INSTALLATION GUIDE

SecGene – Secure i2b2 Plugin for Queries on Encrypted Genomes – v1

June 16, 2017

Gwangbae Choi and Jean Louis Raisaro
School of Computer and Communication Sciences
EPFL

Contents

About This Guide

This guide is designed to help i2b2 users with the installation of *SecGene* plugin. *SecGene* enables queries on encrypted genomic data without the need of decryption as it is based on a sophisticated homomorphic encryption scheme. Only the query end-result is decrypted and displayed in the Web-client. A detailed technical description of the *SecGene* can be found in the original paper [2].

1 Introduction

The deployment of *SecGene* does not follow the standard i2b2 plugin deployment. In addition to the Server and Web Client Cells, it includes a third component, i.e., the *Proxy*. We note that the Proxy has to be deployed on a third independent machine for security reasons. In particular, the Proxy should be installed on a machine that is physically located in a separate institution or administration than the one holding the i2b2 Server installation in order to avoid a collusion and the undesired reconstruction of the decryption key. Figure 1 illustrates the schema of *SecGene* deployment. It includes

- the **Genomic Database** responsible for storing the encrypted genomic data that can be installed on a PostgreSQL server,
- the **Secure Genomic Server Cell** responsible for the plugin business logic that can be installed on a machine running Linux,
- the **Web Client Plugin** responsible for generating genomic query,
- the **Importation tool** responsible for encrypting and loading the genomic variants from a VCF file into the Genomic Database that can be installed on a machine running Linux,
- the **Proxy** and its database responsible for the management of the cryptographic keys and for the decryption phase.

In the following sections we describe all the necessary steps needed to install these components.

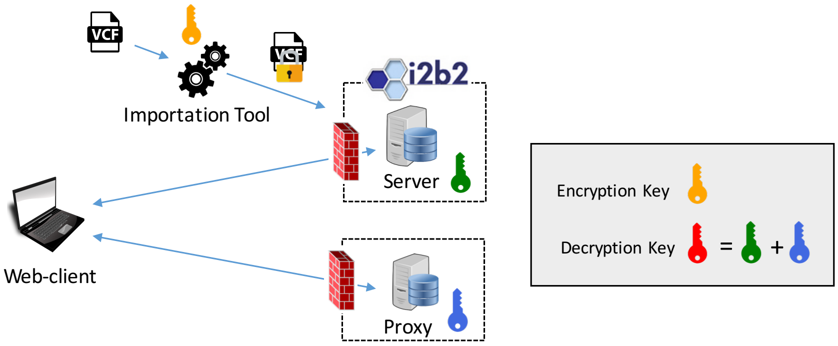


Figure 1: *SecGene* plugin deployment.

After cloning or downloading the *SecGene* Plugin repository from `https://c4science.ch/source/SecGene.git`, you will find the following six directories. We denote the downloaded repository as `$DOWNLOAD_DIR`.

Directory	Description
GEN	Source code of Secure Genomic Cell
PROXY	Source code of Proxy
webclient	Source code of Web Client Plugin
crypto	Source code of JNI crypto bridge
sql	SQL script for Cell and Proxy databases
importation-tool	Source code of the Importation Tool

Table 1: Source directories and descriptions

2 General Requirements

This section outlines the software and system requirements for the installation of the *SecGene* i2b2 plugin. These set of requirements is the same **for the GEN Cell, the PROXY and the Importation Tool**. Some of the requirements involve the installation of additional software from various third parties and when appropriate we have outlined the steps to install and configure it to work in your i2b2 environment.

- Operating System: Linux 64-bit
- Java Development Kit (JDK version 8 or later)
- Apache Ant (1.9.7 or later)

```
$ apt-get install ant
```

- WildFly/JBOSS 9.0.2 or later (to install it, you can follow the instructions of the i2b2 requirements at `http://community.i2b2.org/wiki/display/getstarted/Chapter+2.+Requirements`),
- Apache Axis2 1.7.1 or later (to install it, you can follow the instructions of the i2b2 requirements at `http://community.i2b2.org/wiki/display/getstarted/Chapter+2.+Requirements`),
- OpenJDK 7 or later and Database Management System (i.e., PostgreSQL or Oracle). We note that *SecGene* plugin supports two different installations for the i2b2 database: (i) single i2b2 database and one schema per cell in the Hive, or (ii) one distinct database per cell in the hive.
- Full installation of i2b2 source files `http://community.i2b2.org/wiki/display/getstarted/i2b2+Installation+Guide`) **(NOTE : this requirement is only necessary for the installation of the GEN Cell)**,
- C++11 compiler (g++-4.8 or later). Execute the following command on your command line:

```
$ apt-get install g++-4.8
```

