separate access structures for the read and write policies, we separate between read and write access to medical data.

Furthermore, our scheme is resilient against man-in-the-middle attacks by considering two concerns: the first is attack during communication between entities of system that requires verifying if public key is correct, and belongs to the person or entity claimed, and has not been tampered with, or replaced by, a malicious third party. The second is how ensure that PK key of CP-ABE system is the original PK which is provided by our healthcare authority. In our scheme, to respond to first issue, each emitter sends his digital certificate issued by our public key infrastructure to receiver. Then, the receiver verifies validity of certificate by using public key of our PKI. For second issue, the CP-ABE Public Key is signed by Healthcare authority, and any entity of system can verifies authenticity of CP-ABE public key before to use it.

B. Performance analysis

1) Encryption operations analysis: CP-ABE enables fine grained access control to data but induces important processing overhead with complex access policies like the ones used in medical systems. The encryption time of CPA-ABE is linear with the number of leaf nodes of the used access structure. However, measuring the decryption time is more difficult since it significantly depends on the used access tree (number of leaf nodes, the type of used operators, the depth of the tree ...) and the set of involved attributes [19]. Here we present preliminary performance evaluation to show the benefit of our solution compared to ABE. We considered several random access structures and attribute sets that we can meet in a real medical system. We used the toolkit developed in [26] for ABE and the AES implementation of OpenSSL for the symmetric encryption.

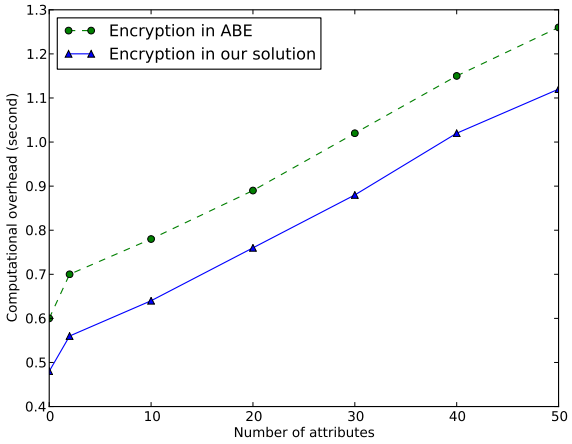

Fig. 9. Encryption evaluation

First we present performance evaluation of encryption and decryption operations that is shown respectively in Figures 9 and 10. For this, we compute time overhead of encryption and decryption while varying the number of leaf nodes of access structure (number of attributes). Figures 9 and 10 respectively show that ABE consumes more time than our solution in both encryption and decryption. These results match our expectations and show that our control access scheme is more efficient in terms of cryptographic operations. Indeed, our solution uses AES to encrypt the data file and uses CP-ABE to only encrypt the AES key (256 bits). Since AES is faster than ABE, we reduce the whole encryption and decryption time. This reduction varies between 11% and 20% for encryption, and between 32% and 41% for decryption in the studied samples. Notice that these performance evaluations do not consider the significant gain that we can achieve in grant and revocation thanks to our access control. Indeed, our evaluation of grant shows that our access control is very

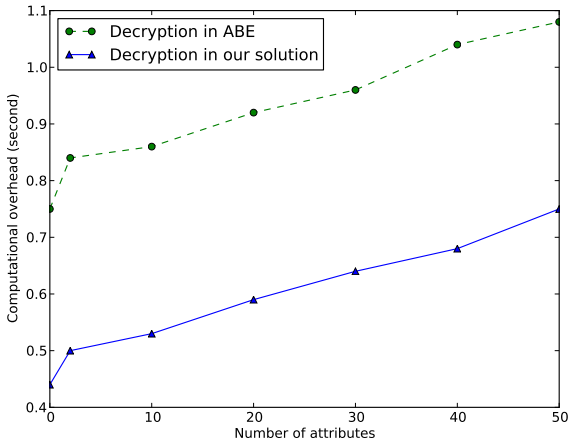

Fig. 10. Decryption evaluation

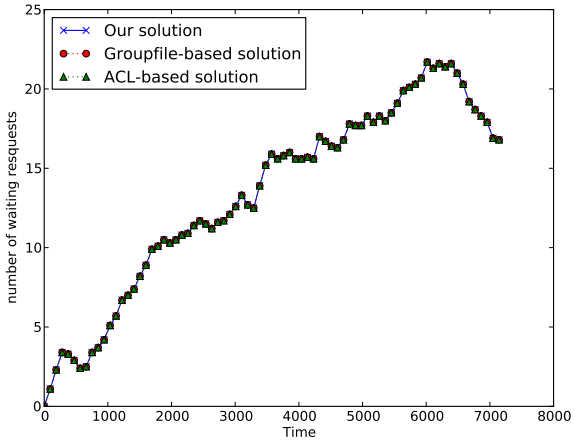

Fig. 12. Performance evaluation with basis operations

efficient when the reduction varies between 60% and 90% in the studied samples, shown in figure 11. We can explain this by the fact that in our solution, we re-encrypt only AES key, so just a data of 256 bits, instead of re-encrypt a whole data file with new access structure in ABE.

