

Book chapter:

Protection of Data Confidentiality and Patient Privacy in Medical Sensor Networks

Ravi Sankar^a, Xuan Hung Le^{b,*}, Dongwen Wang^b

^aDepartment of Electrical Engineering, University of South Florida, Tampa, FL 33620

^bUniversity of Rochester Medical Center, Rochester, NY 14620

Table of Contents

1	Introduction	4
2	Issues and Challenges	5
2.1	Authentication of Data Collected and Transmitted from Medical Sensor Networks (Levels 1 and 2)	6
2.2	Access Management to Medical Information Stored at Central Servers (Level 3)	6
3	Review of Existing Methods and Limitations	7
3.1	Authentication of Data Collected and Transmitted from Medical Sensor Networks	7
3.2	Access Management to Medical Information Stored at Central Servers	7
4	Secure Authentication to Medical Sensing Information at Levels 1 and 2	8
4.1	Overview	8
4.2	Background	9
4.2.1	Elliptic Curve Discret Logarithm Problem (ECDLP)	9
4.2.2	Elliptic Curve Diffie-Hellman protocol (ECDH)	9
4.3	Methodology	10
4.3.1	Cryptographic Key Establishment	10
4.3.2	Authentication Protocol	10
4.4	Security Analysis	12
4.4.1	Mutual Authentication	12
4.4.2	Resilience to Replay Attacks	12
4.4.3	Denial-of-Service (DoS) Mitigation	12
4.5	Performance Evaluation	13
4.5.1	Analysis-based Performance Evaluation	13
4.5.2	Implementation-based Performance Evaluation	14
4.6	Discussion	15
5	Flexible Access Control Method to Medical Information at Level 3	17
5.1	Methodology	17
5.1.1	AOAC Model	18
5.1.2	Formal Representation	18
5.1.3	Activity Activation Rules	19
5.1.4	Permission Activation Rules	19
5.1.5	Privilege Delegation via Digital Credentials	20
5.1.6	Privilege Revocation	21
5.2	Implementation	22
5.2.1	System Design	22
5.2.2	Sample Scenario	24
5.3	Discussion	24
6	Conclusion	26
	References	26
	List of Figures	30

List of Tables 36

ABSTRACT

The security of data collected and transmitted from medical sensor networks, whether inside the networks, during transmission, or when stored at central servers, is a critical issue. In this chapter, existing methods, current issues and challenges in protecting medical data confidentiality and patient privacy are reviewed, after which studies on two critical issues are presented. First, a secure, lightweight user authentication scheme, Securing User Access to Medical Sensing Information (SecMed), is described. SecMed is a mutual authentication protocol where a healthcare professional can be authenticated to an accessed node (a PDA or medical sensor) and vice versa, ensuring that medical data is not exposed to an unauthorized person. It also ensures that medical data sent to healthcare professionals does not originate from a malicious node. SecMed is more scalable and requires less memory compared to symmetric key-based schemes. Second, a flexible and dynamic access control model, Activity-Oriented Access Control (AOAC), which is based on user activity to authorize access permissions, is presented. It is proposed that as user activities are recognized by using sensor devices, this model will help to enhance the authorization process and thus facilitate clinical performance. Security analysis and performance evaluation results are presented.

1 Introduction

Real-time patient health monitoring through wearable sensors can have many potential advantages in reducing healthcare costs, improving quality of life for patients, and providing effective management of chronic diseases (Bricon-Souf and Newman 2007, Haux 2010, Arkoulis et al. 2010, Isern et al. 2010, Koch 2006). Medical sensors, when properly placed on a patient or healthy person, can monitor vital signs and other physiological parameters while providing two-way, real-time feedback between the user and medical professionals (Haux 2006, Ng et al. 2006, Steele et al. 2009). With such continuous and remote monitoring of medical conditions in real-time, medical professionals could react to emergency situations such as heart attacks much more quickly (Sun et al. 2010, Lorincz et al. 2004, Maglogiannis and Hadjiefthymiades 2007). In addition, patient medical data could be collected more intensively for a longer period, and thus can help for more accurate diagnoses and better treatment (Pantazi et al. 2006, Garcia-Saez et al. 2009).

A typical medical sensor network is illustrated in Figure 1, and can be generally divided into three levels: Level 1 (Sensor Level), Level 2 (Coordination Level), and Level 3 (Access Level). Level 1 includes wearable/implantable and embedded micro sensors to collect health condition data as well as context information such as heart rate, body temperature, patient movements, room temperature, etc.

Level 2 is for collecting, aggregating, and transmitting data to central servers and medical staff. At Level 3, data is stored in central servers to which medical professionals and medical staff has access.

The security of data collected from medical sensor networks, whether inside the networks, during transmission, or when stored at central servers, is a critical issue (Smith and Eloff 1999, Samy et al. 2010, Al Ameen et al. 2010, Frenzel 2003, Li et al. 2010). User authentication and access control are two indispensable components that can address this issue. User authentication allows legitimate healthcare professionals to access medical information while declining access from malicious third parties. Access control is used to restrict authenticated healthcare professionals to only the data for which that they have privileges. In this chapter, two critical issues are discussed: (1) authenticating medical staff and central servers to access medical sensor networks and ensuring secure communication between them at Levels 1 and 2; and (2) controlling access to medical data stored in central servers at Level 3. The issues and challenges in ensuring the security of data within the network model shown in Figure 1 (Section 2) are first presented. Next, the existing methods and their limitations (Section 3) are discussed. Two approaches: SecMed (Securing Access to Medical Sensing Information) (Le et al. 2011a, Le et al. 2011b), which provides a secure and lightweight authentication mechanism for medical sensor networks (Section 4), and AOAC (Activity-Oriented Access Control) (Le et al. 2010a), which flexibly and securely grants access to medical sensing data stored at central servers (Section 5), are then presented. We conclude this chapter in Section 6.

[Figure 1]

2 Issues and Challenges

In this section, we discuss the issues and challenges in protecting medical data and ensuring security in its collection and transmission in medical sensor networks (Levels 1 and 2), as well as its storage and access at central servers (Level 3). Due to differences in network format, devices involved, and user activities at each level, we face unique challenges in authentication and access control.

2.1 Authentication of Data Collected and Transmitted from Medical Sensor Networks (Levels 1 and 2)

In general, wireless sensor networks are subject to the same attacks as any other wireless communication, but they also face additional challenges due to limitations in environment, resources, and their dynamic and distributed nature (Stuart et al. 2008). Typical challenges include:

- **Eavesdropping:** Since wireless communications are omni-directional, attackers can easily eavesdrop communications between Levels 1 and 2 by either listening to communication or querying network components. Therefore, there is an increased risk of disclosure of patient health information to an unauthorized public (Stuart et al. 2008, Ng et al. 2006).
- **Disruption:** Attackers can inject forged data transmitted from sensor networks, thereby providing false information about a patient's condition. In some scenarios such as a medical emergency, this can be a serious threat to human life (Stuart et al. 2008).
- **Limited resources:** Typical wireless sensor devices such as Crossbow's MICAz mote (MICAz Datasheet) are equipped with an 8 MHz processor and 0.5 MB of flash memory. Sensor devices for medical applications that are attached to or implanted inside human bodies are even smaller (Wong et al. 2006). Deployment of a security and privacy method is computationally expensive for such small devices.
- **Dynamic network topology:** The sensor network topology is constantly changing. As the topology changes, re-establishing security parameters such as a secret key should not generate too much overhead to ensure proper system performance.
- **Scalability:** The security architecture also needs to be scalable to account for varying numbers of mobile nodes as well as for making the best use of the scarce radio resources.

2.2 Access Management to Medical Information Stored at Central Servers (Level 3)

Once the medical data is collected and stored at central servers, various medical professionals can access it for patient care purposes. The Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule was intended to strike a balance between protecting the confidentiality of individually identifiable health information and preserving the legitimate use and disclosure of this information for important social goals (HIPAA 2000). This means that access to patient medical information must be

strictly controlled for only *need-to-know* medical professionals. The challenging issues are (Le et al. 2010a):

- **Confidentiality:** Medical information is highly sensitive. Confidentiality of such information must be strictly kept, yet at the same time it must not hinder patient care by denying legitimate access required by healthcare providers or other authorized personnel.
- **Access Facilitation:** The access control model should be able to reduce efforts of medical professionals to locate relevant documents according to their specific activity. First, the access control mechanism should be able to dynamically grant permission to a physician to access all data related to a healthcare activity. Second, the mechanism should be able to flexibly permit or revoke permission when the physician is or is not allowed to perform an activity.

3 Review of Existing Methods and Limitations

3.1 Authentication of Data Collected and Transmitted from Medical Sensor Networks

Previous research has mainly focused on how to seamlessly collect and wirelessly transmit medical sensing data within the context of an extremely resource-limited platform of medical sensor devices, aiming to reduce consumption of power, computation time, and network bandwidth (Arkoulis et al. 2010, <http://fiji.eecs.harvard.edu/CodeBlue> , <http://bsn.citris.berkeley.edu/home/> , <http://web.mit.edu/wockets/> , <http://smart.csail.mit.edu/> , <http://www.mobihealth.org/> , Le et al. 2009). Many security mechanisms have been proposed for wireless sensor networks based on symmetric key cryptography (SKC), due to its fast computation and energy efficiency. However, SKC is not scalable, as it requires a large memory for storing keys and a complicated key pre-distribution scheme. These barriers have impeded SKC from being practically deployed in healthcare. Public key cryptography-based schemes are ideal to overcome these challenges due to their high scalability, low memory requirements, easy key-addition/revocation for a new node, and no requirement of complicated key pre-distribution (Le et al. 2011a, Wang et al. 2006). However, it is computationally expensive to apply public key cryptography to resource-limited devices such as medical sensors (Gura et al. 2004).