- GNU Make 3.81 or later. Execute the following command on your command line:

```
$ apt-get install make
```

- NFLlib full installation. To install NFLlib, first clone or download the NFLlib source repository from `https://github.com/quarkslab/NFLlib`. Then, in the command line, change directory to the downloaded repository and execute the following commands:

  ```
  $ apt-get install cmake
  $ apt-get install libgmp3-dev
  $ apt-get install libmpfr-dev
  $ mkdir _build
  $ cd _build
  $ cmake .. -DCMAKE_INSTALL_PREFIX=$NFL_DIR/
  $ make
  $ make test
  $ make install
  ```

  The CMake option `-DCMAKE_INSTALL_PREFIX=$NFL_DIR/` specifies where the NFLlib library will be installed. If no option is specified, the library will be installed in the default location `/usr/local/`.

3 Secure Genomic Cell (GEN) Installation

The Secure Genomic Cell (GEN) manages the storage and querying of the homomorphically encrypted genomic data stored in the demodata tables.

3.1 Cryptographic Library Installation

This section describes the steps for installing the cryptographic library FV-NFLlib that implements the homomorphic encryption scheme by Fan and Vercauteren [1]. The FV-NFLlib library is based on the NFLlib C++ library that is part of the requirements of the plugin and needs to be installed beforehand.

1. Clone or download the FV-NFLlib repository `https://github.com/CryptoExperts/FV-NFLlib` to `$DOWNLOAD_DIR/FV-NFLlib`

2. Copy `$DOWNLOAD_DIR/FV-NFLlib/FV.hpp` to `$DOWNLOAD_DIR/crypto`,

3. Open the file `$DOWNLOAD_DIR/crypto/bridge.properties` and update the value of *NFL_DIR* with the one used in the installation of NFLlib. If you did not specify the option *-DCMAKE_INSTALL_PREFIX* during the installation of NFLlib, leave the default value */usr/local/*.

4. Change the working directory to `$DOWNLOAD_DIR/crypto` and run the following commands on the command line:

   ```
   $ make libFV_NFLlib.so
   $ mv libFV_NFLlib.so /usr/local/lib
   ```

5. (Optional) update shared library cache with the following command:

   ```
   $ ldconfig
   ```

3.2 Data Installation

This section outlines the steps for using the scripts that create the tables for the *SecGene* plugin. Note that the installation of the i2b2 database and the CRC cell must be completed prior to the beginning of *SecGene* database installation. In the followings, we provide the instructions for both PostgreSQL and Oracle Database Management Systems. **However, if possible, we recommend the use of PostgreSQL that was used in the development phase of this plugin.**

3.2.1 PostgreSQL

1. Change the working directory to `$DOWNLOAD_DIR/sql/i2b2/psql/`,
2. Open *grant.sql* and update line 3 (`grant usage on schema public to i2b2gen`) with the schema containing the tables of the CRC Cell in your i2b2 installation (default: `public`),
3. Log in i2b2 PostgreSQL database as administrator (default):

```
$ sudo -u postgres psql
```

4. Once logged in, create the i2b2gen schema into the i2b2demodata database (if the i2b2 database is installed following the one-database-per-cell logic) or into the main database (if the i2b2 database is i2b2 database is installed following the one-schema-per-cell logic):

```
=> \c i2b2demodata
=> CREATE SCHEMA i2b2gen
```

5. Create new user i2b2gen, execute the *grant.sql* script, and log out:

```
=> CREATE USER i2b2gen WITH PASSWORD 'i2b2gen'
=> \i grant.sql
=> \q
```

6. Log in as i2b2gen user:

```
$ psql -U i2b2gen -d i2b2demodata -W
```

7. Execute the *table.sql* script and log out:

```
=> \i table.sql
=> \q
```

3.2.2 Oracle

1. Change the working directory to `$DOWNLOAD_DIR/sql/i2b2/oracle/`,
2. Open *grant.sql* and update line 3 (`grant usage on schema public to i2b2gen`) with the schema containing the tables of the CRC Cell in your i2b2 installation (default: `public`),
3. Log in i2b2 Oracle database as administrator,

4. Create the i2b2gen schema into the i2b2demodata database (if the i2b2 database is installed following the one-database-per-cell logic) or into the main database (if the i2b2 database is i2b2 database is installed following the one-schema-per-cell logic),
5. Create new user i2b2gen,
6. Allocate quota to i2b2gen,
7. Execute the *grant.sql* script, and log out,
8. Log in as i2b2gen, execute the *table.sql* script, and log out.