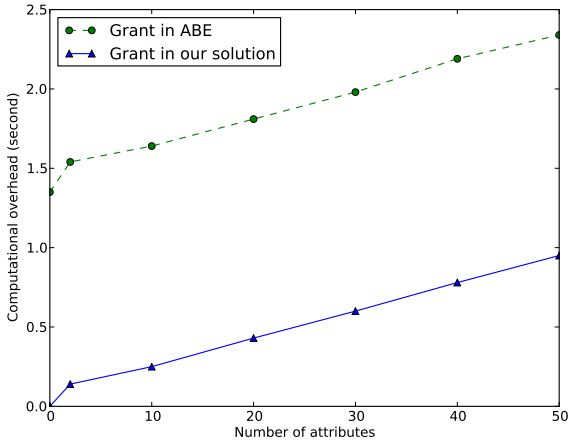

Fig. 11. Grant evaluation

2) Simulation: To evaluate the performance of our solution we simulate several scenarios when multiple parameters are varied to analyze their impact on our solution.

In a first scenario, we do not consider operations induced by emergency situations, and we assume that there is no access policies update during time of evaluation. We consider three operations: read a file from the cloud, write a file on the cloud and create a file on the cloud. We study the mean number of waiting requests during an interval of time. We evaluate three schemes: the first, our security scheme which combines CP-ABE with symmetric AES encryption. The second, filesgroup-based solution where a same symmetric key is used to encrypt and decrypt files of a same group such as in Plutus [12]. The last, access control list (ACL) based solution where each file is associated with a meta data file that contains fils's access control list such as in SiRiUS [13]. In ACL-based solution, each entry in the ACL is the file's encryption key encrypted by using public key of an authorized user. The arrival times of user requests are modeled as Poisson distribution with arrival rate (λ). Also, we use a queue to accommodate different requests which arrive to the cloud. Although, encryption and decryption overhead is not the same for the three solutions the figure 12 shows that we have more or less the same performance with the three solutions. This can be explained by the fact that response time depends more on file transfer delays than encryption or decryption time.

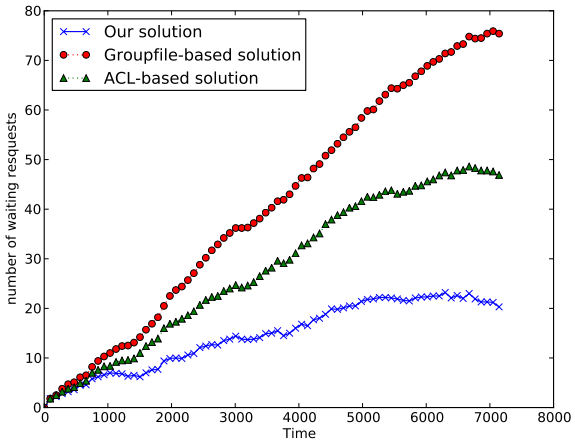

Fig. 13. Performance evaluation with access policy changes

In a second scenario, we introduce multiple changes on access policies that results in right revocations and grants. In this case, we observe that our scheme is depicts higher performance than the other two solutions, as shown in figure 13. Indeed, revocations overhead is high in ACL-based

solution and more in filesgroup-based solution compared to our solution. This overhead is due to re-encryption operations caused by access policies update. In case of files-group based solution, we need to change key of one or several groups that induces re-encryption of all files of group. In case of ACL-based solution, it is necessary to re-encrypt concerned files and update their meta data files which theirs sizes depend on number of authorized users. In our solution, we avoid these operations by using key expiration time where the access rights are temporary assigned to users. Consequently, this shows that unlike to other two solution, with our solution we can achieve simultaneously fine-granied access and scalability.

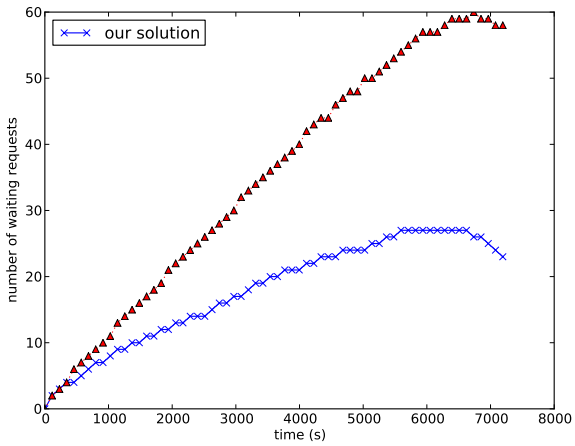

Fig. 14. Performance evaluation with emergency situations

In a third scenario, we consider emergency situations together with previous operations. An emergency situation involves three phases where each phase results in one operation. These operations are respectively: emergency detection, emergency access and emergency revocation. In addition to our solution, we also evaluate break-glass access of Ming Li et al. [23] solution. In Ming Li et al.[23], each data available to emergency access is duplicated and encrypted with emergency key. To revoke emergency access rights, the data is re-encrypted with a new emergency key. The re-encryption of data after each emergency access induces high overhead costs, as shown in figure 14. However, in our solution we avoid this cost thanks to our break-glass access which is presented in section VI-B.