3.2 Access Management to Medical Information Stored at Central Servers

To ensure the security of medical data stored at central servers, several access control methods have been proposed. The most popular access control model is Role-based Access Control (RBAC), introduced

in the early 90s (Ferraiolo et al. 2001). RBAC is conceptually simple. Access to computer system objects is based on a user's role in an organization. The authorizations are not assigned directly to particular users, but to roles. A role denotes a job function describing the authority and responsibility conferred on users assigned to that role. These approaches exploit role information to determine the set of access permissions. RBAC has been widely used thanks to its salient features such as simplicity, effectiveness and generalization. However, it lacks flexibility, and thus requires significant customization to meet the requirements of healthcare applications. Several models have been introduced to control user access to electronic patient records (Motta and Furuie 2003, Rodriguez et al. 2004). In (Motta and Furuie 2003), Motta *et al.* presented a *contextual role-based access control model* to increase patient privacy and the confidentiality of patient data, while being sufficiently flexible enough to consider specific cases. In (Rodríguez et al. 2004), the authors assumed that information required by specialists is highly dependent on their location. The paper presented a location-aware medical information system that was developed to provide access to resources such as patient records or the location of a medical specialist, based on the user's location. Sujansky *et al.* proposed a method to implement fine-grained access control for personal health records (Sujansky et al. 2010) through standard relational database queries instead of eXtensive Access Control Modeling Language (XACML). In (Peleg et al. 2008), a Situation-Based Access Control (SitBAC) was proposed to preserve patient privacy based on circumstances that match predefined patterns. Chen *et al.* developed an aspect-oriented design and implementation scheme to provide adaptable access control for Web-based EMR systems (Chen et al. 2010). It not only accommodates a wide range of fine-grained access control requirements, but also enforces them in a modular and easily adaptable manner without incurring extra performance overhead due to rule interpretation. Most of these studies have fully addressed the confidentiality issues of medical data, but not taken access facilitation issues into account.

4 Secure Authentication to Medical Sensing Information at Levels 1 and 2

4.1 Overview

In this section, the focus is on authentication, as it is a crucial component in addressing other issues, such as the eavesdropping and disruption mentioned in Section 2. After a successful authentication, central servers/medical staff and medical sensor devices are able to establish a trust relationship and a secret key for secure data transmission. The most challenging issue here is how to provide a secure authentication mechanism within the constraints of resource limitations, dynamic network topology, and

mobility. In the following, the proposed method SecMed (Securing Access to Medical Sensor Networks) (Le et al. 2011a) is presented. SecMed applies the *Elliptic Curve Discrete Logarithm Problem* (ECDLP) and the *Elliptic Curve Diffie-Hellman* (ECDH) protocol, two key components of Elliptic Cryptography (ECC). It is based on the communication scheme presented in Figure 1, which involves three parties: medical staff and a central server (say *A*), a Coordination Node (say *C*), and medical sensors (say *S*), as shown in Figure 2. Due to resource constraints, medical sensors are typically not equipped with any tamper-resistant hardware and they are susceptible to node capture attacks. In contrast, coordination nodes have more energy, a longer transmission range, and a higher data rate and thus can be equipped with a tamper-resistant hardware. This assumption is reasonable because the number of coordination nodes is relatively small (e.g., 20 for 1,000 medical sensor nodes (Du et al. 2007)) and thus the total associated cost for tamper-resistant hardware is relatively low. Medical staff can use a powerful computing device, such as a PDA, mobile phone, or laptop, as a coordination node to access data.

[Figure 2]

4.2 Background

4.2.1 *Elliptic Curve Discrete Logarithm Problem (ECDLP)*

Elliptic Curve Cryptography (ECC) was proposed independently by Miller (Miller 1986) and Koblitz (Koblitz 1987) in 1985. ECC is a public-key cryptography based on the algebraic structure of elliptic curves over finite fields. Compared to conventional public key cryptography such as RSA (Rivest et al. 1978), ECC achieves a much better performance with the same security level. An elliptic curve consists of the points satisfying the equation: $y^2 = x^3 + ax + b$, where x, y, a and b are elements in $GF(q)$ (a *Galois Field* of order q , where q is a prime). The elliptic curve group operation is closed under addition so that addition of any two points P and Q is also a point R in the group. If $P = Q$, then $R = P + P = 2 \times P$. The addition of multiple points P will give $R = k \times P$. ECC relies on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP); that is, given points P and Q of the group, it is practically infeasible to find a number k such as $Q = k \times P$.

4.2.2 *Elliptic Curve Diffie-Hellman protocol (ECDH)*

Elliptic Curve Diffie-Hellman (ECDH) protocol is a secret key exchanging protocol to establish a secret key between two parties who have no prior knowledge about each other. Based on ECDLP, a typical ECDH is built as shown in Figure 3. Initially, users Alice and Bob agree on a system base point P and

generate their own key-pair (Q_A, k_A) and (Q_B, k_B) (Q_A and Q_B , k_A and k_B are public and secret keys of *Alice* and *Bob*, respectively). To share a secret, *Alice* and *Bob* exchange their public keys, and then use their own private key, k_A and k_B respectively, to multiply the other's public key, i.e.

Alice computes: $R_A = k_A \times Q_B$, *Bob* computes: $R_B = k_B \times Q_A$.

Since $k_A \times Q_B = k_A \times (k_B \times P) = k_B \times (k_A \times P) = k_B \times Q_A$, thus $R_A = R_B = R(x_R, y_R)$.

The value x_R will be the secret key of *Alice* and *Bob*.

[Figure 3]

The protocol is secure because nothing is disclosed (except for the public keys and the base point P , which are not secret), and no party can derive the private key of the other unless it can solve the *Elliptic Curve Discrete Logarithm Problem* (Koblitz 1987).

4.3 Methodology

4.3.1 Cryptographic Key Establishment

To meet scalability requirements for a large number of sensor nodes, we propose a public key-based access control method using ECC (Miller 1986, Koblitz 1987). The first step is to load a pair of ECC public and private keys into each node. We assume that there is a trusted third party on the network called *Key Distribution Center* (KDC) to generate all the security materials (e.g., keys, certificates) and issue or revoke a user's access privileges (Wang et al. 2006, Le et al. 2009). Note that in this case KDC is not required to be online all the time. The proposed scheme uses *Elliptic Curve Diffie-Hellman* (ECDH) (NIST 2006) to establish a shared secret key between a sensing node and its coordination nodes. Initially, KDC selects a particular elliptic curve over a finite field $GF(p)$ (where p is prime) and publishes a base point P with a large order q (where q is also prime). It picks a random number $k \in GF(p)$ as a private key, and publishes its corresponding public key $Q = k \times P$. It also generates a random number $k_i \in GF(p)$ as a private key for a sensor s_i and generates a corresponding public key $Q_i = k_i \times P$. The key-pair $\{k_i, Q_i\}$ is then loaded into s_i . After this step, every node in the network has an ECC key-pair which will be used to establish secret (symmetric) key for secure communication.

4.3.2 Authentication Protocol

If a user or a central server (say, *Alice* or A) would like to access data from a particular sensor device or a group of sensor devices, or to send data on the coordination node, she must first obtain the base point P

from a KDC and generate her private key (k_A) and public key $Q_A = k_A \times P$. KDC issues a proper access control list ac_A via a certificate $cert_A$. Notations are shown in Table 1.

[Table 1]

[Figure 4]

The SecMed protocol is described in Figure 4, which includes the following steps.

- **STEP 1. $Alice \rightarrow C: (r)L, T_A, S_A$**

Alice selects a random number $r \in GF(p)$ which will be used as a session key with C and S , creates a secret key $L = h(k_{AC} || T_A)$ (where T_A is the current timestamp generated by *Alice*), and encrypts r with the key L (i.e. $(r)L$). *Alice* then signs this encrypted value along with its certificate (i.e. $S_A = sign_A((r)L || cert_A)$) and sends a combination $(r)L, T_A, S_A$ to the sensor S .

- **STEP 2. $C \rightarrow S: (r)M, T_C, ID_A, MAC_1$**

Upon receiving the message from *Alice*, node C first confirmed that the timestamp T_A is valid (i.e. by verifying if $T_A < T_{now}$, where T_{now} is current timestamp). Then it verifies *Alice's* signature S_A . If valid, then *Alice* is authenticated to C . *Alice's* certificate $cert_A$ is also verified by checking the validity of the access list ac_A which was assigned to her. *Alice* is authorized if $cert_A$ is valid. Node C now constructs a secret key $L = h(x_{AC} || T_A)$, and decrypts $(r)L$ to get r . It then generates a secret key $M = h(x_{CS} || T_C)$ (where T_C is the timestamp created by C), encrypts r , and builds a MAC value (i.e. $MAC_1 = MAC(x_{CS}, (r)M || ID_A)$). Finally, the coordination node C sends $(r)M, T_C, ID_A, MAC_1$ to S .

- **STEP 3. $S \rightarrow C: ID_S, MAC_2$**

When S receives the message, it confirms that if $T_C > T_{now}$. Then it verifies the MAC_1 value. If valid, it indicates that *Alice* is authenticated to S . After that, S constructs the secret key $M = h(x_{CS} || T_C)$ and decrypts $(r)M$ to get r . Using this secret key, S builds a MAC ($MAC_2 = MAC(r, ID_S)$) and sends it to *Alice*. Node S sends ID_S, MAC_2 to node C .

- **STEP 4. $C \rightarrow A: ID_C, ID_S, S_C$**

Node C verifies MAC_2 . If valid, it generates a signature $S_C = sign_C(ID_S || ID_C)$ and sends ID_C, ID_S, S_C to *Alice*. Upon receiving the ID_C, ID_S, S_C from C , *Alice* verifies C 's signature S_C . If valid, then S and C are authenticated.

4.4 Security Analysis

Note that the security level of the proposed protocol depends on the security level of the base algorithms including the ECC signature generation, message authentication code (CBC-MAC), and RC5 encryption algorithms. These algorithms have been proven to be secure in literature (Miller 1986, Koblitz 1987, Bellare et al. 2000, Rivest 1995). Therefore, within the scope of this paper, we focus on possible vulnerabilities to the proposed protocol.

4.4.1 Mutual Authentication

In *step 2* of the protocol, node C verifies the signature S_A . If S_A is valid, then the user is authenticated to C because only *Alice* can generate the signature S_A using her private key. Consequently, the user is also authenticated to sensor S because S trusts C (*step 3*). On the other hand, only S shares the secret key x_{CS} with C . This means that only S can decrypt $(r)M$ (where $M = h(x_{CS} \oplus T_C)$). So if S can achieve r from $(r)M$ to build $MAC_2 = MAC(r, ID_S)$, then S is authenticated to the user. The mutual authentication is provided through the trust relationship between *Alice* – C , and S – C .

4.4.2 Resilience to Replay Attacks

An authentication replay attack is a form of network attack in which a valid authentication is maliciously or fraudulently repeated or delayed. There are two possible ways for an adversary to launch replay.

