API usage instruction

xc9pd

October 2021

This is the backend of the central authority. The frontend is in another project:

https://github.com/cxfcdcpu/secure_data_sharing_dashboard

1 setup environment

The backend is developed in Ubuntu but could also be able to run in Windows.

1.1 Ubuntu environment setup:

1.1.1 install database:

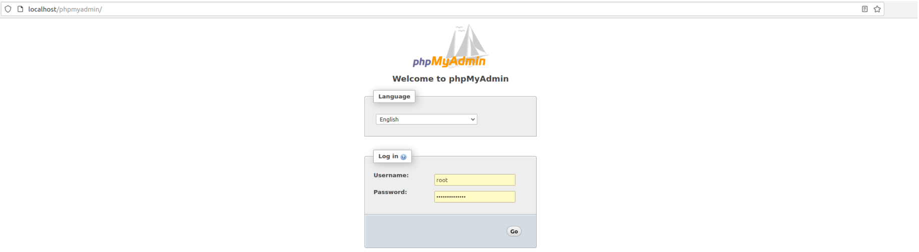
We use MySQL database. To use MySQL, install LAMP service using the following instruction:

https://ubuntu.com/server/docs/lamp-applications

1.1.2 Setup database:

Use phpMyAdmin to setup the database and the table as following:

Login phpMyAdmin:

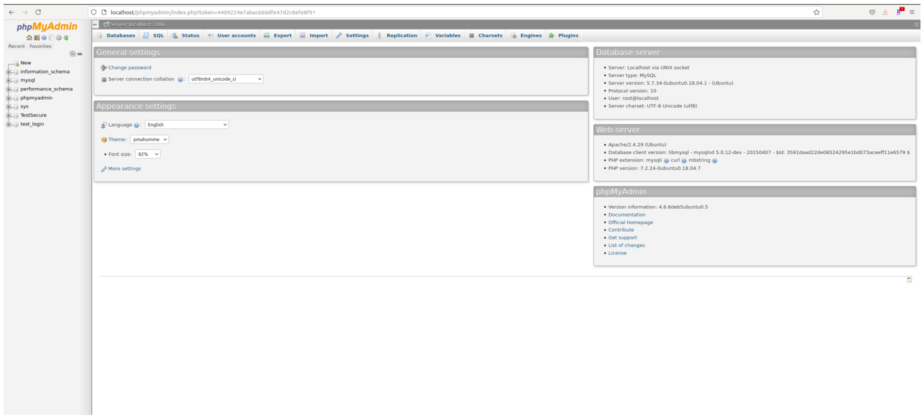


localhost/phpmyadmin/

Username: root

Password: ••••••••••••••

Find import on top:



localhost/phpmyadmin/index.php?token=4409224e7abac666dfe47d2c8efe8f91

General settings

- Change password
- Server connection collation: utf8mb4_unicode_ci

Appearance settings

- Language: English
- Theme: pmahomme
- Font size: 82%
- More settings

Database server

- Server: Localhost via UNIX socket
- Server type: MySQL
- Server version: 5.7.34-0ubuntu0.18.04.1 - (Ubuntu)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