In the fourth scenario, we vary the rate of emergency arrivals and we compute the mean response time, figure 15 shows that increasing the arrival rate of emergency increases the response time. However, this growth in response time is more important in Ming Li et al. [23] solution.

In the fifth scenario, we evaluate the system load of our solution which is hosted on two different kinds of infrastructure: traditional infrastructure with a single server, and the cloud. In different time moments we compute the number of waiting user requests which arrive according to poisson process. In the cloud, initial configuration of resource capacity is similar to

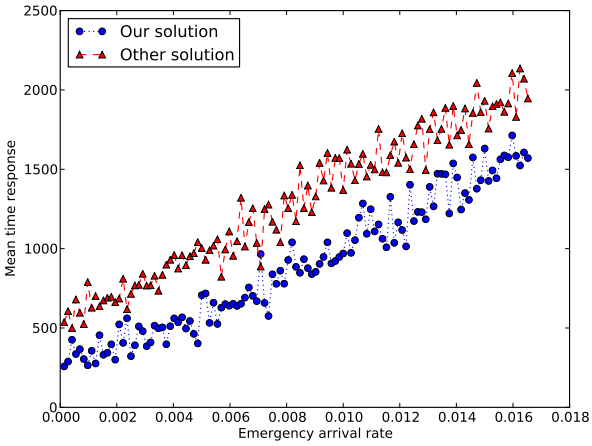

Fig. 15. Average waiting time according arrival rate (λ)

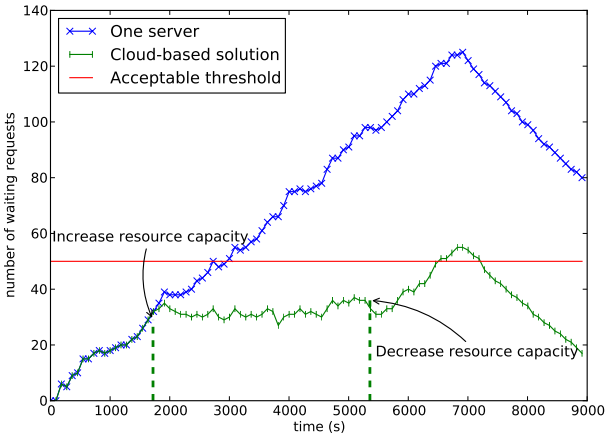

Fig. 16. Performance evaluation of our solution without/with the cloud

the single server configuration, and more resources are added or released to deal with load variation thanks to elasticity of the cloud. Figure 16 shows that with a single server the increasing load induces performance degrade of solution. Indeed, the number of waiting requests in queue will be high and may exceed acceptable threshold level. However, the using of cloud elasticity allows dealing with load variation to keep the system stable with acceptable threshold level of waiting requests.

VIII. Conclusion

In this paper, we addressed the challenge of data management in wireless sensor networks for patient supervision. We proposed a secure and scalable architecture that leverages cloud computing technology to dynamically scale storage resources via on demand provisioning. Furthermore, we proposed an innovative security scheme that eliminates potential security threats of medical data outsourcing and guarantees confidentiality, integrity without involving patients or doctors interventions. To implement complex and dynamic security

policies necessary to medical application, we developed a fine grained access control that combines attributes based encryption and symmetric cryptography. This combination reduced the management overhead and the encryption/decryption time as showed by our preliminary performance evaluation. In additional, we extend our access control to support emergency situations while preserving initial properties of our access control: secure, scalable and fine-grained. Finally, we carried out extensive simulations that allowed showing that our scheme provides an efficient, fine-grained and scalable access control in normal and emergency situations. In future works, we plan to use distributed attribute-based encryption to have multi healthcare authorities.

REFERENCES

[1] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, Oct. 2010.

[2] M. Li, W. Lou, and K. Ren, "Data security and privacy in wireless body area networks," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 51–58, Feb. 2010.

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Journal of Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, Jun. 2009.

[4] R. Liu and I. J. Wassell, "Opportunities and challenges of wireless sensor networks using cloud services," in *Proceedings of the workshop on Internet of Things and Service Platforms, IoTSP '11*, New York, NY, USA, 2011, pp. 41–47.