First, the adversary can intercept the message sent out from *Alice* (in *step 1*) or from the sensor S (*step 3*). However, both cases are not possible in SecMed because C can easily detect an interception by verifying the timestamp T_A (*step 3*). If T_A is older than a predefined threshold (Wong et al. 2006, Song et al. 2007), it is invalid because it has been used for previous authentication. If T_A was changed, then S_A is not valid.

Second, the adversary can intercept the message sent out from C (*step 2*). Node S can detect an interception by confirming timestamp T_C . If T_C is older than the predefined threshold, it is not valid. If T_C has been changed to T_C^* , then the MAC_1^* is not consistent to MAC_1 .

4.4.3 Denial-of-Service (DoS) Mitigation

A Denial-of-Service attack (DoS) is an attempt to make a system resource unavailable to its intended users. An attacker can send a large number of authentication requests to deplete sensors' energy in order to disable sensors to collect and transmit data. Upon receiving the message from C (*step 2*), sensor node S

first checks the validity of timestamp T_C . If it is not valid, then S discards the message. Otherwise, it computes a MAC value to compare with the MAC_1 received. A message authentication code (MAC) generation, e.g. CBC-MAC algorithm, is very fast (Karlof et al. 2004). A CBC-MAC operation on Mica2 mote takes 3.12 ms (Karlof et al. 2004), while the ECC point multiplications used by Wang *et al.*'s scheme (Wang et al. 2006) (hereafter we call this HBQ for convenience) takes 3.5 s, which is about 1,121 times longer. Therefore, the proposed scheme significantly reduces DoS compared to HBQ.

4.5 Performance Evaluation

4.5.1 Analysis-based Performance Evaluation

To evaluate the performance of SecMed, it is compared with ENABLE (Le et al. 2011a) and HBQ (Wang et al. 2006) schemes based on theoretical analyses. Since *Alice* and coordination node C are powerful devices, the computational overhead is trivial compared to that of the sensors. For example, HP iPAQ handheld devices which are equipped with a 624 MHz processor and 256 MB of flash memory are much faster compared to MICAz devices (MICAz Datasheet) which are equipped with an 8 MHz processor and 0.5 MB of flash memory. In addition, handheld device batteries are frequently recharged. Therefore, we only consider computational overhead for sensors. Computational overhead (the computation time required by sensors, denoted by T) is used to analyze the performance. According to practical implementations on MICAz sensor motes (Chakravorty 2006, Karlof et al. 2004), the computational time of each security primitive is listed in Table 2.

[Table 2]

The total computational times of each proposed scheme, ENABLE and HBQ, are shown in Table 3. In SecMed, both user authentication and node authentication take $2T_{MAC}+T_H+T_{RC5}$. For user authentication, ENABLE requires $1T_{MAC}$ (approximately 3.12 ms), while the HBQ scheme requires $2T_H$, $2T_{MAC}$, $2T_{RC5}$, and $3T_{MUL}$ (total cost is approximately 2,451.04 ms). For node authentication, ENABLE requires $2T_{MAC}+1T_{RC5}+1T_H$, while the HBQ scheme does not support it. SecMed takes only 10.136 ms, which is less than ENABLE (13.256 ms) and HBQ (2,451.04 ms). The formula $E = U \cdot I \cdot t$ is used to estimate the energy consumption of security computations (Karlof et al. 2004). For the MICA2 sensor mote, when its processor is in active mode, $I = 8 \text{ mA}$. Typically, $U = 3.0 \text{ V}$ if two new AA batteries are used (Karlof et al. 2004). Total energy consumption shows that our approach consumes 0.24 mJ, which is more efficient than ENABLE (0.32 mJ) and HBQ (57.96 mJ).

[Table 3]

4.5.2 Implementation-based Performance Evaluation

To evaluate the performance of SecMed in actual implementation, a prototype system using Crossbow's MICAz sensor devices (MICAz Datasheet) was developed. The performance results were compared with the HBQ method. MICAz sensor devices have the same hardware capacities as common medical sensors, such as *CodeBlue* (<http://fiji.eecs.harvard.edu/CodeBlue>). Since HBQ protocol was only implemented on TelosB mote, its performance on MICAz motes was estimated in order to make a fair comparison. The estimation was based on realistic performance results of basic ECC operations on MICAz and TelosB motes (An and Peng 2008). It was concluded that MICAz's performance is 85% when compared with TelosB's. Therefore, we estimated the performance of HBQ scheme on MICAz to be 85% of the performance of TelosB as presented in (Wang et al. 2006).

[Table 4]

In the experiment, a user accessed the medical data on a sensing device one hundred times and the cumulative average computation time and energy consumption was computed. Table 4 shows the raw data of the cumulative average computation time and energy consumption for the SecMed and HBQ schemes. Figure 5 and Figure 6 present the data comparison in log scale. As shown in Figure 5, the first time when the user accessed the sensor, it took 2,121 ms, which was only 24.66% of access time for the HBQ scheme that took 8,602 ms. The secret key establishment between sensor and CN nodes using ECDH occurred only once during the first access. As the number of authentications and authorizations increased, the cumulative average delay significantly dropped. As shown in Figure 5, after the user accessed the mote 5 times, the cumulative average delay dropped to 5.05% of HBQ, and continually decreased to 0.41% as the user accessed 100 times.

[Figure 5]

[Figure 6]

The formula $E = U * I * t$ was used to compute the energy consumption, where U is voltage, I is current, and t is execution time. For the MICAz mote, when a processor is in active mode, $I = 8$ mA. Typically, $U = 3.0$ V because a MICAz mote is assumed to be powered by two new AA batteries. Figure 6 shows cumulative energy consumption. The first time when the user accessed the sensor, SecMed consumed 24.66% energy compared to the HBQ scheme, mostly due to the secret key establishment between the

sensor and its coordination node. As the number of access control times increased, there was barely any increase in the cumulative energy consumption of SecMed (almost constant), while HBQ's increases linearly. In conclusion, SecMed achieved better performance than HBQ. SecMed's computation time was only 24.66%, 5.05%, 2.60%, 1.38%, 0.66%, and 0.41% of HBQ when the user accessed 1, 5, 10, 20, 50, and 100 times, respectively. In terms of energy consumption, SecMed was only 24.66%, 5.05%, 2.60%, 1.38%, 0.64%, and 0.39% of HBQ, when the user accessed 1, 5, 10, 20, 50, and 100 times.

4.6 Discussion

Medical sensing systems possess unique features (thus bringing unique requirements) compared with general sensing systems such as home monitoring, surveillance, and military applications. The requirements include data security and patient privacy, high scalability with possible deployment of millions of medical sensors, and mobility, since patients may be moving in time. That brings a challenge because these requirements must be fulfilled along with other general requirements such as performance. The proposed method, SecMed, is especially developed to bring the medical sensing system into practice. It can be deployed in patient homes, nursing homes, offices, and hospitals. The security strength of the proposed scheme was evaluated based on mathematical analysis. A prototype was developed in the laboratory environment for the purpose of performance evaluation. It represents a typical medical sensing system that can be deployed in a practical healthcare application.

SecMed provides secure user access to medical sensing information. The security strength of SecMed partially relies on the security of the base algorithms ECC, CBC-MAC, and RC5. It provides a mutual authentication protocol where a healthcare professional can be authenticated to a medical sensor device and vice versa, ensuring that medical data is not exposed to an unauthorized person, and medical data sent to healthcare professionals is not originating from a malicious node. Also, analysis has shown that it is resilient to replay attacks, one of the most common attacks in authentication and access control protocols. In addition, when compared to other existing public key methods, SecMed is better at mitigating DoS attack, which is a serious attack in wireless sensor networks, but hard (almost impossible) to completely eliminate. This is because the computational-expensive operations are not performed on medical sensor devices themselves.

SecMed is more scalable and requires less memory compared to symmetric key-based methods. The scalability was achieved by applying the public key approach. Each node does not have to maintain a huge

number of secret shared keys with all other nodes in the networks (about $n(n-1)/2$ where n is the number of nodes in the network, which would be thousands to millions in reality). With scalability, SecMed enables the practical use of medical sensing systems for a possibly large number of patients in both metropolitan and rural areas. SecMed is more lightweight than the other existing public key-based methods. The performance was achieved by applying elliptic curve cryptography and other base security algorithms, which are efficient, yet proven secure enough for medical sensor systems. The proposed method also takes advantage of powerful nodes in the network to perform computational-expensive operations such as digital signature generation and verification, instead of operating on medical sensors themselves like the existing methods. Based on theoretical analyses and experimental data obtained from the prototype implementation, SecMed has been shown to achieve a significant performance improvement (at least 4 times better in terms of delay and energy consumption under a worst-case scenario) compared to existing methods.

It is necessary to note that in the proposed method CBC-MAC and RC5 were used as base security algorithms for message authentication code and encryption/decryption. This was due to their wide use in wireless sensor networks (Karlof et al. 2004, Le et al. 2009, Wang et al. 2006). However, other algorithms can be used as well. For example, *Advanced Encryption Standard* (AES) (Anon 1997), *SkipJack* (Skipjack 1998), or other recent encryption algorithms could be alternative options, as they are more secure and lightweight than public key encryption algorithms, and thus suitable for medical sensor devices. Using these methods would not affect the performance and security of the proposed method.

A limitation of SecMed is the potential weakness of the coordination nodes that are used to establish a trust relationship between users and medical sensor nodes. However, it is assumed that coordination nodes are equipped with tamper-resistant hardware, which is reasonable because coordination nodes are powerful enough to do so. Many current mobile devices provide this feature (Du et al. 2007). In addition, there have been a number of studies in literature dealing with the potential security breaches of mobile devices (Viega and Michael 2010, Anon 2008). Applying strong security algorithms to mobile devices is easier than applying them to medical sensor devices.

5 Flexible Access Control Method to Medical Information at Level 3

5.1 Methodology

Since Level 3 is composed of powerful computing devices like desktop computers, laptops, iPads, etc., many existing authentication solutions can be applied (Menezes et al. 1996). However, user access management is a non-trivial task. Protecting the confidentiality of health information, while at the same time allowing authorized physicians to access it conveniently, is a crucial requirement. Delivering health information at the *point-of-care* is a primary factor in increasing healthcare quality and cost efficiency. The current systems require considerable coordination effort of hospital professionals to locate relevant documents to support a specific activity. In this section, we present a flexible and dynamic access control model, *Activity-Oriented Access Control* (AOAC), which is based on user activity to authorize access permissions. An overview of AOAC is provided first, followed by the description of the AOAC model, privilege delegation mechanism, activity activation rules, and permission activation rules.

