Abstract—Implementing a mobile application secure enough to store personal medical information and properly enforce security policies is an important task. There is currently no good implementation of document level security for patient information, and there is no way to properly enforce HIPAA medical privacy law. To try to resolve this, we have created a granular, role-based access control model for XML schemas and applied this model to digital medical records. Using Java Eclipse IDE, the Android SDK and JavaScript Object Notation (JSON) we adapted a mobile application called the Personal Health Assistant (PHA). This application interacts with Microsoft’s HealthVault through JSON API calls. We split the application into two separate implementations, and modified each with different objectives. One application is designed for the medical provider, and the other for the patient. In the future, both applications can be transitioned to use XML and Open mHealth standards.

I. INTRODUCTION

Implementing a mobile application secure enough to store personal medical information and properly enforce security policies is an important task. In this paper we will describe the background information, and discuss the workings of our application and how it will help to secure personal medical records.

In the field of digital medical records, security is of paramount concern [1]. Access control refers to features which allow us to control who can interact with elements of the health record. There are several different types of access control: In discretionary access control, the owner of an object decides the permissions other individuals have over that object. In mandatory access control, the system decides the permissions individuals have over an object. In role-based access control, permissions are assigned to roles, which are assigned to users [4].

[1] is a recently proposed access control framework for XML schemas which was designed with documents for healthcare in mind. The framework provides granular access control to XML documents from the design stage. The framework uses new UML XML Schema Class and XML Role Slice Diagrams to generate XACML security policies. These XACML policies, which correspond to the schema, are enforced on XML instances of that schema [1].

Personal health records are digital patient medical records curated and managed by the patient, as opposed to an organization that delivers care, such as a hospital. Personal health records (PHR) are often managed using online services which allow the patient to store, view and modify their records, while providing additional services. One such service is Microsoft HealthVault. Microsoft HealthVault is an online system designed to help users organize and share their Personal Health Record (PHR). The service aims to allow users to monitor their chronic conditions, track their progress towards goals, and share their medical information with their health providers at their discretion.

We worked with a mobile application for Android smartphones developed at the University of Connecticut called Personal Health Assistant (PHA). This application lets users manage their medical information such as recording daily living information and managing their medications. This application interacts with data pulled from Microsoft’s HealthVault through JSON API calls. Using Java Eclipse IDE, the Android SDK and JavaScript Object Notation (JSON) we adapted the Personal Health Assistant.

To apply the security framework to PHRs, we split the PHA into two separate implementations, and modified each with different objectives. One application is designed for patients looking to manage their PHR, while the other is designed to allow the patient to share their PHR with their providers. Together, the applications generate and enforce patient security policy.

The Personal Health Assistant consists of two applications, a patient oriented application and a provider oriented application. The patient oriented application allows users to manage their medical data that is stored in Microsoft HealthVault and gives them permission to decide which medical provider can view / edit information. The provider oriented application is a role-based access control application, therefore, depending on the role a medical provider has, that will determine whether he can view certain information or not.

In the future, both applications can be transitioned to fully use the XML and XACML framework and take advantage of Open mHealth standards.

II. BACKGROUND

A. Access Control

Access control refers to features which allow us to control who can interact with elements of the health record. From a security policy perspective, we are treating digital medical records as instances of an XML schema. We granted users access to those XML instances, and based on the role they were currently expressing established their ability to share, read, append or edit the individual sections of the instance, as
well as pass those privileges to other people. In the medical provider application we used a role-based access control approach, while the patient application presents the user with discretionary access control.

Role-based access control is an approach to access control which restricts access to an element based on the users “role,” or the job they are currently fulfilling. The application grants the user permissions which determine which methods can the user access or which parts of a method they can access.

B. Security Framework

A security framework has been proposed for the Personal Health Assistant mobile application (PHA). The proposed security framework provides access control capabilities to achieve customizable access to an XML documents elements by applying secure software engineering methodologies and defining new UML XML Schema Class and XML Role Slice Diagrams for schemas and permissions, generating XACML security policies, and enforcing security at the runtime level on XML instances to insure that correct and required patient data is securely delivered. XACML is used to define security policies for XML schemas and instances, providing a variety of access control capabilities to achieve customizable access to an XML documents elements.

C. HealthVault PHR

Microsoft HealthVault is an online system designed to help users organize and share their Personal Health Record (PHR). The service aims to allow users to monitor their chronic conditions, track their progress towards goals, and share their medical information with their health providers at their discretion. The user can share any part of their health record with anyone he chooses, with the hope that sharing their information with health care providers can help their providers better manage the health of their patients and monitor their progress with greater insight.

D. Personal Health Assistant (PHA)

The patient information present in a users HealthVault is used in by our Personal Health Assistant (PHA) application, through an intermediary server wrapper. The Personal Health Assistant application enables users to manage the medications they are taking (Figure 1), record any allergies they have, manage alerts that tell them when to take a medication, record how they are feeling at an specific time (Observations for Daily Living, Figure 2), and make that information available through reports.