3.3 Cell Configuration

This section describes the steps required to update the files in the `$DOWNLOAD_DIR/GEN/` directory in order to reflect the appropriate configuration at your site.

3.3.1 Configure the build properties

During this step you need to set the *jboss.home* and *axis2.war.name* properties.

1. Change the working directory to `$DOWNLOAD_DIR/GEN/`,
2. Open the file `build.properties` and update the value of *jboss.home* to `$YOUR_JBOSS_HOME_DIR` and of *axis2.war.name* to `$YOUR_i2b2_WAR_NAME`,

3.3.2 Configure the GEN application directory properties

During this step you need to specify a location for the application properties directory.

1. Change the working directory to `$DOWNLOAD_DIR/GEN/etc/spring/`,
2. Open the file `gen_application_directory.properties` and update the prefix of *edu.harvard.2b2.gen.applicationdir* to `$YOUR_JBOSS_HOME_DIR` (*standalone/configuration/genapp* does not need to be modified).

3.3.3 Configure the GEN properties

The *SecGene* cell needs these properties to deploy correctly and to set required parameters.

1. Open the file `gen.properties`,
2. Update the value of *gen.database.type* to either *POSTGRESQL* or *ORACLE*,
3. Update the IP address and the port of the Project Management (PM) Cell (*gen.ws.pm.url*),
   Default: *http://127.0.0.1:9090/i2b2/services/PMService/getServices*,
4. Set the database information for the CRC cell (*gen.ds.crc.schemaname*) and for the GEN cell (*gen.ds.gen.schemaname*),
5. Set the value of *gen.core.query.threads* to 8
6. Set the value of *gen.data.genotype.phase* to 0

3.3.4 Configure Data Source

The GEN cell needs to communicate with your i2b2 database and in order to do this it needs to know where the tables of **i2b2gen** reside. These tables were created during the **Data Installation** step.

1. Change the working directory to `$DOWNLOAD_DIR/GEN/etc/jboss/`,
2. Open the file *gen-ds.xml* and update <*connection-url*>, <*user-name*> and <*password*> as follows:

PostgreSQL:

```
<datasource jta="false" jndi-name="java:/GenDS"
        pool-name="GenDS" enabled="true" use-ccm="false">
        <connection-url>jdbc:postgresql://localhost:5432/i2b2demodata</connection-url>
        <driver-class>org.postgresql.Driver</driver-class>
        <driver>postgresql-9.2-1002.jdbc4.jar</driver>
        <security>
                <user-name>i2b2gen</user-name>
                <password>i2b2gen</password>
        </security>
        <validation>
                <validate-on-match>false</validate-on-match>
                <background-validation>false</background-validation>
        </validation>
        <statement>
                <share-prepared-statements>false</share-prepared-statements>
        </statement>
</datasource>
```

Oracle:

```
<datasource jta="false" jndi-name="java:/GenDS"
        pool-name="GenDS" enabled="true" use-ccm="false">
        <connection-url>jdbc:oracle:thin:@localhost:1521:xe</connection-url>
        <driver-class>oracle.jdbc.OracleDriver</driver-class>
        <driver>ojdbc6.jar</driver>
        <security>
                <user-name>i2b2gen</user-name>
                <password>i2b2gen</password>
        </security>
        <validation>
                <validate-on-match>false</validate-on-match>
                <background-validation>false</background-validation>
        </validation>
        <statement>
            <share−prepared−statements>false</share−prepared−statements>
        </statement>
    </datasource>
```

3.4 Deployment

The following steps will walk you through the deployment process. We denote as `$I2B2_SRC_DIR/` the source repository of the standard i2b2 installation.