The nature of organizational authorizations is to determine who is allowed to do what. The AOAC model regulates the permissions based on specific activities. For example, *Dr. John* is permitted to carry out treatment of pneumonia for patient *Carol*. In order to accomplish an action for treatment, *Dr. John* needs access permissions to a number of resources. By connecting each activity with access permissions, the AOAC model highly supports user activity. To achieve this goal, the access control model is divided into three levels: user level, activity level, and privilege level, as illustrated in Figure 7. Each user holds a number of credentials (Chadwick and Otenko 2003) specifying attributes, such as hospital role, experience, and assignment. A credential can be a certificate/qualification (e.g. medical license), a hospital role (e.g. screening nurse), or an assignment from another user (e.g. a doctor is on leave, so he/she assigns another doctor is assigned to treat his/her patients during the leave). A user is authorized to perform a certain activity if the conditions are satisfied. The condition is defined as the *activity activation rule*. Each activity is associated with a number of access privileges. Those access privileges are needed to support the user to accomplish this activity. For example, the activity '*prescribe medicine for patient Carol*' requires access permission to medical record, X-ray images, blood test results, and medicine charts. This rule is defined as *permission activation rule*.

[Figure 7]

5.1.1 AOAC Model

Figure 8 shows the AOAC model. It is comprised of five administrative elements: (1) *users* (i.e. *users' attributes*), (2) *activities* and (3) *permissions* (i.e. privileges), where *permissions* are composed of (4) *objects* and (5) *operations*. *User* is a human interacting with a computing system. *Activity* is a human activity. It differs from the term '*task*' in a workflow system in the sense that it does not model nor control real-world human activities. An activity can be created and modified according to the desire of the user. *Object* is medical data as well as a system resource. An operation is an executable image of a program, for example '*read*', '*write*', '*execute*'. *Permission* is an authorization to perform certain operations within the system. *Constraint*, similar to the concept of constraint from the RBAC model (Ferraiolo et al. 2001), is defined as a predicate that is applied to a relation between two elements returning a value of '*acceptable*' or '*not-acceptable*'.

[Figure 8]

Basically, an AOAC protocol consists of the following steps:

1. The user logs onto the system through a *Single Sign-On* (SSO) authentication. Once authenticated, AOAC queries <uID, attrs> from a LDAP (Lightweight Directory Access Protocol) server to achieve the user's attributes based on the user's ID.
2. When AOAC detects that the user is performing an activity, the activity needs to be matched with the user's attributes. Based on the *Activity Activation Rule* (AAR) in the LDAP server, AOAC checks whether the user is allowed to perform the activity. If the attributes are not satisfied under AAR, AOAC will not take any further step.
3. If the user is allowed to perform the activity, then corresponding access permissions are queried from the policies. Medical data is then sent to the user.

A flowchart to summarize this process is shown in Figure 9.

[Figure 9]

5.1.2 Formal Representation

In the AOAC model, the *Activity Activation Rule* (AAR) is to allow a user to perform a certain activity if he/she holds a number of attributes including roles, user ID, and other credentials (e.g. an assignment). The *Permission Activation Rule* (PAR) is to provide access permissions to an activity. Technically, AOAC differs from RBAC in that a *role* in AOAC is considered as an attribute of a user and it alone cannot

decide what permission is allowed. It is not a bridge to connect between a user and permission. Permission is only directly connected to an activity which a user is allowed to perform. Therefore, if a user holds a number of roles, it will not cause the conflict of authorization that can occur in an RBAC model.

AOAC is formally represented in three prolog-like expressions:

- $ACT \vdash CON_1, CON_2, \dots, CON_n$: *activity activation rule*, where ACT is an activity, and CON_i is a condition including a user's attribute, such as a privilege to perform an activity or a privilege to access a resource.
- $PERM \vdash ACT, CC_1, CC_2, \dots, CC_m$: *permission activation rule*, if the user can activate an activity ACT under satisfied context constraints CC_1, CC_2, \dots, CC_m , then the corresponding permissions $PERM$ are granted.
- $A \rightarrow B$: *assignment* ($CRED(USERS_1, USERS_2), N$): *access delegation via assignment from User 1 (A) to User 2 (B)*.

5.1.3 Activity Activation Rules

The user must satisfy the conditions of the *activity activation rule* in order to be authorized for an activity. A prolog-like expression is used to formulate the *activity activation rule* as follows:

$$ACT \vdash CON_1, CON_2, \dots, CON_n$$

where ACT is an activity and CON_i is a condition including a user's attribute such as a privilege to perform an activity or a privilege to access a resource.

For example:

$$\begin{aligned} &treating_patient(John, Carol) \vdash med_doctor(John), screening_nurse(Alice), patient(Carol), \\ &treating_assignment(Alice, John) \end{aligned}$$

i.e., Dr. John is allowed to treat Carol if he holds a medical doctor license, and is appointed by screening nurse Alice to treat Carol.

5.1.4 Permission Activation Rules

Whenever a user initiates an activity, corresponding permissions are automatically activated if a number of context constraints are satisfied. A *context constraint* is defined as any requirement about contextual information, such as time and location. Context constraints play a key role in specifying context-sensitive policies. It is important to revoke privileges if contextual requirements are not met. A *permission activation rule* is formally defined as follows:

$$PERM \vdash ACT, CC_1, CC_2, \dots, CC_m$$

i.e., if the user can activate an activity *ACT* under satisfied context constraints CC_1, CC_2, \dots, CC_m , then the corresponding permissions *PERM* are granted.

For example, if *Alice* is authorized the activity “*taking_note*”, then she is permitted to access the EPR of *Carol* during work time:

$$read_EPR(Alice, Carol) \vdash taking_note(Alice), time(8:00, 17:00)$$

5.1.5 Privilege Delegation via Digital Credentials

Privilege delegation is a term indicating that *A* delegates to *B* a particular privilege. In AOAC, this is defined as *assignment*, where *A* is an *assigner* and *B* is an *assignee*. Formally:

$$A \rightarrow B: assignment(CRED(USERS_1, USERS_2), N),$$

where user *A* assigns user *B* an assignment (i.e. task/activity) wrapped in a credential *CRED*(.), where $USERS_1$ is a subset of users who can grant the credential, $USERS_2$ is a subset of users who can be so granted, *N* is the number of further assignments that an assigner can delegate to an assignee ($N=1, 2, 3, \dots$); if $N=1$, then *B* cannot grant this credential to anyone; if $N=\infty$: anyone possessing the credential *CRED* can grant it to another user (*B* must be indicated in *CRED*'s $USERS_1$).

For example,

$$A \rightarrow B: assignment(DIAGNOSE(\{doctors, nurses\}, \{doctors, nurses\}), 1).$$

Assignment occurs when a user grants a digital credential that directly or indirectly allows another user to perform one or more activities. The credential content may be an assignment of activities (*direct delegation*), or may be an assignment of role, qualification, etc., so that the other user may activate an activity (*indirect delegation*). The delegation approach in this study differs from *Appointment* (Bacon et al. 2002) in several aspects. First, not only user *A* may delegate the object right to user *B*, but once user *B* obtains the access permission through user *A*, *B* may also delegate the right to another user *C*, and so on (see Figure 10). We call this *multi-step delegation*. Second, our privilege delegation may be *restricted* or *unrestricted*. It means that user *A* can restrict how user *B* can further delegate the access right to user *C*. Delegation in *appointment* approach is only concerned with how to grant another user a credential to activate one or more roles without any concern about further delegation or restriction.

[Figure 10]

Digital credentials are the digital equivalent of paper-based credentials. An example a paper based credential could be a passport or a driver's license. A credential is a proof of qualification, competence, or

clearance that is attached to a person. Digital credentials prove something about their owner or deliver some information of their owner. Both may contain personal and professional information such as the person's name, birthplace, birthdate, job title, task assignment, or biometric information such as a picture or a finger print. The credential is abstracted into the form of *CRED* ($USERS_1, USERS_2$), where $USERS_1$ is a subset of users who can grant the credential, and $USERS_2$ is a subset of users who can be so granted.

It is essential to restrict which users can grant (or be granted) a credential. This can bring two advantages. First, this feature can be used for compliance with certain policies. For instance, some sensitive credentials must be only granted by superiors to senior personnel in the hospital. A specific example is that only oncologists are allowed to treat cancer patients, whereas others who are not specialized in cancer treatment would not be permitted to work on this assignment. Second, it can prevent the potential dangers if legitimate users abuse their privileges. For example, only a *screening nurse* can assign patients to doctors for treatment, and only a doctor can designate a nurse to administer medicine.

The *Privilege Delegation* can be implemented based on the X.509 *Privilege Management Infrastructure* (X.509 PMI) (Chadwick and Otenko 2003). X.509 PMI differs from X.509 Public Key Infrastructure (X.509 PKI) in that PMI holds *Attribute Certificates* (i.e. *Attribute Credentials*, which can be used for authorization), whereas X.509 PKI holds a *Public Key* (which is only for authentication). More importantly, a *Source of Authority* (SOA) can assign an attribute certificate, whereas only *Certification of Authority* (CA) can assign a Public Key (being a CA is a very specialized function and there are usually very few of them in an organization, whereas any attribute holder can be a SOA). A number of user attributes can be defined in each attribute credential. There are a number of ways a secure credential delegation mechanism such as PERMIS (*PrivilEge and Role Management Infrastructure Standards*) (Chadwick and Otenko 2003) can be implemented. PERMIS provides a cryptographically secure PMI using public key encryption technologies and X.509 Attribute Certificates to maintain users' attributes.

5.1.6 Privilege Revocation

In many situations, credentials should be revoked. For example, *A* delegated a credential to *B* for a particular task; if the task is accomplished, or *B* is transferred to another department, the credential should be revoked immediately. Privilege revocation can be done in four ways (Bacon et al. 2002): by its assigner only, by anyone active in the credential, by the assignee's resignation, or by the rule-based revocation (revocations can be carried out by the system itself). There are different reasons to revoke a privilege: time

duration on the credential is expired, constraint on the credential is violated, the task is completed, or the assigner session or the assignee's session has ended.

