A FHIR-Based PPS System Can Keep Your Genes Private

Shilin Zhu, BS¹, Tianlong Chen¹, Yi Wang¹, Luo Xiao¹, Gil Alterovitz, PhD²³⁴*

Abstract
When the Human Genome Project was launched in 1990, people believed we can know more about ourselves as human. It is no doubt that by decrypting the code of genetic information, we can get lots of the answers to illness and how to cure people from genetic information. However, the pioneers in the genetic field also foresaw the potential for misuse of human’s genetic information. The complete gene exposure may cause great social influences such as prejudice, injustice, discrimination and assault. Therefore, in order to prevent misuse of genetic information by strange people or parties, it is very important to protect the privacy according to people’s preferences, especially for patients. Electronic Health Record(EHR) has been widely spread in the world because it helps the doctor and patient record all the essential information such as genes by electronic devices, not by papers. In this paper we proposed a FHIR-based privacy server EHR system called Privacy Protected Server(PPS) system which contains a proxy server, a privacy server and the original data server. We also developed a web based software to let the patients submit their preferences and let the clients to request certain resources in privacy protecting mode. The client is assumed to be the one who wants to get access to the patient’s clinical EHR record and contact with proxy server only. The proxy server coordinates the original data server which contains all the patients’ information with the privacy server which follows the patients’ preferences about revealing the genetic information. The EHR in original data server will periodically update its privacy policy by requesting an update transmissions from privacy policy server. The complete system can help the doctors to protect the privacy of their patients according to preferences and makes the genetic information safe on the server as well.

Keywords
FHIR, Privacy, Genes

¹University of Science and Technology of China, Hefei, China
²Boston Children’s Hospital Computational Health Informatics Program, Boston, MA, USA
³Harvard Medical School, Boston, MA, USA
⁴Massachusetts Institute of Technology, Cambridge, MA, USA
*Corresponding author: gil@mit.edu

Contents

<table>
<thead>
<tr>
<th>Introduction</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Materials and Methods</td>
<td>2</td>
</tr>
<tr>
<td>1.1 FHIR Standard</td>
<td>2</td>
</tr>
<tr>
<td>1.2 PPS System Architecture</td>
<td>2</td>
</tr>
<tr>
<td>1.3 The Privacy Policy in PPS System</td>
<td>2</td>
</tr>
<tr>
<td>1.4 The Protection Mechanism</td>
<td>3</td>
</tr>
<tr>
<td>1.5 PPS Software Overview Web Application Interface Overview ● Server Development ● Two Basic Operations in PPS</td>
<td>3</td>
</tr>
<tr>
<td>2 Results</td>
<td>6</td>
</tr>
<tr>
<td>3 Discussion</td>
<td>7</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>7</td>
</tr>
</tbody>
</table>

Introduction
Medical services industry is an information-intensive area. And with the development of genetics and information technology, the combination of this two has largely changed how medical systems work — a system has been rapidly developed that can store patients’ genetic information, genetic variance for example, in EHR and allow doctors and researchers to retrieve these data for therapeutic or research purposes. Unfortunately not only advantages does it bring to us. For an easy example, suppose a person with some genetic disease is seeking for employment. Any company that gets his genetic data will definitely carefully consider or even directly reject his application for a potential expenditure on his medical costs in the future. Here comes prejudice and unfairness. In other cases, some patients have strong privacy-protection awareness and they don’t want there private information be accessed by any individuals or organizations. Genetic Information Nondis-
A FHIR-Based PPS System Can Keep Your Genes Private — 2/7

crimination Act of 2008, an act to prohibit discrimination on the basis of genetic information with respect to health insurance and employment, has been signed into law in 2008.(a reference will be nice here.) In this sense, the protection on patients’ genetic information is particularly important. However, although lots of work has been done to set up universal medical information exchange platforms, privacy protection systems remain a relatively unworked area. Therefore, this paper focuses on the research and the design of the Privacy Protected Server.

We sought to establish a medical information protection mechanism, based on the Substitutable Medical Applications and Reusable Technology(SMART) Health IT platform(www.smarthealthit.org), an open accessible application programming interface(API), running as an intervening filter between the original FHIR server and users(researchers, doctors, etc.). This system will smartly filter sensitive information so that they cannot be accessed by unauthorized users. The system also provide API for patients to customize their privacy protection policy that determines which part of their information should be masked and by whom they wish their information can be accessed. The representation of privacy policy is consistent with FHIR information standard and works for all types of clinical resources.

1. Materials and Methods

In this section we will introduce the FHIR-based PPS system and its working principle on protecting the privacy of patients. Then a brief overview of our developed web software will be given.

1.1 FHIR Standard

FHIR - Fast Healthcare Interoperability Resources is newly created by Health Level Seven International(HL7), a standards development organization. The resources defined in FHIR can be implemented into real world clinical situations and this standard can be widely used in mobile apps, EHR-based management, server communication and so on. The FHIR standard supports many types of resource description and transmission such as REpresentational State Transfer(RESTful) API, the Extensible Markup Language(XML), JavaScript Object Notation(JSON) and much more.