1. Copy the repository `$I2B2_SRC_DIR/edu.harvard.2i2b.server-common/` to `$DOWNLOAD_DIR/`
2. Copy the repository `$I2B2_SRC_DIR/edu.harvard.2i2b.xml/` to `$DOWNLOAD_DIR/`
3. Change the working directory to `$DOWNLOAD_DIR/edu.harvard.2i2b.server-common/`,
4. Run ant script with the following command to build the server common files:

   ```
   $ ant clean dist
   ```

5. Change the working directory to `$DOWNLOAD_DIR/GEN/`
6. Run ant script with the following command to build the GEN cell:

   ```
   $ ant clean dist deploy
   ```

3.5 WildFly/JBoss Configuration

In this section, we describe how to update the configuration of WildFly/JBoss in order to work with our GEN server plugin.

1. Change the working directory to `$JBOSS_HOME/standalone/configuration/` where `$JBOSS_HOME` is the directory with the installation of WildFly/JBoss,
2. Open the file `standalone.xml` and update the public interface as follows (*jboss.bind.address* should be 0.0.0.0 or the server external IP):

   ```
   <interface name="public">
       <inet−address value="${jboss.bind.address:0.0.0.0}"/>
   </interface>
   ```

3. Change the working directory to `$JBOSS_HOME/bin/`
4. Open `standalone.conf` and update JAVA_OPTS by updating *-Xms64m* and *-Xmx4096m* and by adding *-Djava.library.path=/usr/local/lib*, where in *-Djava.library.path* you specify the path of `libFV_NFLlib.so`.

4 Web Client Plugin Installation

This section will guide you through the installation of the SecGene Web Client plugin. We assume the i2b2 Web Client is already installed in the following repository `/var/www/html/webclient` that we denote as `$WEB-CLIENT_HOME`.

1. Move the `$DOWNLOAD_DIR/webclient/GEN/` repository to `$WEB-CLIENT_HOME/js-i2b2/cells/plugins/standard/` and grant rights with *chmod*,
2. Change the working directory to `$WEB-CLIENT_HOME/js-i2b2/`,
3. Open the file `i2b2_loader.js` and add the following lines to *i2b2.hive.tempCellsList*

```
{ code: "GEN",
    forceLoading: true,
    forceConfigMsg: { params: [] },
    roles: [ "DATA_LDS", "DATA_DEID", "DATA_PROT" ],
    forceDir: "cells/plugins/standard"
}
```

- *roles:* required i2b2 access level for the access to the plugin
- *forceDir:* path to `./GEN/` repository containing the Web Client plugin.

5 Proxy Installation

The Proxy is responsible for the management and storage of the decryption key shares used in the decryption of the final query result. For testing purposes, the proxy can be installed on the same machine as the i2b2 server, but we **strongly recommend** to use a physically separate and independent machine in case of a more "production-like" environment. Clone or download the *SecGene* Plugin repository as mentioned in Sec. 1

In the case of deployment on a separate machine, you should make sure to install all the required software specified in Sec. 2. Note that the proxy needs both WildFly/JBoss, Apache Axis2, a Database Management System such as PostgreSQL and NFLlib to run correctly.

5.1 Cryptographic Library Installation

This section describes how to build the cryptographic library for the polynomial multiplication used in the decryption algorithm.

1. Change the working directory to `$DOWNLOAD_DIR/crypto` and run the following commands on the command line:

```
$ make libNFL_JAVA.so
$ mv libNFL_JAVA.so /usr/local/lib
```

2. (Optional) update shared library cache with the following command:

```
$ ldconfig
```

5.2 Data Installation

This section outlines the steps for using the scripts that create the tables for the proxy component of the *SecGene* plugin.

5.2.1 PostgreSQL

1. Change the working directory to `$DOWNLOAD_DIR/sql/proxy/psql/`,

2. Log in i2b2 PostgreSQL database as administrator (default):

```
$ sudo –u postgres psql
```

3. Once logged in, create new database i2b2proxy:

```
=> CREATE DATABASE i2b2proxy
```

4. Create new user i2b2proxy, execute the *grant.sql* script, and log out:

```
=> CREATE USER i2b2gen WITH PASSWORD 'i2b2proxy'
=> \i grant.sql
=> \q
```

5. Log in as i2b2proxy user:

```
$ psql –U i2b2proxy –d i2b2proxy –W
```

6. Execute the *table.sql* script and log out:

```
=> \i table.sql
=> \q
```

5.2.2 Oracle

1. Change the working directory to `$DOWNLOAD_DIR/sql/proxy/oracle/`,

2. Once logged in, create a new database i2b2proxy,

3. Create new user i2b2proxy,

4. Allocate quota to i2b2proxy,

5. Execute the *grant.sql* script, and log out,

6. Log in as i2b2proxy, execute the *table.sql* script, and log out.

5.3 Proxy Configuration

This section describes the steps required to update the files in the `$DOWNLOAD_DIR/PROXY/` in order to reflect to appropriate configuration at your site.