Two revocation types are proposed: *single-step revocation* (SR) and *multi-step revocation* (MR). SR is applied for *single-step delegations*. This means that *B* only delegates a credential to *C* without any further delegation from *C*. MR is applied for *multi-step delegations*. SR can be considered as a case of the *multi-step revocation* with the number of delegation steps set to one. There are different ways to cascade revocations. One of the simplest ways is based on credential identifiers (IDs) to revoke them. Thus, the original assigner (*A*) attaches a unique ID to each credential. Whenever a revocation is needed, *A* just indicates the credentials' IDs and the system will consider those credentials as invalid ones (see Figure 10).

5.2 Implementation

5.2.1 System Design

The AOAC system design is shown in Figure 11. The *Activity Recognition Manager* (ARM) detects a user's activities. The user's attributes, activities and activity activation rules are stored on a *Lightweight Directory Access Protocol* (LDAP) server. Permission activation rules are defined by *eXtensible Access Control Markup Language* (Lyon and Hsueh 2003). To be compliant with XACML standard, AOAC integrates a *Policy Enforcement Point* (PEP), and a *Policy Decision Point* (PDP). PEP makes the decision on requests and enforces authorization decisions. PEP incorporates a *Trigger* module to ease access. Once AOAC identifies authorized activity and access privileges, the trigger can deliver that information to users at the *point-of-care*. PDP evaluates applicable policy and renders an authorization decision. PDP is composed of three sub-components: the *User-Attribute Manager* (UAM), which retrieves users' attributes associated to their identifiers (uid) from the LDAP server; the *Attribute-Activity Manager* (AAM), which matches users' attributes to a set of allowed activities; and the *Activity-Permission Manager* (APM), which retrieves all access privileges for given activities from XACML policies. The *Admin Tool* (AT) is used to define activities and policies. Possible activities in the hospital are predicted and defined in advance. In fact, the bootstrapping phase is not much more onerous than other approaches such as RBAC. In RBAC, a number of roles need to be defined, and a number of access permissions related to each role. In a similar way, AOAC requires a number of activities and related access privileges to be defined. Technically this is feasible, as there is extensive research in defining specific clinical activities for management of certain types of patients and diseases (Wang and Peleg 2007, Wang et al. 2004, Boxwala

et al. 2004, Wang et al. 2002). Usually all possible activities are defined in advance, so that this rarely requires a change. This is similar to RBAC, in which all roles must be predefined and the system does not require many changes later.

[Figure 11]

All of AOAC's components were implemented in Java JDK 6.03 using *Eclipse 3.2.2*. The LDAP server was installed by *Apache Directory Server 1.0.2* (<http://apache.tt.co.kr/directory/apacheds/stable/1.0/1.0.2>). The AOAC policy structure written in XACML format is shown in Figure 12. In this policy specification, an activity is defined in the subject field. The policy defines what activity (*activity_id*) can access which data (*resource_id*) with what permission (*permission_id*). To connect AOAC components with XACML policies, query policies and assess users' permissions, Sun Inc.'s XACML Library (<http://dev.mysql.com/downloads/connector/j/5.1.htm>) was used.

[Figure 12]

There are two ways of activity recognition, as discussed in (Bardram 2009). First, context awareness can provide all or part of the user's intention. Second, users may explicitly state intentions. In the former approach, the *Activity Recognition (AR)* engine from the *Secured Sensor Network-integrated Cloud Computing for u-Lifecare (SC³)* system (Le et al. 2010b) will be incorporated. Currently, the activity recognition engines can detect a basic activity using a wearable sensor on the human wrist (Vinh et al. 2010) or using embedded sensors (Sarkar et al. 2010). Although the results have shown about 90 percent accuracy of detection, it is believed that it can reach 100 percent if cyber context information such as where, when and what computing device a user is using is involved. In the latter approach, users can explicitly state intentions. In the current prototype implementation, the system may request an activity description from a user by displaying a dialog-box "*Enter your current activity*". To facilitate fast access to information in an emergency, an auto-searching dialog-box, in which the system displays a list of similar activity names as the user is typing, was implemented. An example is illustrated in Figure 13. When the user is typing "*Treatment*", the box shows a list of similar activities, including "*Taking note of treatment progress for patient Carol*", "*Treatment of patient Carol*", etc. Additional discussion on implementation in real world clinical settings can be found in Section 5.3.

[Figure 13]

5.2.2 Sample Scenario

The use of the AOAC system and how it supports user activity in a ubiquitous hospital environment is illustrated with the following sample scenario.

Dr. John obtained a medical doctor license from the National Medical Council before he was employed by the hospital. A Human Resource (HR) staff person, who is in charge of *Dr. John's* employment, assigns him three attributes. UAM adds his hospital ID, department ID, and a credential specifying his medical doctor license to the LDAP server (*{uID, attrs}*).

One day, new patient *Carol* is hospitalized. After admission, she is assigned to *Dr. John* for diagnosis and treatment. UAM inserts new attributes to the LDAP server (*{uID, attrs}*). This is the *treatment assignment* from *Alice*.

At 8:00AM, *Dr. John* arrives at his office. The first thing he does, as usual, is to discuss the patient's progress with his colleagues. After logging in, the monitor displays the dialog-box "*Enter your current activity*". As he types "*Discussing patient progress*", the box displays all similar activities. *Dr. John* selects "*Discussing patient progress*". The agent sends the activity to PEP. PEP forwards the activity name along with *Dr. John's* hospital ID to PDP. At PDP, a list of permissions is sent to PEP. Since *Dr. John* is authorized to carry out that activity, he has full permission to access the patient's progress reports and treatment plans. The trigger module looks up the list, and then queries proper data from the hospital information database. The data is then displayed on the computer, as shown in Figure 14.

[Figure 14]

After that, *Dr. John* starts a visiting round to his patient ward with a nurse, say *Alice*, to see his patient *Carol*. He enters the activity "*Pneumonia treatment for patient Carol*" into the dialog-box on the monitor attached to the patient bed. Since *Dr. John* holds an assignment credential, he is authorized to perform that activity. After authorization, the monitor attached on the bed shows related medical information, including treatment history, detailed progress, and the latest symptoms. When the treatment is completed, *Alice* enters "*Taking notes on Carol*" to record all the treatment results and progress. In the above experiment, information is considered priority so that the display window containing the most critical information will be displayed at the top level of the screen.

5.3 Discussion

In general, there are different ways to protect the confidentiality of data. One way is to protect at the

communication level by a strong encryption algorithm such as *Advanced Encryption Standard* (AES). Another way is to protect at the application level, in which access by unauthorized users is prohibited by an access control mechanism. In this paper, the confidentiality of medical records is protected at the application layer by the proposed access control mechanism. The sample scenario was extended with different activities, such as medical treatment, diagnosis, admission/discharge, taking notes, medical prescription. When a physician was performing an activity, it was entered into the pop-up dialog box, and then corresponding data was shown. In several cases, the physicians were asked to intentionally enter an activity that they are not permitted to perform. Because the activity is not allowed, the corresponding access permissions were prohibited. Consequently, none of the information was provided to them. That indicates AOAC meets the first requirement mentioned in Section 2. The hospital information maintains confidentiality without hindering patient care by denying unauthorized access request from hospital employees. If the activity was permitted, the system correctly and instantly provided related information and services to physicians and nurses. This shows AOAC meets the second requirement mentioned in Section 4. It delivers appropriate information to physicians at the *point-of-care*. The experiment was repeated 70 times with different scenarios. It ran on a PC server (Pentium IV 3.2 GHz and 1 GB of RAM) with the average execution time of approximately 0.078 s. This included receiving an activity description, processing authorization, and displaying appropriate data to users. This means that AOAC is able to work in real-time. In the first phase of implementation, the code was not optimized. If the optimization is taken into account, it is believed that the performance will be significantly improved. Even when the user performs many activities simultaneously, the execution time will not be significantly increased.

An auto-searching dialog-box was implemented as a tentative solution for demonstration purpose. For real world implementation in clinical settings, activity recognition (the first approach mentioned in Section 5.3.1) will be integrated into the system, so that users (physicians, nurses, etc.) will not have to enter their activity every time they access. A mechanism to override the system for emergencies will be considered as well.

Also, the situation where many simultaneous activities are executed has been considered. The system performance for access control does not seem to be an issue, as it has been observed that the execution time was not increasing when the number of simultaneous activities was increased. If the experiment is deployed on a faster PC server, it is believed that the execution time would be decreased significantly.

Regarding activity recognition, the case when a user performs more than one activity at the same time has not been considered. Multiple activity recognition will add high complexity because the activity recognition depends on various sensing devices to gather activity context. When a user performs more than one activity, the sensing devices and recognition engine should be able to differentiate these activities. This will be addressed in a future study.

6 Conclusion

In this chapter, the issues and challenges in protecting confidentiality of medical data and patient privacy in medical sensor networks has been reviewed. To address these issues, two methodology studies have been presented. The first method, Securing User Access to Medical Sensor Networks (SecMed), addresses data confidentiality and patient privacy when data is collected from sensors and transmitted between nodes in the network. The second method, Activity-Oriented Access Control (AOAC), addresses data confidentiality and patient privacy when data is accessed from central servers. Implementation and evaluation have shown that these methods can effectively protect confidentiality of medical data and patient privacy. Although the work presented here was implemented on medical sensor networks, it is believed that the proposed methods could be used for other sensor network applications, such as military, agricultural surveillance, and homeland security.

References

- Al Ameen, M., Liu, J. and Kwak, K. (2010) 'Security and Privacy Issues in Wireless Sensor Networks for Healthcare Applications', *J Med Syst*, 10.1007/s10916-010-9449-4.
- An, L. and Peng, N. (2008) *TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks*, translated by 245-256, doi 10.1109/IPSIN.2008.47.
- Anon (1997) 'Advanced Encryption Standard', *Ieee Micro*, 17(1), 6-6.
- Anon (2008) 'Mobile device security is a concern', *Communications News*, 45(7), 6-6.
- Arkoulis, S., Spanos, D. E., Barbounakis, S., Zafeiropoulos, A. and Mitrou, N. (2010) 'Cognitive radio-aided wireless sensor networks for emergency response', *Measurement Science & Technology*, 21(12), doi 10.1088/0957-0233/21/12/124002.
- Bacon, J., Moody, K. and Yao, W. (2002) 'A model of OASIS role-based access control and its support for active security', *ACM Transaction on Information System Security*, 5(4), 51, doi 10.1145/581271.581276.
- Bardram, J. E. (2009) 'Activity-Based Computing for Medical Work in Hospitals', *Acm Transactions on Computer-Human Interaction*, 16(2), doi 10.1145/1534903.1534907.
- Bellare, M., Kilian, J. and Rogaway, P. (2000) 'The security of the cipher block chaining message authentication code', *Journal of Computer and System Sciences*, 61(3), 362-399, doi10.1006/jcss.1999.1694.

