DESIGNING ACCESS CONTROL MODEL AND ENFORCING SECURITY POLICIES USING PERMIS FOR A SMART ITEM E-HEALTH SCENARIO

HASAN MAHMUD
Department of Computer Science and Information Technology (CIT), Islamic University of Technology (IUT),
Board Bazar, Gazipur-1704, Bangladesh
hasan@iut-dhaka.edu

S. M. DIDAR-AL-ALAM
Department of Computer Science and Information Technology (CIT), Islamic University of Technology (IUT),
Board Bazar, Gazipur-1704, Bangladesh
didar522@gmail.com

MD. SARWAR MORSHED
Department of Computer Science and Information Technology (CIT), Islamic University of Technology (IUT),
Board Bazar, Gazipur-1704, Bangladesh
morshed4@iut-dhaka.edu

MOHAMMAD OBAIDUL HAQUE
Department of Information Engineering and Computer Science, University of Trento (UNITN),
Trento, Italy
mohammad.haque@studenti.unitn.it

MD. KAMRUL HASAN
Computer and Communication Engineering department, Patuakhali Science and Technology University (PSTU),
Patuakhali, Bangladesh
kamrul@pstu.ac.bd

Abstract:
Sensor networks in medical applications are the edge component of the health care system. This type of network comprises a significant number of different sensor devices called smart items which are tightly connected and interacts continuously. Smart items measure the values of different health variables and send them through suitable communication interface. Measured data forms a crucial part of personal health information which must be protected from the aspect of integrity and patient privacy. This paper presents security concerns regarding an e-health application built on a wireless sensor network environment. We have designed security policies for the e-health scenario and enforced those policies through PERMIS (Privilege and Role Management Infrastructure Standards) that uses RBAC (Role Based Access Control), X.509 and PMI (Privilege Management Infrastructure) to implement authentication and authorization scheme. The concept of reference monitor has also been shown with an example policy implementation using ConSpec policy language.

Keywords: Smart items; E-health application; Security policy; Reference monitor; Access control model.

1. Introduction

Medical Emergency Response is an application of smart items in the domain of sensor network. Smart Item is a device that provides data about itself or the object with which it is associated or about its physical environment
(temperature, humidity etc.) and can transmit this information through a communication interface. Our work is about a Health care system that consists of smart items and is called e-health monitoring program. The application framework monitors patient after hospitalization, e.g., after cardiac infarction. The application scenario is a subject to increased risk and probability of attack, since their components (i.e., the sensors) communicate and interact in an open and thus insecure environment where security threats, e.g., illegal disclosure of information, transmission of false data, authentication and/or authorization violations, must be taken carefully into account. The application scenario has taken from the case study named, “Smart Item-Medical Emergency Response” by Serenity consortium [Serenity consortium – MER technical report (2007)].

Health care scenario like, monitoring patient after hospitalization involves a multitude of actors (patients, doctors, social workers, software framework, etc), devices (health-care terminals, PDAs, phones, computers, temperature sensors, etc), patient databases, and applications remotely cooperating to enhance and help daily lives of patients as well as the quality and the effectiveness of health care systems and institutions. In this kind of setting patients’ vital signals must be mediated and integrated by other signals and information coming both from personal characteristics (e.g., age, sex, family history) and from the environmental context (e.g., the patient is asleep or awake, walking or sit down, at work or at home; the season and the temperature, etc) with respect to the degree of alert detected, must be carefully considered in this smart items scenario. Smart items application designed to supervise the patient’s status after hospitalization should be as much as possible transparent to the patient yet accomplishing all the necessary monitoring and security needs.