1.2 PPS System Architecture

The PPS System Architecture is shown in Figure 1, where the system mainly contains a proxy server, a data server and a privacy server. The proxy server is the one that directly deal with the applications built on the client. The original data server stores all the information of the patients, including their privacy policies. The privacy policy server opens an interface to the doctor or patient to let them establish and update the preferences. Then the privacy policy server will periodically update the policy database to the one in EHR at the original data server. Note that there is a firewall between the clients and the others because clients should not be given possible access to the original database and privacy policy according to the security consideration. In other words, only the proxy server can be authorized to visit the other two servers in our PPS system.

When a client wants to check the information of any patient, it will make a HTTP connection call to the proxy server by Application Programming Interface(API) on the client. Then the proxy server will forward the HTTP request to the data server and meanwhile make a policy inquiry request which contains the information of the client to the same server. The information of the client includes the client identification which is determined by the client’s IP address and the information of the patient requested from the client. The data server will transmit the information of the patient which is required by the client in its response message. Meanwhile, the data server will return the specific policy for that patient based on the client’s identity information and return the accordingly private policy. However, if the policy in EHR is outdated and has not been updated by the privacy policy server for some unknown reasons, the data server will soon request a policy update inquiry and the policy server will respond the update immediately besides the periodical policy update operations. If the responded policy includes nothing on the required information, the proxy server will directly forward the data to the client. Otherwise, the proxy server will first hide those information which is not allowed to be exposed according to the policy and then transmit the modified data to the client application.

1.3 The Privacy Policy in PPS System

Before the client can require access to the resources of the patient, the preferences of that patient(privacy policy) must be submitted to the privacy server by herself or her doctor as shown in Figure 2. Here we want the doctor to help the patient submit the form to the database since they are more professional. The patient needs to fill out a preferences form and give it to the doctor for submission. The data structure of the privacy policy is designed as well in Figure 2. Each policy must contain the identities of the patient and the doctor so they can be mapped whenever it is needed. The scope list contains the people to whom the patient wants to hide. The resource type restrains the type of information the patient is not willing to show. The resource ID is an unique identifier for this resource, and the content is the specific information

![Figure 1. PPS system architecture](image-url)
in JSON format of the preferences of the patients.

1.4 The Protection Mechanism
We have seen that the proxy server should integrate the policy into the returned EHR information from the data server. In our PPS system, the privacy protecting operations are achieved by proper comparison between the information of the policy and the EHR using layered JSON format, and then mask the corresponding information according to the specific policies. The JSON is well implemented in EHR at the data server based on FHIR platform. Therefore, it is reasonable and applicable to use JSON as an indicator for information comparison and masking. Figure 3 is an example to explain how JSON works with the privacy information protection.

1.5 PPS Software Overview
In this section, we will first introduce the front-end web based application overview, then we discuss the technical detail of each designed server mentioned above in the following subsections. Finally we will give the principle of two main operations: the policy submitting operation from the doctors and the information requesting operation from the clients.

1.5.1 Web Application Interface Overview
We design two interfaces, one for the outside clients and one for the doctor. The interface designed for clients is an open application to anyone who can get authorized by it. And the interface designed for the doctor is for the doctor to help the patients submit their preferences to privacy policy server. Both the interfaces are developed under the web technologies and the front-end view is shown in Figure 4 and Figure 5, each of which is the doctor’s view and search view interface, respectively.

After the client or the doctor is given the authorization to the EHR located at the data server and chooses the interested information she wants to obtain, a new interface will display the corresponding EHR information. If the login user is a doctor, the EHR will display all the information of the corresponding patients which is shown in Figure 4. Here we present all the information in EHR using a friendly user interface to make it applicable into the reality.

1.5.2 Server Development
In PPS system, there are 3 servers coordinated to achieve the privacy protections of EHR information. Now we first discuss the proxy server since it is the key server for coordinating
with the data server and the privacy policy server. The file architecture on the proxy server is listed in Figure 6. The Socket.py is used for packet dispatching. The respwrapper.py is an API for information filtering policy. The jsonparser.py is used for extracting layered JSON EHR data. Since the proxy server is the one which interacts with the client app directly, our designed proxy server can distinguish the three main types of messages sent from app and then redirect them to the correct locations, which are OAuth2.0 authentication as shown in Figure 8, and RESTful API interaction which is defined in FHIR platform and is used for data retrieving, and submissions such as privacy policies. Note that the authentication process includes two parts. The front-end client application must be authorized by the PPS system and the user must accepts the proper use of the EHR data.