- Boxwala, A., Peleg, M., Tu, S., Ogunyemi, O., Zeng, Q., Wang, D., Patel, V., Greenes, R. and Shortliffe, E. (2004) 'GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines', *J Biomed Inform*, 37(3), 61, doi:10.1016/j.jbi.2004.04.002.
- Bricon-Souf, N. and Newman, C. R. (2007) 'Context awareness in health care: a review', *Int J Med Inform*, 76(1), 2-12, doi 10.1016/j.ijmedinf.2006.01.003.
- Chadwick, D. W. and Otenko, A. (2003) 'The PERMIS X.509 role based privilege management infrastructure', *Future Generation Computer Systems-the International Journal of Grid Computing Theory Methods and Applications*, 19(2), 277-289, doi Pii S0167-739x(02)00153-X.
- Chakravorty, R. (2006) *A programmable service architecture for mobile medical care*, translated by 5 pp.-536, 10.1109/PERCOMW.2006.11
- Chen, K., Chang, Y. C. and Wang, D. W. (2010) 'Aspect-oriented design and implementation of adaptable access control for electronic medical records', *Int J Med Inform*, 79(3), 181-203, doi 10.1016/j.ijmedinf.2009.12.007.
- Du, X. J., Guizani, M., Xiao, Y. and Chen, H. H. (2007) 'Two tier secure routing protocol for heterogeneous sensor networks', *Ieee Transactions on Wireless Communications*, 6(9), 3395-3401, doi 10.1109/Twc.2007.06095.
- Ferraiolo, D. R., Sandhu, S., Kuhn, D. R. and Chandramouli, R. (2001) 'Proposed NIST standard for role-based access control', *ACM Transactions on Information System Security*, 4(3), 224-274, doi 10.1145/501978.501980.
- Frenzel, J. C. (2003) 'Data security issues arising from integration of wireless access into healthcare networks', *J Med Syst*, 27(2), 163-75, doi 10.1023/A:1021865011765.
- Garcia-Saez, G., Hernando, M. E., Martinez-Sarriegui, I., Rigla, M., Torralba, V., Brugues, E., de Leiva, A. and Gomez, E. J. (2009) 'Architecture of a wireless Personal Assistant for telemedical diabetes care', *Int J Med Inform*, 78(6), 391-403, doi 10.1016/j.ijmedinf.2008.12.003.
- Gura, N., Patel, A., Wander, A., Eberle, H. and Shantz, S. C. (2004) 'Comparing elliptic curve cryptography and RSA on 8-bit CPUs', *Cryptographic Hardware and Embedded Systems - Ches 2004, Proceedings*, 3156, 119-132, doi 10.1.1.69.2593.
- Haux, R. (2006) 'Individualization, globalization and health-about sustainable information technologies and the aim of medical informatics', *Int J Med Inform*, 75(12), 795-808, doi 10.1016/j.ijmedinf.2006.05.045.
- Haux, R. (2010) 'Medical informatics: past, present, future', *Int J Med Inform*, 79(9), 599-610, doi S1386-5056(10)00114-0 [pii]10.1016/j.ijmedinf.2010.06.003.
- HIPAA (2000) 'Standards for privacy of individually identifiable health information: Final Rule.', *Fed Regist*, 65(250), 829.
- The UC Berkeley Body Sensor Network, <http://bsn.citris.berkeley.edu/home/>
- Harvard University's Wireless Sensors for Medical Care (CodeBlue), <http://fiji.eecs.harvard.edu/CodeBlue>
- MIT's Scalable Medical Alert Response Technology (SMART), <http://smart.csail.mit.edu/>
- MIT's Sensors and Software for real-Time Recognition on Mobile Phones (WOCKETS), <http://web.mit.edu/wockets/>
- European Mobile Healthcare (MobilHealth), <http://www.mobihealth.org/>
- Isern, D., Sanchez, D. and Moreno, A. (2010) 'Agents applied in health care: A review', *Int J Med Inform*, 79(3), 145-66, doi S1386-5056(10)00016-X [pii]10.1016/j.ijmedinf.2010.01.003.
- James, N. L., Harrison, D. G. and Nerem, R. M. (1995) 'Effects of shear on endothelial cell calcium in the presence and absence of ATP', *Faseb J*, 9(10), 968-73,
- Karlof, C., Sastry, N. and Wagner, D. (2004) 'TinySec: a link layer security architecture for wireless sensor networks', in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, Baltimore, MD, USA, 1031515: ACM, 162-175, doi 10.1145/1031495.1031515.

- Koblitz, N. (1987) 'Elliptic Curve Cryptosystems', *Mathematics of Computation*, 48(177), 203-209.
- Koch, S. (2006) 'Home telehealth-current state and future trends', *Int J Med Inform*, 75(8), 565-76, doi S1386-5056(05)00188-7 [pii]10.1016/j.ijmedinf.2005.09.002.
- Le, X. H., Khalid, M., Sankar, R. and Lee, S. (2011a) 'Energy-efficient mutual authentication and access control for wireless sensor networks', *Journal of Networks (Academy Publisher)*, doi10.4304/jnw.6.3.355-364.
- Le, X. H., Khalid, M., Sankar, R. and Wang, D. (2011b) 'Development of Secure User Access to Medical Sensing Information', *International Journal of Medical Informatics (under review)*.
- Le, X. H., Lee, S., Butun, I., Khalid, M., Sankar, R., Kim, M., Han, M., Lee, Y. K. and Lee, H. (2009) 'An Energy-Efficient Access Control Scheme for Wireless Sensor Networks based on Elliptic Curve Cryptography', *Journal of Communications and Networks*, 11(6), 599-606, doi10.4304/jnw.6.3.355-364.
- Le, X. H., Lee, S., Lee, Y.-K., Lee, H., Khalid, M. and Sankar, R. (2010a) 'Activity-oriented access control to ubiquitous hospital information and services', *Information Sciences (Elsevier)*, 180(16), 2979-2990, doi:10.1016/j.ins.2010.04.020.
- Le, X. H., Lee, S., True, P. T. H., Vinh, L. T., Khattak, A. M., Han, M., Hung, D. V., Hassan, M. M., Kim, M., Koo, K.-H., Lee, Y.-K. and Huh, E.-N. (2010b) 'Secured WSN-integrated cloud computing for u-life care', in *Proceedings of the 7th IEEE conference on Consumer communications and networking conference*, Las Vegas, Nevada, USA, 1834374: IEEE Press, 702-703, doi 10.1109/CCNC.2010.5421618.
- Li, M., Lou, W. J. and Ren, K. (2010) 'Data Security and Privacy in Wireless Body Area Networks', *Ieee Wireless Communications*, 17(1), 51-58, doi 10.1109/MWC.2010.5416350.
- Lorincz, K., Malan, D. J., Fulford-Jones, T. R. F., Nawoj, A., Clavel, A., Shnayder, V., Mainland, G., Welsh, M. and Moulton, S. (2004) 'Sensor networks for emergency response: Challenges and opportunities', *Ieee Pervasive Computing*, 3(4), 16-23, doi 10.1109/MPRV.2004.18.
- Lyon, C. J. and Hsueh, W. A. (2003) 'Effect of plasminogen activator inhibitor-1 in diabetes mellitus and cardiovascular disease', *Am J Med*, 115 Suppl 8A, 62S-68S.
- Maglogiannis, I. and Hadjiefthymiades, S. (2007) 'EmerLoc: location-based services for emergency medical incidents', *Int J Med Inform*, 76(10), 747-59, doi S1386-5056(06)00200-0 [pii]10.1016/j.ijmedinf.2006.07.010.
- Menezes, A., Oorschot, P. v. and Vanstone, S. (1996) *Handbook of Applied Cryptography*, CRC Press.
- MICAz Datasheet, www.xbow.com.
- Miller, V. S. (1986) 'Use of Elliptic-Curves in Cryptography', *Lecture Notes in Computer Science*, 218, 417-426.
- Motta, G. H. and Furuie, S. S. (2003) 'A contextual role-based access control authorization model for electronic patient record', *IEEE Trans Inf Technol Biomed*, 7(3), 202-7, doi 10.1109/TITB.2003.816562.
- Ng, H. S., Sim, M. L. and Tan, C. M. (2006) 'Security issues of wireless sensor networks in healthcare applications', *Bt Technology Journal*, 24(2), 138-144, doi 10.1007/s10550-006-0051-8.
- NIST (2006) 'Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography '.
- Pantazi, S. V., Kushniruk, A. and Moehr, J. R. (2006) 'The usability axiom of medical information systems', *Int J Med Inform*, 75(12), 829-39, doi S1386-5056(06)00154-7 [pii]10.1016/j.ijmedinf.2006.05.039.
- Peleg, M., Beimel, D., Dori, D. and Denekamp, Y. (2008) 'Situation-Based Access Control: privacy management via modeling of patient data access scenarios', *J Biomed Inform*, 41(6), 1028-40, doi S1532-0464(08)00050-6 [pii]10.1016/j.jbi.2008.03.014.
- Rivest, R. L. (1995) 'The Rc5 Encryption Algorithm', *Dr Dobbs Journal*, 20(1), 146.
- Rivest, R. L., Shamir, A. and Adleman, L. (1978) 'Method for Obtaining Digital Signatures and Public-Key Cryptosystems', *Communications of the Acm*, 21(2), 120-126, doi 10.1145/359340.359342.