The actors using the Web Services infrastructure need first to get the necessary credentials from the authentication and authorization services hosted on the external authentication and authorization server. This server hosts the services responsible for authenticating the actors and then assigns them required authorization operations. So, we have implemented our architecture in such a way that we can maintain the authentication externally by a third party entity. We have actually designed that external block with PERMIS [Ferraiolo et al. (2007)] as a PMI. We know that the PMI manages the authentication and authorization externally; here the PERMIS does the same with the additional concept of Role which motivates us to select it. PERMIS is a policy based authorization system which uses X.509 [Arsenault and Tumer (2000)] attributes to hold the role. It also needs fewer administrators which reduces cost. RBAC [Ferraiolo et al. (2007)] access decisions are taken based on the roles where roles are assigned to individual users of an organization. Access with predefined privileges is given only while a user activates a specified role.

1.1. Actor, roles, resources and their relationship

SI* modelling concept [Massacci et al. (2007)] consists of actors, roles, goals, processes, events and resources.

- Actor: An actor is an intentional entity. Actor plays a role in an organization or an application system. An actor is an agent either human or software.
- Goal: Goal is the strategic interest of an actor. An actor can have his own goal or goal imposed on him.
- Process or Task: Process or task is the sequence of actions to be executed in order to satisfy a goal.
- Resource: Resource is the physical or informational entity without intentionality. Resources are used by tasks for the actors or agents to accomplish a goal.

The actors, roles, resources, and goals are identified from the case study as shown in Table 1, Table 2, and Table 3. From the case study, we can model the relationship between different actors and roles which is shown in the Fig. 1. It consists of links with labels indicating type of relationships like Is Part Of, Plays and Supervise.

<table>
<thead>
<tr>
<th>Name of actors (agent)</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>Patient</td>
</tr>
<tr>
<td>Andrew</td>
<td>Family Doctor</td>
</tr>
<tr>
<td>Charlie</td>
<td>Substitute Doctor</td>
</tr>
<tr>
<td>Alison</td>
<td>Social worker</td>
</tr>
<tr>
<td>Dan</td>
<td>Pharmacist</td>
</tr>
<tr>
<td>Emergency Response Centre (ERC)</td>
<td>Software Framework</td>
</tr>
<tr>
<td>Medical Team</td>
<td>Emergency medical assistant</td>
</tr>
<tr>
<td>Mr. X</td>
<td>Electrician</td>
</tr>
</tbody>
</table>
Table 2. Resources for e-health scenario

<table>
<thead>
<tr>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-health terminal</td>
</tr>
<tr>
<td>Home Sensor Network</td>
</tr>
<tr>
<td>Sensor network central station</td>
</tr>
<tr>
<td>Location Information Centre</td>
</tr>
<tr>
<td>Smart T-Shirt</td>
</tr>
<tr>
<td>Pharmacy Computer</td>
</tr>
</tbody>
</table>

Table 3. Goals and wants of actors

<table>
<thead>
<tr>
<th>Actors (agents)</th>
<th>Goals</th>
<th>Wants (security concern)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient</td>
<td>Medical assistance</td>
<td>Privacy and security of medical data</td>
</tr>
<tr>
<td>Family Doctor</td>
<td>Assist patient</td>
<td>Handle patient data securely</td>
</tr>
<tr>
<td>Substitute Doctor</td>
<td>Assist patient</td>
<td>Handle patient data securely</td>
</tr>
<tr>
<td>Social worker</td>
<td>Provide social service</td>
<td>Reliable social service</td>
</tr>
<tr>
<td>Pharmacist</td>
<td>Give medicine</td>
<td>Check identity, credential and delegation</td>
</tr>
<tr>
<td>ERC</td>
<td>Medical assistance</td>
<td>Preserve privacy and security of medical data and ensure secure communication between all agents</td>
</tr>
<tr>
<td>Medical Team</td>
<td>Assist patient in emergency case</td>
<td>Handle patient data securely</td>
</tr>
<tr>
<td>Electrician</td>
<td>Repair Motion Sensor</td>
<td>To be an authentic electrician</td>
</tr>
</tbody>
</table>

For instance, Bob plays the role of patient, family doctor is part of medical team, ERC (Emergency Response Centre - a software agent) supervises medical team etc. The relationship diagram is drawn using SI* tool which also depicts a visible organizational structure of the case study scenario. The supervise relation gives the notion of dependability between supervisor and supervisee. Supervisee is the one who is being supervised with some assigned goal. For example in the Fig.1, we can see that, ERC supervises medical team to give real-time treatment to patient, social worker for the delivery of medicine to patient’s house and an electrician to repair motion sensor of the patient.