In the medication entry screen shown in figure 1, the patient can enter all the pharmacological information such as the medication name and strength, along with their own personal prescription information such as the date they began taking the medication.

The ODL entry screen shown in figure 2 demonstrates the smiling face icons which help the user with self assessment. As the sliders move from 1 to 10, the faces on the right change expression.

![Fig. 1. Entry screen for medications.](image1)

![Fig. 2. Entry screen for a new ODL.](image2)
All this can be done from a handheld smartphone device running Android, an open source and fully customizable mobile platform. The application is currently developed for Android 2.3.3, but has run on older and newer versions of the platform throughout development. PHA uses a central server to generate reports and retrieves information from MS HealthVault through the server wrapper using a series of JSON API calls.

JSON (JavaScript Object Notation) is a data interchange format used to represent objects as plain text. Once serialized, the structured data can be transmitted over a network connection. The PHA must exchange session and request information with the server wrapper and receive personal health records in return. In order to do so, we use JSON calls to serialize and transmit data, and we receive a JSON response in return.

III. SECURITY FOR MOBILE HEALTH APPLICATIONS

A. Overall Architecture

The security for our application is enforced in three levels: the storage, the wrapper, and the mobile application. To begin, we split up the Personal Health Assistant application into two distinct branches: the security policy is created within the user PHA and stored in HealthVault alongside their PHR, where it is used by the wrapper to filter PHR before it is sent to the healthcare providers PHA.

The PHA uses Java models and server classes which generate and interpret JSON to connect to the server wrapper. The server wrapper then interacts with Microsoft HealthVault using an XML HealthVault API. When the provider application makes a request for patient information, the users security policy is enforced on the medical record before it is sent to the provider.

B. Patient Oriented

The patient oriented version of our Personal Health Assistant allows the user to control who can read and update their personal health records stored within Microsoft HealthVault. The security policy is generated on the Permissions tab of the patient application (Figure 4). The permissions tab contains, for each medical provider attached to the patient, a list of attributes from the patients health record and checkboxes representing the ability of that provider to read or write that attribute. The application then delivers the security policy to the HealthVault server, where it is used to filter information being sent to the provider application.

C. Provider Oriented

In the medical provider oriented version of the Personal Health Assistant application, the user holds one or more roles, and selects their current role after starting the application (Figure 6). The user is then presented with a list of patients they oversee dependent on the role they chose to fulfill when they entered the application (Figure 7). The medical provider can access their other patients by selecting an alternate role. This application has role based access control, and depending on the role that the medical provider chooses, he can view
and/or edit information (Figure 9) that is only related to that specific role.

To check a patient’s medical data, medical providers would log in, as shown in figure 5. They would then select their currently active role, as shown in figure 6. The provider is then presented with the information stored in our system (figure 7), and a list of their current patients (figure 8). Finally, selecting a patient name shows the patient medical record as shown in figure 9. Only the parts of the medical record which the patient has allowed that provider to see will appear, and only changes to sections which the patient has allowed the provider to edit will persist.

D. Achieving Security

The security policy generated by the patient in the PHA is sent to the wrapper and stored alongside their PHR. When a provider makes a request for the PHR of one of their patients, the security policy is used to filter the data, removing the information which should not be visible to the provider, before sending it to the provider application. The wrapper also filters any data generated by the provider before it is written to the patient’s HealthVault PHR.

IV. Future Trends

SMART and Open mHealth are two projects made to improve mobile health applications. The SMART project is focused on fostering and accelerating innovative health applications while Open mHealth is more focused on improving the performance of the mHealth application by sharing software
and ideas that enable people to do so. In the future we plan to combine this projects with the apps we created to improve the security.

V. CONCLUSION AND FUTURE WORK

Our work with these two applications represents considerable progress towards our goals of securing personal medical records in a precise and user-friendly way. We were able to establish a connection between MS HealthVault and the mobile applications using JSON calls and we also, add a permissions page so that patients can manage who can view or edit their medical data. Furthermore, we made changes to the interface in order to add the permissions for the patient oriented application and a roles page for the provider oriented application. We also added Role-Based Access Control to the provider oriented application and PHR Access Control to the patient oriented application.

In the future, we will adapt our applications to work with open standards such as XACML, SMART and Open mHealth to use the security we have developed to benefit the growing mobile health ecosystem.

ACKNOWLEDGEMENTS

Funding for the University of Connecticut Research Experience for Undergraduates in Trustable Computing Systems program was graciously provided by the National Science Foundation and by the United States Department of Defense. We gratefully acknowledge their support, without which our research in computer systems security and trustable computing systems would not be possible.

REFERENCES