- Rodriguez, M. D., Favela, J., Martinez, E. A. and Muñoz, M. A. (2004) 'Location-aware access to hospital information and services', *IEEE Transactions on Information Technology in Biomedicine*, 4, 448–455, doi 10.1109/TITB.2004.837887.
- Samy, G. N., Ahmad, R. and Ismail, Z. (2010) 'Security threats categories in healthcare information systems', *Health Informatics J*, 16(3), 201-9, doi 16/3/201 [pii]10.1177/1460458210377468.
- Sarkar, A. M. J., Lee, Y. K. and Lee, S. (2010) 'A Smoothed Naive Bayes-Based Classifier for Activity Recognition', *Iete Technical Review*, 27(2), 107-119, doi 10.4103/0256-4602.60164.
- Skipjack (1998) 'SkipJack and KEA Algorithm Specifications'.
- Smith, E. and Eloff, J. H. (1999) 'Security in health-care information systems-current trends', *Int J Med Inform*, 54(1), 39-54, doi S1386-5056(98)00168-3 [pii].
- Song, H., Zhu, S. C. and Cao, G. H. (2007) 'Attack-resilient time synchronization for wireless sensor networks', *Ad Hoc Networks*, 5(1), 112-125, doi 10.1016/j.adhoc.2006.05.016.
- Steele, R., Lo, A., Secombe, C. and Wong, Y. K. (2009) 'Elderly persons' perception and acceptance of using wireless sensor networks to assist healthcare', *International Journal of Medical Informatics*, 78(12), 788-801, DOI 10.1016/j.ijmedinf.2009.08.001.
- Stuart, E., Moh, M. and Teng-Sheng, M. (2008) *Privacy and security in biomedical applications of wireless sensor networks*, translated by 1-5, doi 10.1109/ISABEL.2008.4712575.
- Sujansky, W. V., Faus, S. A., Stone, E. and Brennan, P. F. (2010) 'A method to implement fine-grained access control for personal health records through standard relational database queries', *J Biomed Inform*, 43(5 Suppl), S46-50, doi S1532-0464(10)00111-5 [pii]10.1016/j.jbi.2010.08.001.
- Sun, J. Y., Fang, Y. G. and Zhu, X. Y. (2010) 'Privacy and Emergency Response in E-Healthcare Leveraging Wireless Body Sensor Networks', *Ieee Wireless Communications*, 17(1), 66-73, doi 10.1109/MWC.2010.5416352
- Viega, J. and Michael, B. (2010) 'Mobile Device Security Introduction', *Ieee Security & Privacy*, 8(2), 11-12, doi 10.1109/MSP.2010.76.
- Vinh, L., Lee, S., Le, H., Ngo, H., Kim, H., Han, M. and Lee, Y.-K. (2010) 'Semi-Markov conditional random fields for accelerometer-based activity recognition', *Applied Intelligence*, 1-16, doi 10.1007/s10489-010-0216-5.
- Wang, D. and Peleg, M. (2007) 'Using GLIF and GLEE to facilitate knowledge management in development of clinical decision support systems', *Medinfo*, 12, 70.
- Wang, D., Peleg, M., Tu, S., Boxwala, A., Greenes, R., Patel, V. and Shortliffe, E. (2002) 'Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models', *Int J Med Inform*, 68(1), 12, doi:10.1016/S1386-5056(02)00065-5.
- Wang, D., Peleg, M., Tu, S., Boxwala, A., Ogunyemi, O., Zeng, Q., Greenes, R., Patel, V. and Shortliffe, E. (2004) 'Design and implementation of the GLIF3 guideline execution engine', *J Biomed Inform*, 37(5), 18, doi:10.1016/j.jbi.2004.06.002.
- Wang, H., Sheng, B. and Li, Q. (2006) 'Elliptic curve cryptography-based access control in sensor networks', *Int J Secur Netw*, 1(3/4), 127-137, doi 10.1504/ijsn.2006.011772.
- Wong, K. H. M., Yuan, Z., Jiannong, C. and Shengwei, W. (2006) *A dynamic user authentication scheme for wireless sensor networks*, translated by 8 pp., doi 10.1109/SUTC.2006.1636182.

List of Figures

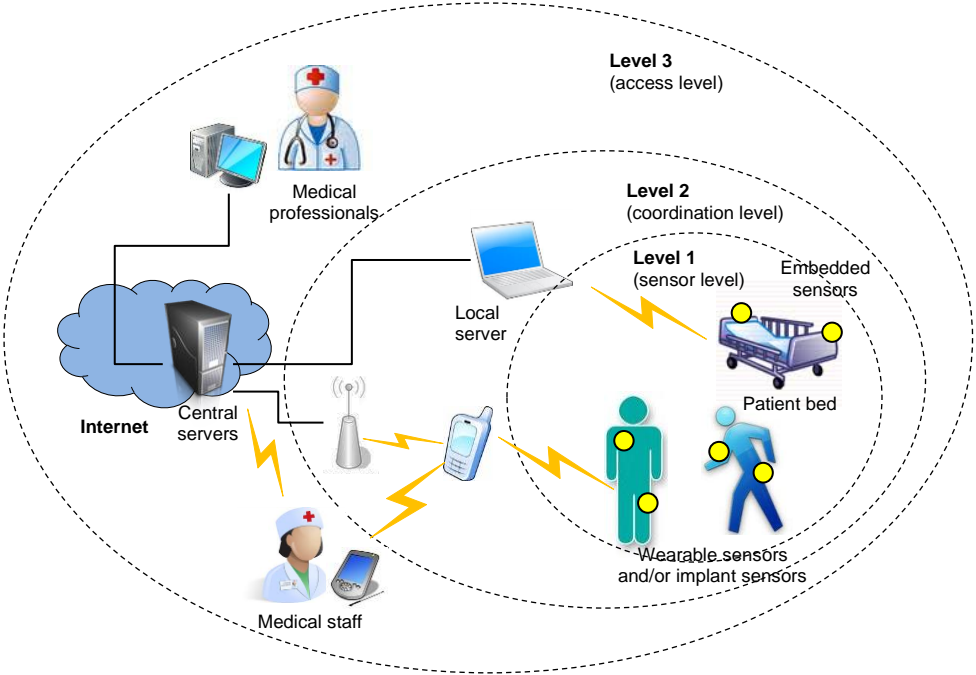


Figure 1. Typical sensor network in healthcare

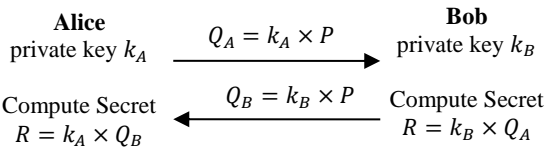


Figure 2. ECDH key exchange protocol

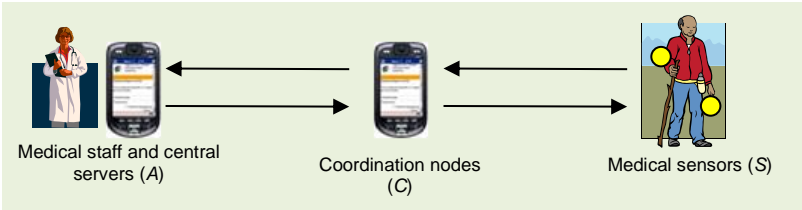


Figure 3. Modeling medical monitoring systems

```

Alice computes      :  $L = h(x_{AC} || T_A)$ 
                   :  $S_A = \text{sign}_A((r)L || \text{cert}_A)$ 

Step 1. Alice  $\rightarrow$  C :  $(r)L, T_A, S_A$ 
C computes          : check if  $T_A$  is valid?,
                   :  $\text{verify}(S_A), \text{verify}(\text{cert}_A)$ ,
                   : computes  $L = h(x_A || T_A)$ ,
                   :  $r = \text{decrypt}((r)L)$ ,
                   : computes  $M = h(x_{CS} || T_C)$ ,
                   :  $\text{MAC}_1 = \text{MAC}(x_{CS}, (r)M || ID_A)$ 

Step 2. C  $\rightarrow$  S      :  $(r)M, T_C, ID_A, \text{MAC}_1$ 
S computes          : check if  $T_C$  is valid?
                   :  $\text{verify}(\text{MAC}_1)$ : IF  $\text{MAC}_1$  is valid. THEN Alice is authenticated
                   : computes  $M = h(x_{CS} || T_C)$ ,
                   :  $r = \text{decrypt}((r)M)$ 
                   :  $\text{MAC}_2 = \text{MAC}(r, ID_S)$ 

Step 3. S  $\rightarrow$  C      :  $ID_S, \text{MAC}_2$ 
C computes          :  $\text{verify}(\text{MAC}_2)$ : IF  $\text{MAC}_2$  is valid THEN S is authenticated.
                   :  $S_C = \text{sign}_C(ID_S || ID_C || \text{cert}_C)$ 

Step 4. C  $\rightarrow$  Alice  :  $ID_C, ID_S, S_C$ 
Alice computes      :  $\text{verify}(S_C)$ : IF  $S_C$  is valid THEN S and C is authenticated.

```

Figure 4. SecMed protocol

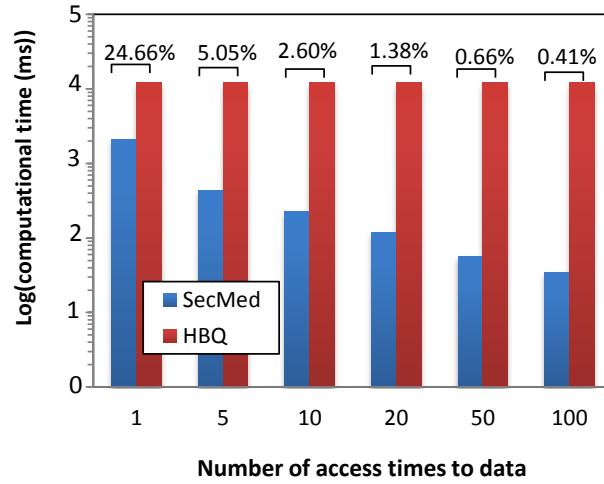


Figure 5. Log_{10} of computational time (ms)

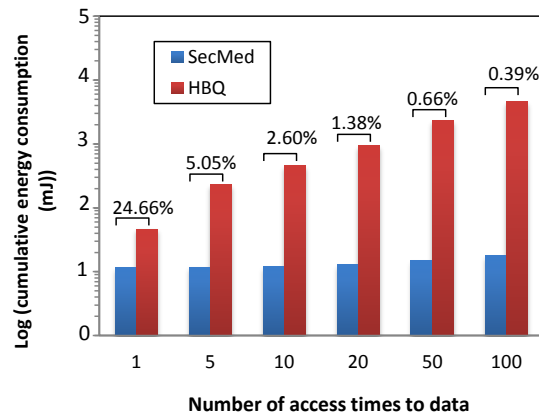


Figure 6. Log₁₀ of cumulative energy consumption (mJ)