![Fig. 1. Relationship between agents and roles](image)

We are working on a multi-actor perspective where each actor interacts with other actors based on some relationship of trust and delegation. In a multi-actor settings of SI* approach, one actor can delegate his goal to be fulfilled by another actor, can trust other actors to execute certain goal or task.

2. Security Requirements

At first, let us have the general aspects of security concern in this smart item scenario. Smart item is a scenario with a constantly connected network environment where availability of all the actors and resources is a constraint. This type of networks may change their structure, availability, and communications abilities. The networks and systems building on top of them thus should show a certain level of redundancy and fault tolerance, be able to deal with uncertain or less reliable information, and adapt flexibility to changing environment. Reliability and validation of sensor data is an important issue.
Secure communication with business system and evaluation of access control policies for retrieving critical and confidential data is a major security requirement. The framework is expected to support the achievement of these advanced requirements in the following way:

1. Supporting the systematic design of appropriate security architecture
2. Providing support to meet restrictions imposed by device capabilities
3. Providing integration schemes addressing reliability and fault-tolerance aspects

Security requirements that are captured from the case study are:

- Patient medical record cannot be accessed or modified without authorized access
- Communications between ERC and Doctor must be appropriately protected to guarantee confidentiality and integrity of the data, authentication between involved parties. Context-aware authorization issue need to be considered
- While retrieving patient information (patient’s health information or location information) authentication and authorization need to be checked. Also confidentiality and integrity of retrieved data need to be ensured
- Communication between ERC and social worker should be kept secure
- Home sensor network data and data from patient’s motion sensor should not be used as illegal access by the Electrician
- Patient data should be stored by ERC in an anonymous way to maintain privacy
- A log file should be maintained to store all updates
- Non repudiation of signatures by involving parties should be ensured
- Electronic prescription by the doctor should follow a specific pattern. For instance, a reference to patient’s identity and expiration time, prescription should be digitally signed by doctor etc.
- A valid request for medicine need to be checked by the pharmacist in terms of identity, delegation and prescription and need to give correct medicine.

3. Security Policies

This section describes the security engineering design part that consists of identifying and designing an access control model to implement security policies.

Security policy is the link between control objectives and actual control processes to satisfy business goals and security goals [Massacci et al. (2007)]. Security policies are enforced by organizational policies or security mechanisms. An access control model for this case scenario must implement general security policy as well as some more specific policies. General policy like, patient medical record must be protected by authorized access; the smart item scenario is the close system where the lack of authorization implies no access. Some more specific policies are given in the section access control policy.

3.1. Access control model selection

Access control denotes the fact of determining whether an actor is able to perform an operation (read, write, execute, delete, search, etc.) on an object (medical record file, prescription etc.). An operation right on an object is called permission. An access control model defines how to organize the permissions of different actors/agents or users.

We found RBAC (Role Based Access Control) is the best suited model for Smart Item-Medical Emergency Response scenario because scenario demands authorized access for interaction of different actors/roles in the system. Access control decision is determined by the role that individual actors take place.