[5] V. Rajesh, J. M. Gnanasekar, R. S. Ponmagal, and P. Anbalagan, "Integration of wireless sensor network with cloud," in *International Conference on Recent Trends in Information, Telecommunication and Computing, ITC'10*, Kochi, India, Mar. 2010, pp. 321–323.

[6] C. O. Rolim, F. L. Koch, C. B. Westphall, J. Werner, A. Fracalossi, and G. S. Salvador, "A cloud computing solution for patient's data collection in health care institutions," in *Second International Conference on eHealth, Telemedicine, and Social Medicine, ETELEMED '10*, St. Maarten, Netherlands Antilles, Feb. 2010, pp. 95–99.

[7] W. Kurschl and W. Beer, "Combining cloud computing and wireless sensor networks," in *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services, iiWAS '09*, New York, NY, USA, 2009, pp. 512–518.

[8] K. Lee, D. Murray, D. Hughes, and W. Joosen, "Extending sensor networks into the cloud using amazon web services," in *IEEE International Conference on Networked Embedded Systems for Enterprise Applications, NESEA'10*, Suzhou, China, Nov. 2010, pp. 1–7.

[9] M. M. Hassan, B. Song, and E. Huh, "A framework of sensor-cloud integration opportunities and challenges," in *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, ICUIMC '09*, New York, NY, USA, 2009, pp. 618–626.

[10] A. M. Khattak, L. T. Vinh, D. V. Hung, P. T. H. Truc, L. X. Hung, D. Guan, Z. Pervez, M. Han, S. Lee, and Y. Lee, "Context-aware human activity recognition and decision making," in *12th IEEE International Conference on e-Health Networking Applications and Services, Healthcom'10*, Lyon, France, Jul. 2010, pp. 112–118.

[11] M. Barua, X. Liang, R. Lu, and X. Shen, "ESPAC: enabling security and patient-centric access control for eHealth in cloud computing," *International Journal of Security and Networks*, vol. 6, no. 2/3, pp. 67–76, Nov. 2011.

[12] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*. Berkeley, CA, USA: USENIX Association, 2003, pp. 29–42.

[13] E.-j. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing remote untrusted storage," *Network and distributed systems security, NDSS'03*, pp. 131—145, 2003.

[14] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient controlled encryption: ensuring privacy of electronic medical records," in *Proceedings of the 2009 ACM workshop on Cloud computing security, CCSW '09*, New York, NY, USA, 2009, pp. 103–114.

[15] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: management of access control evolution on outsourced data," in *Proceedings of the 33rd international conference on Very large data bases*, ser. VLDB '07, 2007, pp. 123–134.

[16] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and efficient access to outsourced data," in *Proceedings of the 2009 ACM workshop on Cloud computing security, CCSW '09*, New York, NY, USA, 2009, pp. 55–66.

[17] L. Ibraimi, M. Asim, and M. Petkovic, "Secure management of personal health records by applying attribute-based encryption," in *6th International Workshop on Wearable Micro and Nano Technologies for Personalized Health, pHealth'09*, Oslo, Norway, Jun. 2009, pp. 71–74.

[18] O. Akan, P. Bellavista, J. Cao, F. Dressler, D. Ferrari, M. Gerla, H. Kobayashi, S. Palazzo, S. Sahni, X. S. Shen, M. Stan, J. Xiaohua, A. Zomaya, G. Coulson, M. Li, S. Yu, K. Ren, and W. Lou, "Securing personal health records in cloud computing: Patient-Centric and Fine-Grained data access control in multi-owner settings," in *Security and Privacy in Communication Networks*. Springer Berlin Heidelberg, 2010, vol. 50, pp. 89–106.

[19] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy, SP '07*, Washington, DC, USA, 2007, pp. 321–334.

[20] A. Sahai and B. Waters, "Fuzzy Identity-Based encryption," in *Lecture Notes in Computer Science*, vol. 3494, 2005, pp. 457–473.

[21] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security, CCS '06*, New York, NY, USA, 2006, pp. 89–98.

[22] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS '10*, New York, NY, USA, 2010, pp. 261–270.

[23] Y. Z. K. R. e. W. L. M. Li, S. Yu, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, pp. 131 –143, 2013.

[24] A. D. Brucker, H. Petritsch, and S. G. Weber, "Attribute-Based encryption with Break-Glass," in *Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices*. Springer Berlin Heidelberg, 2010.

[25] T. M. Krishna K. Venkatasubramanian and S. K. S. Gupta, "Caac - an adaptive and proactive access control approach for emergencies for smart infrastructures," *ACM Transactions on Autonomous and Adaptive Systems Special Issue on Adaptive Security*, 2012.

[26] [Online]. Available: http://acsc.cs.utexas.edu/cpabe/