5.3.1 Configure the build properties

During this step, you need to set the *jboss.home* property.

1. Change the working directory to `$DOWNLOAD_DIR/PROXY/`
2. Open the file `build.properties` and update the value of *jboss.home* to `$YOUR_JBOSS_HOME_DIR`.

5.3.2 Configure the PROXY application directory properties

During this step, you need to specify a location for the application properties directory.

1. Change the working directory to `$DOWNLOAD_DIR/PROXY/etc/spring/`,
2. Open the file `proxy_application_directory.properties` and update the prefix of *edu.harvard.i2b2.gen.proxy.appliactiondir* to `$YOUR_JBOSS_HOME_DIR`.

The file `$DOWNLOAD_DIR/PROXY/etc/spring/proxy.properties` does not need to be modified if the default schema (*public*) is used.

5.3.3 Configure Data Source

The PROXY needs to communicate with your i2b2proxy database and in order to do this, it needs to know where the table of i2b2proxy reside. These tables were created during the **Data Installation** step.

1. Change the working directory to `$DOWNLOAD_DIR/PROXY/etc/jboss/`,
2. Open the file *proxy-ds.xml* and update <*connection-url*>, <*user-name*> and <*password*> as follows:

PostgreSQL:

```
    <datasource jta="false" jndi−name="java:/ProxyDS"
    pool−name="ProxyDS" enabled="true" use−ccm="false">
        <connection−url>jdbc:postgresql://localhost:5432/postgres</connection−url>
        <driver−class>org.postgresql.Driver</driver−class>
        <driver>postgresql−9.2−1002.jdbc4.jar</driver>
        <security>
            <user−name>i2b2proxy</user−name>
            <password>i2b2proxy</password>
        </security>
        <validation>
            <validate−on−match>false</validate−on−match>
            <background−validation>false</background−validation>
        </validation>
        <statement>
            <share−prepared−statements>false</share−prepared−statements>
        </statement>
</datasource>
```

Oracle:

```
    <datasource jta="false" jndi−name="java:/ProxyDS"
    pool−name="ProxyDS" enabled="true" use−ccm="false">
        <connection−url>jdbc:oracle:thin:@localhost:1521:xe</connection−url>
        <driver−class>oracle.jdbc.OracleDriver</driver−class>
        <driver>ojdbc6.jar</driver>
        <security>
            <user−name>i2b2proxy</user−name>
            <password>i2b2proxy</password>
        </security>
        <validation>
            <validate−on−match>false</validate−on−match>
            <background−validation>false</background−validation>
        </validation>
        <statement>
            <share−prepared−statements>false</share−prepared−statements>
        </statement>
</datasource>
```

5.4 Deployment

The following steps will walk you through the deployment process. We denote as `$I2B2_SRC_DIR/` the source repository of the standard i2b2 installation. Because the PROXY uses the same library as i2b2, you need to install the i2b2 Server Common files first.

1. Copy the repository `$I2B2_SRC_DIR/edu.harvard.2i2b.server-common/` to `$DOWNLOAD_DIR/`
2. Copy the repository `$I2B2_SRC_DIR/edu.harvard.2i2b.xml/` to `$DOWNLOAD_DIR/`
3. Change the working directory to `$DOWNLOAD_DIR/edu.harvard.2i2b.server-common/`,
4. Run ant script with the following command to build the server common files:

   ```
   $ ant clean dist
   ```

5. Change the working directory to `$DOWNLOAD_DIR/PROXY/`
6. Run ant script with the following command to build the PROXY:

   ```
   $ ant clean dist deploy
   ```

5.5 WildFly/JBoss Configuration

In this section, we describe how to update the configuration of WildFly/JBoss in order to work with the PROXY.