The privacy policy server runs in a safe location where only the proxy server can visit, update and retrieve the data. The list of files designed for the privacy server is shown in Figure 8. The directory of COMMON is the database storage of the privacy policies. The api-privacy.py is the RESTful API for the privacy policy server and the example.py is the tutorial for the implementation of the server. We define three unit operations for the information on the privacy policy server, which are POST, PUT and DELETE, each of which is used for policy submission, update and deletion, respectively. The last one that is required for mention is the original data server which stores the EHR data of each patient. This EHR data is safely stored on this FHIR platform based data server and often ready for update.
1.5.3 Two Basic Operations in PPS

We first illustrate the corresponding operations on the doctor side, in which case the doctor submits the preferences for the patients. In the logging page of the doctor’s view as shown in Figure 4, the doctor needs to provide the patient’s identification as well as the relative disease which can help the PPS system to retrieve the observation and sequence information of the specific disease. After clicking the submit button, the PPS system will return an EHR interface for the doctor as shown in Figure 9. The EHR report page consists of four main parts which are PATIENT, OBSERVATION, SEQUENCE and SUBMIT. The PATIENT contains all the EHR data of the patient. The OBSERVATION contains the relative disease observations while the SEQUENCE presents the relative disease sequence, respectively. The SUBMIT is used for private profile submission. After logging into the local EHR interface designed for the doctor, the doctor can establish the privacy policy by simply clicking the HIDE button according to the preferences response from the patients. The doctor can also help the patient update the rules at any time if required in this interface. The technical detailed code of HIDE button is shown in Figure ??, which will store the privacy policy into the policy server after pressing submit button by the doctor. If the user is a client, the corresponding interface will only show the information after implementing the privacy policy filter as we mentioned just now.

Another basic operation is the Query process, which means the client requests information of interest using its application. The logging page designed for the outside user has been mentioned in Figure 5. In this page, the user must provide the identification and disease of the patient. We also design a list of items which allows the users to select the interested part by her will. After the client logs into the system, the EHR display will only include the information which is allowed for look by the corresponding patients as shown in Figure 16. It can be obviously seen that each observation contains a number of items which have SYSTEM, CODE and DISPLAY sections. The SYSTEM presents the website such as HGNC (Hugo Gene Nomenclature Committee) where people can find more explanations about this gene related information. The CODE is a specific searching index for that genetic information presented on the website provided by the SYSTEM. The DISPLAY presents the abbreviate and short explanation for that information. For illustration, an example is provided in Figure 12 which shows the retrieved information of ALK on the HGNC website pointed by EHR. It can be seen that the information on the symbol report contains the APPROVED NAME, the LOCUS TYPE, the CHROMOSOMAL LOCATION and the GENE FAMILY. The APPROVED NAME is the full genetic name approved by the HGNC. The LOCUS TYPE specifies the genetic class of that entry. The CHROMOSOMAL LOCATION indicates the cytogenetic location of the gene or specific region on the chromosome. The GENE FAMILY links to HGNC-curated gene family pages, where each link is a specific relevant gene family or group of the gene that has been assigned to.

Note that a marker will display to inform the protected EHR information according to the privacy policy of the corresponding patient. The technical details of such JSON information under the user interface is shown in Figure 13 in Figure 14, as two examples in EHR privacy protection in JSON format behind the front-end applications.
2. Results

For example, we choose an APP to use our API. This is an app for genetic report viewing. It gather data from a patient’s observation report for genetics and diagnostic report and locate genetic biology information in sequence data. The viewer will combine this result and generate the list of report with some specific sequence variation concerning with some certain diseases.

For each variation, it might involve these data: Patient, Sequence, Condition etc. And if we point app to our proxy sever, and put the resource of policy like this:

```json
{    'Policy': {        "variation": {            "observedAllele": "fhir_mask",            "referenceAllele": "fhir_mask",        },    'patient_id': 'b4ac73b3-8f3f-4717-99f0-41eb63694a1f',    'Policy_ResourceType': 'Sequence',    'Score': 'Clinician',    'ResourceIdentifier': '752124ec-dbaf-4128-991b-ae2a189c597c'    }
```

Then it will mask the result like the picture.
This is a simple genetics report viewer for a certain patient. It will show the related gene variation information as a list for report. If the app point to the proxy server, the source data from original server will be wrapped by the proxy sever and send back to client app and RESTful API does not make any change.

So you will see something (like allele in above picture) is hidden and replaced by the marker.

3. Discussion

This report viewer extracts part of information for a sequence variation, including its sample source, genetic coding name, the variation coding and its allele, the coordinate on chromosome, its status and the frequency of its appearance in all patient’s EMR. (0.9 means in all observation for genetics with all patient, for each 10 report, 9 report involve with this variation).

The information for demonstration of this sequence variation can help doctor save time in reading complex layer constructed json (JavaScript Object Notation) data and these information is clear enough to understand a patient’s genetic’s status. Nevertheless, a reference link might be useful if the doctor wants to see the correlation between these variations.

Acknowledgments

We would thank Heming, Xiaohan Chen and Miaofeng Liu for the database.