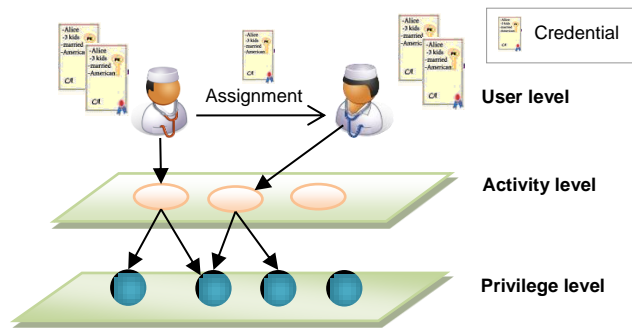


Figure 7. AOAC abstraction levels

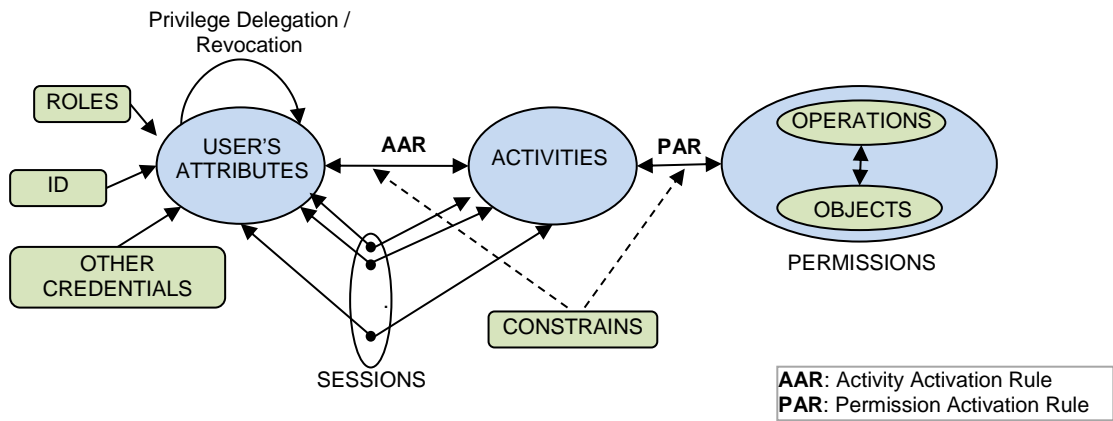


Figure 8. AOAC model

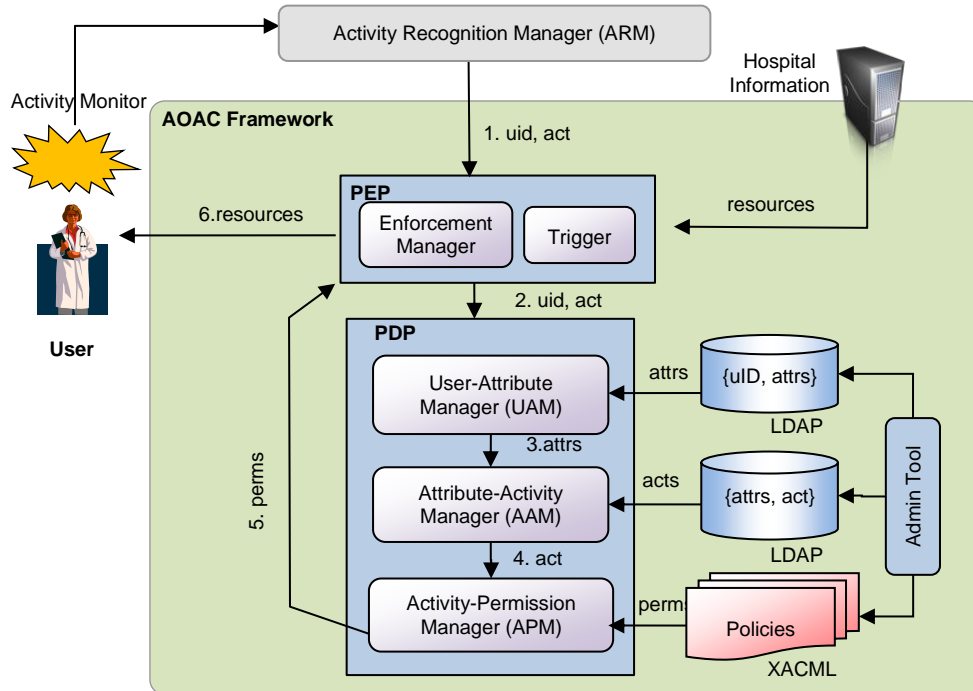
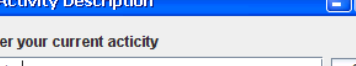


Figure 11. AOAC system design

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy PolicyId="AOACpolicy100" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:ordered-permit-overrides">
  <Description>This policy was automatically created by AOAC engine </Description>
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType = "http://www.w3.org/2001/XMLSchema#string">
            activity id</AttributeValue>
          <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-
            id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue DataType = "http://www.w3.org/2001/XMLSchema#anyURI">
            resource id</AttributeValue>
          <ResourceAttributeDesignator AttributeId =
            "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
            DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Rule RuleId="CommitRule" Effect="Permit">
    <Target>
      <Subjects>
        <AnySubject/>
      </Subjects>
      <Resources>
        <AnyResource/>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              permission id</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-
              id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
</Policy>
```

```
</Action>
</Actions>
</Target>
</Rule>
<Rule RuleId="FinalRule" Effect="Deny"/>
</Policy>
```

Figure 12. AOAC policy structure written in XACML



Activity Description

Enter your current activity

treatm GO

- Taking note of treatment progress for patient Carol
- Taking note of treatment progress for patient Hassen
- Taking note of treatment progress for patient Jessen
- Treatment of patient Carol
- Treatment of patient Hassen

Figure 13. A list of auto-searched activities is shown as the user is typing

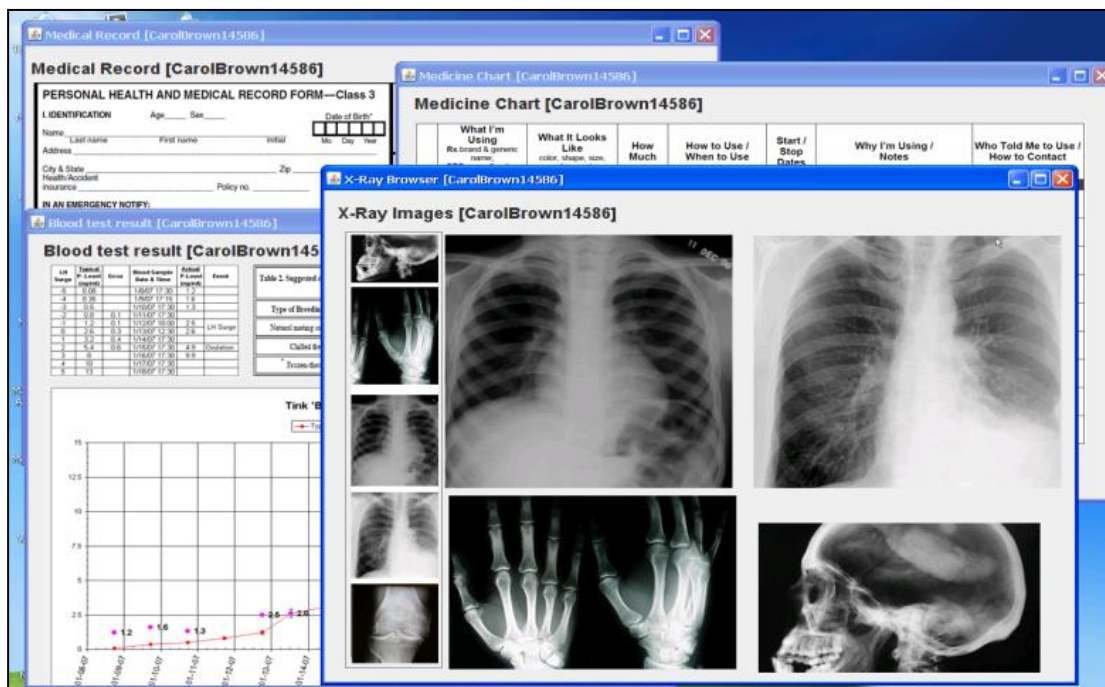


Figure 14. Data supported for treatment

List of Tables

Table 1 Notations

Symbol	Description
$A, C, \text{ and } S$	User, coordination node, and medical sensor device, respectively
ID_A	Identification of entity A
k_{AB}	Shared secret key between two entities A and B
ac_A	Access control list issued to entity A
$sign_A(m)$	Message m is signed by entity A
$cert_A$	Digital certificate of entity A
$A \rightarrow B : m$	Entity A sends entity B a message m
$(m)K$	Symmetric encryption of message m with key K
$MAC(K, m)$	A message authentication code of message m with key K
$h(m)$	Hashing value of message m
$ $	Concatenation

Table 2 Computational time on MICAz sensor

Notation	Description	Time (ms)
T_H	Time to perform one-way hash function (e.g. SHA-1)	3.636
T_{MAC}	Time to generate MAC value (e.g. CBC-MAC)	3.12
T_{RC5}	Time to encrypt/decrypt by RC5	0.26
T_{MUL}	Time to perform ECC point multiplication	810

Table 3 Comparison of computational time and energy consumption

	SecMed	ENABLE	HBQ
User authentication	$2T_{MAC} + T_H + T_{RC5}$	T_{MAC}	$2T_H + 2T_{MAC} + T_{RC5} + 3T_{MUL}$
Node authentication		$2T_{MAC} + 1T_{RC5} + 1T_H$	None
Total	$2T_{MAC} + T_H + T_{RC5}$	$2T_{MAC} + 1T_{RC5} + 1T_H$	$2T_H + 2T_{MAC} + 2T_{RC5} + 3T_{MUL}$
Total execution time	10.136ms (23.54% and 99.58% reduction compared to ENABLE and HBQ respectively)	13.256ms	2,415.04ms
Energy consumption	0.24mJ (25.00% and 99.59% reduction compared to ENABLE and HBQ respectively)	0.32mJ	57.96mJ

Table 4 Raw data of computational time and energy consumption

#Access	Average computational time (ms)			Cumulative energy consumption (mJ)		
	SecMed	HBQ	SecMed/HBQ	SecMed	HBQ	SecMed/HBQ
1	2,121.0	8,602.0	24.66 %	11.5	46.5	24.66%
5	434.6	8,602.0	5.05%	11.7	232.3	5.05%
10	223.8	8,602.0	2.60%	12.1	464.5	2.60%
20	118.4	8,602.0	1.38%	12.8	929.0	1.38%
50	56.9	8,602.0	0.66%	14.8	2,322.5	0.64%
100	35.0	8,602.0	0.41%	18.1	4,645.1	0.39%