In the application scenario, authentication and authorization are handled by separate authentication and authorization server in a distributed way. So, we have selected PERMIS [Chadwick and Otenko (2003)] that implements RBAC based on PMI [Hui Xie et al. (2008)]. PMI is for the authorization where PKI [Arsenault and Tumer (2000)] is for authentication. PERMIS infrastructure [Chadwick and Otenko (2003)] is shown in Fig. 2. PMI uses Attribute Certificate (AC) [Mavridis et al. (2001)] for user authorization. PERMIS uses X.509 Attribute Certificates (ACs) to store authorization policies. Authorization policies specify which roles are assigned to which user identities, which roles are granted to perform which actions on which targets, etc. Privileges are assigned to roles, not users.

ISSN: 0975-5462
3.2. Access control policy

Access control policies define the users’ rights on objects, in order to enforce the security of an organization. In the RBAC model, policies define which permissions are granted to roles (permission-role assignment in Fig. 3). Thus users are granted permissions through role assignment (user-role assignment in Fig. 3).

The policies are what grants user access to various objects. Unless they are authorized to perform their responsibilities through one or more access control policies, users have no access to any of the objects. Some example policies from the case study are

- Doctor can assist patient by accessing patient’s PDA and send electronic prescription only if he has the proper right to access patient’s PDA.
- Patients give consent to the use of their records and have the right to be informed of their actual use.
- Patient’s home location information is sent from home central station only through authorized request.
- Patient can access, read their medical record but cannot modify, only doctor can modify patient’s record.

RBAC model supports a variety of security policies, in particular for instance least privileges and Separation of Duty (SoD). We will use least privilege principle for our case study which selectively assigns privileges or permissions to users based on role such that the user is given no more permission than is necessary to perform his job function.

<table>
<thead>
<tr>
<th>Patient medical record</th>
<th>Patient’s e-health terminal</th>
<th>Home network central station</th>
<th>e-prescription</th>
<th>Location Information Centre</th>
<th>Health Directory System</th>
<th>Invoice / Receipt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient</td>
<td>Read()</td>
<td>Access()</td>
<td>Read()</td>
<td>Sign()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Doctor</td>
<td>Modify()</td>
<td>Access()</td>
<td>Access()</td>
<td>Create()</td>
<td>Access()</td>
<td>Sign()</td>
</tr>
<tr>
<td>ERC</td>
<td>Read()</td>
<td>Store()</td>
<td>Access()</td>
<td>Browse()</td>
<td>Modify()</td>
<td></td>
</tr>
<tr>
<td>Social worker</td>
<td></td>
<td></td>
<td>Read()</td>
<td>Browse()</td>
<td>Sign()</td>
<td></td>
</tr>
<tr>
<td>Pharmacist</td>
<td></td>
<td></td>
<td>Read()</td>
<td>Browse()</td>
<td>Modify()</td>
<td>Sign()</td>
</tr>
<tr>
<td>Electrician</td>
<td></td>
<td></td>
<td>Access()</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.2.1. Access control matrix

The access control matrix provides a framework for analysing protection properties [Ferraiolo et al. (2007)]. The state of an access control system is defined by a triple \((S, O, A)\), where \(S\) is the set of subjects, \(O\) is the set of objects, and 
\(A\) is an access matrix, with rows corresponding to subjects and columns corresponding to objects. An entry \(A[s, o]\) is a set of rights. A description of access control matrix for this case study is given bellow

3.2.1.1. Subjects

Subjects are the principles that use objects. Subjects in our case study are

- Patient
- Doctor
- Emergency response centre (ERC)
- Social worker
- Pharmacist
- Electrician

Subject may be a user or software framework agent. First column of the matrix shows the subjects in Table 4.