1. Change the working directory to `$JBOSS_HOME/standalone/configuration/` where `$JBOSS_HOME` is the directory with the installation of WildFly/JBoss,
2. Open the file `standalone.xml` and update the public interface as follows (*jboss.bind.address* should be 0.0.0.0 or the PROXY's external IP):

   ```
   <interface name="public">
           <inet−address value="${jboss.bind.address:0.0.0.0}"/>
   </interface>
   ```

3. Find the <*subsystem*> whose XML namesapece is *urn:jboss:domain:undertow:2.0*, add new response headers as below and update "$IP_ADDRESS_I2B2_SERVER:PORT" with the IP address and port of your i2b2 server on which you have installed the GEN plugin (NOTE: the IP address and port exact match is supported).

Listing 1: standalone.xml

```
<subsystem xmlns="urn:jboss:domain:undertow:2.0">
        <buffer−cache name="default"/>
        <server name="default−server">
                <http−listener name="default" socket−binding="http" redirect−socket="https"/>
                <host name="default−host" alias="localhost">
                        <location name="/" handler="welcome−content"/>
                        < filter −ref name="server−header"/>
                        < filter −ref name="x−powered−by−header"/>
                        <filter-ref name="acao"/>
                        <filter-ref name="acah"/>
                </host>
        </server>
        <servlet−container name="default">
                <jsp−config/>
                <websockets/>
        </servlet−container>
        <handlers>
                <file  name="welcome−content" path="${jboss.home.dir}/welcome−content"/>
        </handlers>
        < filters >
                <response−header name="server−header" header−name="Server"
                header−value="WildFly/9"/>
                <response−header name="x−powered−by−header" header−name="X−Powered−By"
                header−value="Undertow/1"/>
                <response-header name="acao" header-name="Access-Control-Allow-Origin" header-
value="$IP_ADDRESS_I2B2_SERVER"/>
                <response-header name="acah" header-name="Access-Control-Allow-Headers" header-
value="Origin, Content-Type"/>
        </ filters >
</subsystem>
```

4. Change the working directory to `$JBOSS_HOME/bin/`

5. Open `standalone.conf` and update JAVA_OPTS by updating *-Xms64m* and *-Xmx4096m* and by adding *-Djava.library.path=/usr/local/lib*, where in *-Djava.library.path* you specify the path of `libNFL_JAVA.so`..

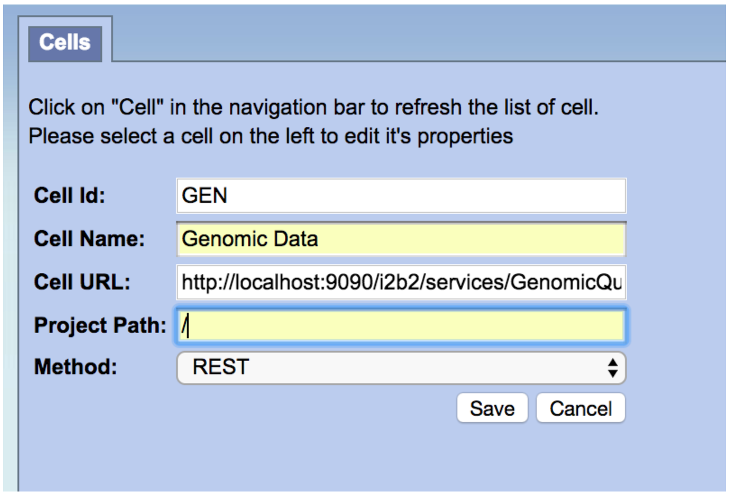
6 I2B2 ADMINISTRATION CONFIGURATION

This section describes the steps for configuring the i2b2 admin tool in order to register the new *SecGene* plugin.