3.2.1.2. Objects

Objects are the resources that need to be protected. Various objects identified from the case study are

- Patient medical record
- e-health terminal (running e-health application)
- Smart t-shirt
- Home network central station
- e-prescription
- Location Information Centre
- Health Directory System
- Invoice/Receipts

3.2.1.3. Operations

An entry in the access control matrix represents operations that are allowed by the subject to object. Operations that are identified from the case study are

- Read () - This operation is used to read the patient medical record, e-prescription or messages
- Modify () - This operation is used to modify patient medical record which is only allowed by doctor that means the classic update operation
- Create () - This operation is to create messages to send. Message created by patient for sending request, ERC to notify doctors, social worker, electrician and vice-versa.
- Send () - This operation allows sending e-prescription or messages and location information which requires authorization
- Store () - This operation is used to store patient medical record, e-prescriptions in to backend system
- Sign () - This operation is executed by the corresponding actor on the receipt or invoice
- Access () - This operation is used to allow access to patient’s PDA, home sensor network or central station
- Browse () - This is used to browse health directory system for finding medicine stock or nearest pharmacy location etc.

By the e-health application running on patient, doctor, social worker and pharmacist’s PDA each actor will activate respective access rights based on authorization assigned to respective roles which means will activate individual role. The application inherently supports read (), create () and send () messages as required for communication. For instance, patient will send a request message to ERC for delivery of medicine; then ERC will send a notify message to social worker in turn. Smart t-shirt will be accessed through patient’s e-health terminal so it has not shown in Table 4 of access control matrix. Electrician can access all the sensors installed in patient’s house as well as motion sensor but he need to be an authentic one.
3.3. Designing Access Control Model

From the case study we can see that, the smart item web service infrastructure contains the authentication/authorization server which we can use as PERMIS authentication / authorization infrastructure. As an authority HCC (Health Care Centre) will act as the Source of Authority (SoA) [Gollmann D. (2006)] for PERMIS. User request for access to any resources will be forwarded to this block for authorization. For instance, doctor will submit an access request through his PDA for patient medical record. Patient medical records are stored in ERC’s data warehouse server.

![Diagram of Access Control Model](image)

**Fig. 4: Doctor make request for patient’s medical record**

The reference monitor placed in ERC will ask PERMIS for the authorization of the request. Attribute Certificate (AC) repository contains all the policy attributes. After getting the authorization doctor will get a credential to access patient record and use this credential to read or update the record. We assume that user authentication is performed only once when each user makes a login request in their respective e-health application through PDA. When a doctor will make a request to access patient’s PDA to get current health data from patient’s motion sensor then the reference monitor in patient’s PDA will check doctor’s credential. The model of the access control for possible access is shown in the Fig. 4. This figure depicts the interaction of doctor and MERC (Monitoring Emergency Response Centre). Actually every request in the smart item scenario will pass through the MERC software framework. Doctor makes a request through his e-health terminal in a graphical user interface for patient’s medical record to MERC. RM (Reference Monitor) in lined [Gollmann D. (2006)] with MERC will check for authorization after authentication of doctor.

![Diagram of Access Control Model](image)

**Fig. 5: Doctor make request for patient’s current health data**

After authorization, doctor is allowed to access medical record. In Fig. 5 doctor’s request for patient’s motion sensor data has shown which will be forwarded from patient’s PDA to MERC’s Reference Monitor. This step is not shown in the Fig. 4, 5. After getting the authorization doctor’s PDA and patient’s PDA will work as a peer. The evaluation of authorization decision will require information from smart t-shirt which acts as one of the policy information point. Policy enforcement mechanisms are discussed in the next section.

We have used X-RBAC [Joshi et al. (2004)] language to build the system components. The using of web services like connecting to the E-health Application through the internet motivates us to select this language. X-RBAC works with three different kinds of sheets as RBAC concerns about User, Role, and Permission. Followings are the sheets:
4. Enforcement Mechanism

This section will describe the security engineering deployment part which deals with enforcement mechanism with respect to general system architecture. Security requirements that we have identified in security analysis part are modeled using PERMIS as a role based access control.
Smart item scenario is an example of web based application which consists of web service infrastructure and ambient intelligent infrastructure. We need to correlate our security architecture with this smart item infrastructure. The general system architecture is shown in Fig. 6.

![Fig. 6. General system architecture of smart item scenario](image)

The actors using the Web Services infrastructure in [Serenity consortium, MER technical report (2007)] first need to get the necessary credentials from the authentication and authorization services hosted on the authentication and authorization server. This server hosts the services responsible for authenticating the actors beside the Health Care Centre (HCC) [Lopez et al. (2007)] authority and then assigns them the required authorized operations. In case the actors are the medical teams, the doctors, or the patients and need to get the patients’ records inside the MERC’s data warehouse, then they should find the unique patient identifier from the Master Patient Index. Once the patient unique identifier is acquired, the MERC will log the actors for later auditing. Then controlled by the authorized operations, the actors query the data warehouse for patient’s data and/or search the service locator for the pharmacies, medical teams, social workers or doctors.

4.1. Policy enforcement model

Policy enforcement model in [Gollmann D. (2006)] consists of policy actors which are

- Policy Enforcement Point (PEP) : The (logical) system entity that performs access control, by making decision requests and enforcing authorization decisions
- Policy Decision Point (PDP) : The (logical) system entity that evaluates applicable policy and renders an authorization decision
- Policy Administration Point (PAP) : The (logical) system entity that creates a policy or policy set
- Policy Information Point (PIP) : The (logical) entity that acts as a source of attribute values – attributes describing subjects (users), resources, environments (contexts) used to decide whether a control process apply

In our application each reference monitor will act as PEP. PEP is the gateway to the resources. It receives users’ requests and asks PDP whether the requests are granted or not. Reference Monitor has been placed inside the MERC and Patient’s PDA in lined with the application software because we think that these are the two main important places to preserve the privacy of patient’s medical information.
All the requests and accesses will mediate through these RMs. An example implementation of policy by RM will be discussed later in this section.

The authentication/authorization server under MERC will act as PDP which uses PEMRIS authentication and authorization mechanism. PEP enforces access control to resources in accordance with the PDP’s decisions. PDP accepts inquiries from PEP and makes decisions whether the requests are granted or not according to the authorization policies. The Fig 7 shows the policy actors with respect to general system architecture. Here in the Fig. 7 two operations are shown. If doctor request patient’s medical record stored in MERC’s data warehouse then the request path follow A.1 to A.6. On the other hand, if doctor request to access patient’s current health data (blood pressure, body temperature etc.) from motion sensor then the request path follows 1 to 8.

We assume that LIC, smart t-shirt, home sensor network is the authentic source of information. Physical security of devices are also assumed to be present and we will not consider communication security for Bluetooth, Zigbee, Wi-Fi, Ethernet, GSM, UMTS / GPRS.

4.2. Reference monitor
Reference Monitor [Gollmann D. (2006)] [Vanoverbergh and Piessens (2008)] observes the execution of a program and halts the program if it is going to violate the security policy. It is used as an assurance framework. In our application mainly two RM has been placed for the enforcement of policy. We will use precise enforcement by suppression automata as security automata because smart item scenario is a real time application scenario where delay in information will lead to huge loss for patient. For writing the policy for this automaton ConSpec [Aktuga and Naliuka (2007)] can be used as policy specification language. The policies that we have define earlier for instance; patient medical data cannot be accessed without authorized access. The security automata in the Fig. 8 and ConSpec of this policy is given below
5. Conclusion

For designing the access control model of e-health system we have used PERMIS that uses RBAC, X.509 and PMI. The reason behind the use of PERMIS to implement authentication and authorization mechanism of smart item scenario was its’ distributed web based infrastructure. The policy model has been developed with the corresponding policy actors and some example interactions. The use of reference monitor has also introduced with an example policy implemented in ConSpec.

Smart Item Medical scenario requires 24 hours availability of actors. General medical scenario requires shifting duties of doctors. So, time constraint is another important fact on assigning doctors for different patient. It also applies for social worker. GTRBAC [Ferraiolo et al. (2007)] can be used to implement this time constraint in RBAC model which can be considered as the future work.

References