1. Open your Web Browser and log into the i2b2 Administration Tool (`http://IP_ADDRESS_I2B2_SERVER/admin` – default: `http://localhost:80/admin`),

2. Click on **Manage Cells** in the **PM Navigation** menu on the left,

3. Click on the **Add New Cell** button and enter the following information:



Click on "Cell" in the navigation bar to refresh the list of cell.
Please select a cell on the left to edit it's properties

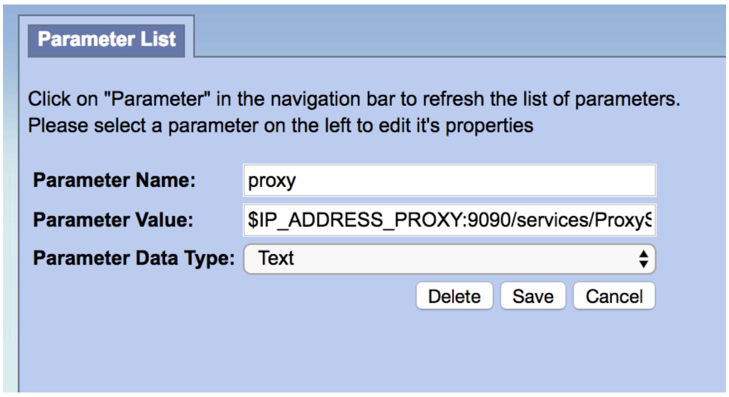
Cell Id:	GEN
Cell Name:	Genomic Data
Cell URL:	http://localhost:9090/i2b2/services/GenomicQu
Project Path:	/
Method:	REST

4. Click on the **Save** button,

5. Click again on **Manage Cells** in the **PM Navigation** menu on the left,

6. Click on the **Genomic Data** Cell and then on **Params**,

7. Click on **Add New Parameter** button in the **Parameter List** tab and enter the following information:



Click on "Parameter" in the navigation bar to refresh the list of parameters.
Please select a parameter on the left to edit it's properties

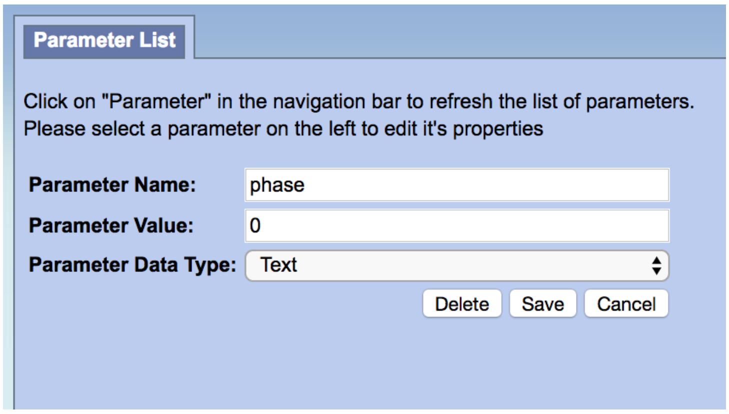
Parameter Name:	proxy
Parameter Value:	$IP_ADDRESS_PROXY:9090/services/ProxyS
Parameter Data Type:	Text

8. Substitute "`$IP_ADDRESS_PROXY`" in `$IP_ADDRESS_PROXY:9090/i2b2/services/ProxyService/` with the IP address of your PROXY.

9. Click on **Save** button,

10. Click again on the **Add New Parameter** button and enter the following information:

11. Click on the **Save** button.



Click on "Parameter" in the navigation bar to refresh the list of parameters.
Please select a parameter on the left to edit it's properties

Parameter Name: phase
Parameter Value: 0
Parameter Data Type: Text

7 Importation Tool

The Importation Tool allows the i2b2 administrator to encrypt genetic variants in a VCF file with a new cryptographic key set or with an existing one and to grant access to genomic data stored in the GEN Cell to new users. In this section, we describe the steps for its installation and use. So, first thing to do is to clone or download the *SecGene* Plugin repository as mentioned in Sec. 1

7.1 Installation

In this section we go through the steps necessary for the Importation Tool installation.

7.1.1 Data Installation

In the followings, we outline the steps for using the scripts that create the tables used by the Importation Tool. (NOTE : the Importation Tool does not necessarily need to be installed on the same machine as the GEN Cell)

1. Change the working directory to `$DOWNLOAD_DIR/sql/importation-tool/`,

2. Log in i2b2 PostgreSQL database as administrator (default):

```
$ sudo -u postgres psql
```

3. Once logged in, create new database i2b2importationtool:

```
=> CREATE DATABASE i2b2importationtool
```

4. Create new user i2b2importationtool, execute the *grant.sql* script, and log out:

```
=> CREATE USER i2b2importationtool WITH PASSWORD 'i2b2importationtool'
=> \i grant.sql
=> \q
```

5. Log in as i2b2importationtool user:

```
$ psql -U i2b2importationtool -d i2b2importationtool -W
```

6. Execute the *table.sql* script and log out:

```
=> \i table.sql
=> \q
```

7.1.2 Deployment

1. Clone or download the FV-NFLlib repository `https://github.com/CryptoExperts/FV-NFLlib` to `$DOWNLOAD_DIR/FV-NFLlib`
2. Copy `$DOWNLOAD_DIR/FV-NFLlib/FV.hpp` to `$DOWNLOAD_DIR/importation-tool`,
3. Copy `$DOWNLOAD_DIR/crypto/edu_harvard_i2b2_gen_util_CryptosystemBridge.*` to `$DOWNLOAD_DIR/importation-tool`,
4.
5. Copy `$DOWNLOAD_DIR/crypto/makefile` to `$DOWNLOAD_DIR/importation-tool`,
6. Change the working directory to `$DOWNLOAD_DIR/importation-tool` and run the following command on the command line:

```
$ make ImportationTool
```

7.2 How to Use it

In this section we describe how to use the Importation Tool to (i) encrypt variants in a VCF file and (ii) grant access to the encrypted genetic data to a new user.

7.2.1 Data ETL

To encrypt genetic variants in a VCF file and load them into the GEN database you should go through the following steps:

1. Change the working directory to `$DOWNLOAD_DIR/importation-tool` and run the following command on the command line:

```
$ ./ImportationTool -f -i your-vcf-file.vcf
```

2. This command generate the following set of SQL files:
   - *parameter.sql*: this file contains the parameters of the crypto system used to encrypt the data,
   - *individuals.sql*: this file contains the list of individuals in the input VCF file,
   - *variants.sql*: this file contains the list of genetic variants in the input VCF file,
   - *genotype.sql*: this file contains the list of encrypted genotypes for all individuals in the input VCF file.
3. Log in as i2b2gen user into the i2b2gen schem:

```
$ psql -U i2b2gen -d i2b2demodata -W
```

4. Execute the SQL scripts and log out:

```
=> \i parameter.sql
=> \i individuals.sql
=> \i variants.sql
=> \i genotypes.sql
=> \q
```

7.2.2 Granting access

To grant access to the encrypted genomic data stored in the GEN Cell, you can go through the following steps:

1. Create new user access level in the i2b2gen database and log out:

   ```
   $ psql –U i2b2gen –d i2b2demodata –W
   $ INSERT INTO gen\_user\_roles VALUES(1,'any_name',0)
   $ \q
   ```

2. Log into the i2b2 Administration tool (`http://IP_ADDRESS_I2B2_SERVER/admin` – default: `http://localhost:80/admin`)
3. Click on **Manage Users** in the **PM Navigation** menu and then on the **Add New User** button,
4. Fill in the required information,
5. Click on **Manage Projects** in the **PM Navigation** menu,
6. Then click on the desired project and on *Users*,
7. Click on the **Add User to Project** button and enter the user's username,
8. Click on the **Add User to Project** button the on the user's username in the **PM Navigation** menu,
9. Click on **Roles** and assign the "Limited Data Set" role as maximum role,
10. Change the working directory to `$DOWNLOAD_DIR/importation-tool` and run the following command on the command line:

    ```
    $ ./ImportationTool –u keyset the–assigned–username 1
    ```

    where the last parameter represents the access level previously inserted in the database,
11. Execute the i2b2server.sql script on the i2b2gen database with i2b2gen user in order to insert the decryption key shares of the new added user on the i2b2 server,
12. Execute the i2b2server.sql script on the i2b2proxy database with i2b2proxy user in order to insert the other decryption key shares of the new added user also on the PROXY.

8 Troubleshooting

If you have successfully followed all these steps, you should be able to log in to the Web Client and use our *SecGene* i2b2 plugin smoothly. For troubleshooting or questions, please contact us at gwangbae.choi@epfl.ch or jean.raisaro@epfl.ch.

References

[1] Fan, J., and Vercauteren, F. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive 2012* (2012), 144.

[2] Raisaro, J. L., Choi, G., Pradervand, S., Colsenet, R., Jacquemont, N., Rosat, N., Mooser, V., and Hubaux, J.-P. Protecting Privacy and Security of Genomic Data in i2b2. *Accepted for publication in IEEE Transactions of Computational Biology and Bioinformatics* (2017).