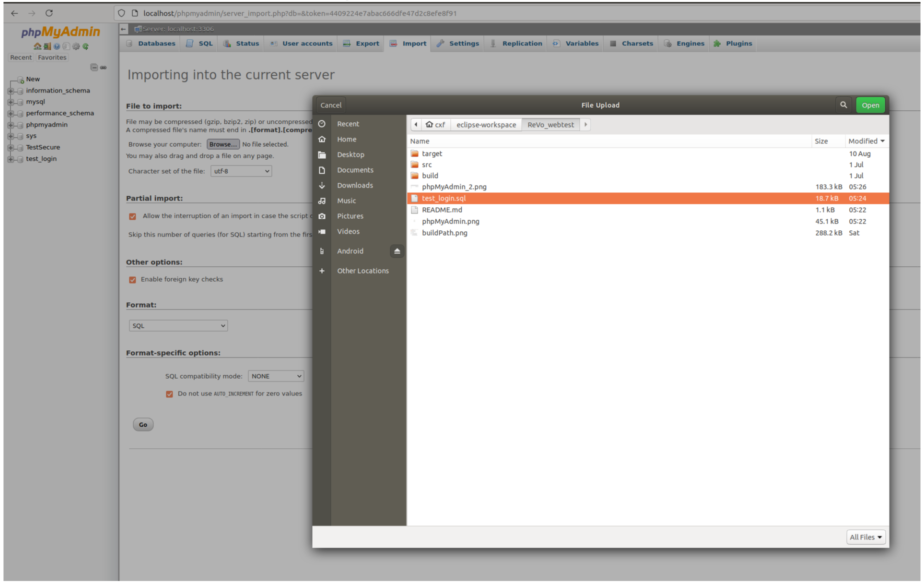
Web server

- Apache/2.4.29 (Ubuntu)
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - $Id: 3591daad22de08524295e1bd073aceeff11e6579 $
- PHP extension: mysqli curl mbstring
- PHP version: 7.2.24-0ubuntu0.18.04.7

phpMyAdmin

- Version information: 4.6.6deb5ubuntu0.5
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

Import database from test_login.sql:



localhost/phpmyadmin/server_import.php?db=&token=4409224e7abac666dfe47d2c8efe8f91

Importing into the current server

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed
A compressed file's name must end in **.[format].[compre**

Browse your computer: Browse... No file selected.

You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

☑ Allow the interruption of an import in case the script

Skip this number of queries (for SQL) starting from the firs

Other options:

☑ Enable foreign key checks

Format:

SQL

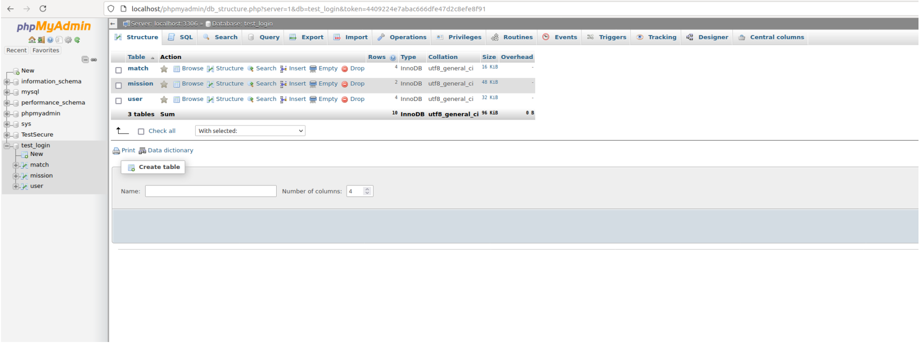
Format-specific options:

SQL compatibility mode: NONE

☑ Do not use AUTO_INCREMENT for zero values

Name	Size	Modified
target		10 Aug
src		1 Jul
build		1 Jul
phpMyAdmin_2.png	183.3 kB	05:26
test_login.sql	18.7 kB	05:24
README.md	1.1 kB	05:22
phpMyAdmin.png	45.1 kB	05:22
buildPath.png	288.2 kB	Sat

Check you got all three tables:



localhost/phpmyadmin/db_structure.php?server=1&db=test_login&token=4409224e7abac666dfe47d2c8efe8f91

Table	Action	Rows	Type	Collation	Size	Overhead
match	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16 KiB	-
mission	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	48 KiB	-
user	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	32 KiB	-
3 tables	Sum	10	InnoDB	utf8_general_ci	96 KiB	0 B

Name: Number of columns: 4

1.1.3 install eclipse:

As our application is still under development, use eclipse to develop and run our applications. Follow the following instruction to install eclipse in Unbuntu:

https://linuxize.com/post/how-to-install-the-latest-eclipse-ide-on-ubuntu-18-04/

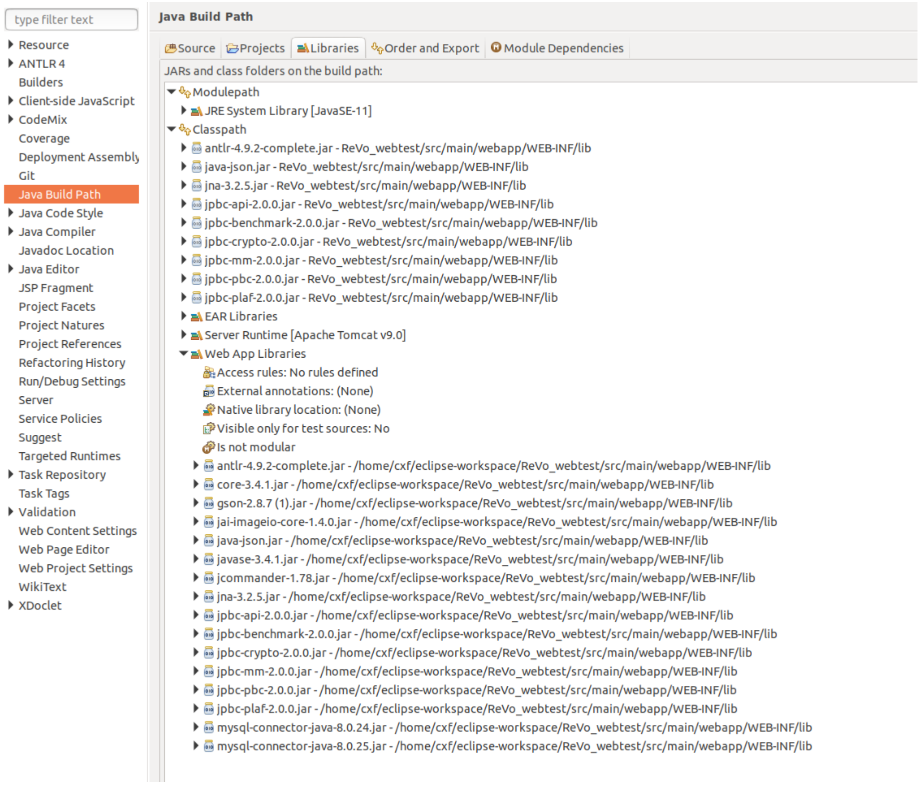
1.1.4 install Tomcat server for eclipse:

Our backend server is run with Tomcat. Installing tomcat server in eclipse is super easy, following the instruction below:

https://crunchify.com/step-by-step-guide-to-setup-and-install-apache-tomcat-server-in-eclipse-development-environment-ide/

1.1.5 Buildpath:

All the required library are in the path: ReVo_webtest/src/main/webapp/WEB-INF/lib/ Using the following to build your build path in eclipse IDE:



Java Build Path

JARs and class folders on the build path:

```
Modulepath
  JRE System Library [JavaSE-11]
Classpath
  antlr-4.9.2-complete.jar - ReVo_webtest/src/main/webapp/WEB-INF/lib
  java-json.jar - ReVo_webtest/src/main/webapp/WEB-INF/lib
  jna-3.2.5.jar - ReVo_webtest/src/main/webapp/WEB-INF/lib
  jpbc-api-2.0.0.jar - ReVo_webtest/src/main/webapp/WEB-INF/lib
  jpbc-benchmark-2.0.0.jar - ReVo_webtest/src/main/webapp/WEB-INF/lib
  jpbc-crypto-2.0.0.jar - ReVo_webtest/src/main/webapp/WEB-INF/lib
  jpbc-mm-2.0.0.jar - ReVo_webtest/src/main/webapp/WEB-INF/lib
  jpbc-pbc-2.0.0.jar - ReVo_webtest/src/main/webapp/WEB-INF/lib
  jpbc-plaf-2.0.0.jar - ReVo_webtest/src/main/webapp/WEB-INF/lib
  EAR Libraries
  Server Runtime [Apache Tomcat v9.0]
  Web App Libraries
    Access rules: No rules defined
    External annotations: (None)
    Native library location: (None)
    Visible only for test sources: No
    Is not modular
    antlr-4.9.2-complete.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    core-3.4.1.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    gson-2.8.7 (1).jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    jai-imageio-core-1.4.0.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    java-json.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    javase-3.4.1.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    jcommander-1.78.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    jna-3.2.5.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    jpbc-api-2.0.0.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    jpbc-benchmark-2.0.0.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    jpbc-crypto-2.0.0.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    jpbc-mm-2.0.0.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    jpbc-pbc-2.0.0.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    jpbc-plaf-2.0.0.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    mysql-connector-java-8.0.24.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
    mysql-connector-java-8.0.25.jar - /home/cxf/eclipse-workspace/ReVo_webtest/src/main/webapp/WEB-INF/lib
```

1.2 Windows and MacOS environment setup:

The only difference of windows and Mac system is the SQL database installation and LAMP server setup. In Windows and MacOS, we use MAMP. To install MAMP follow the following links:
https://www.mamp.info/en/downloads/
Instructions:

1.2.1 Windows:

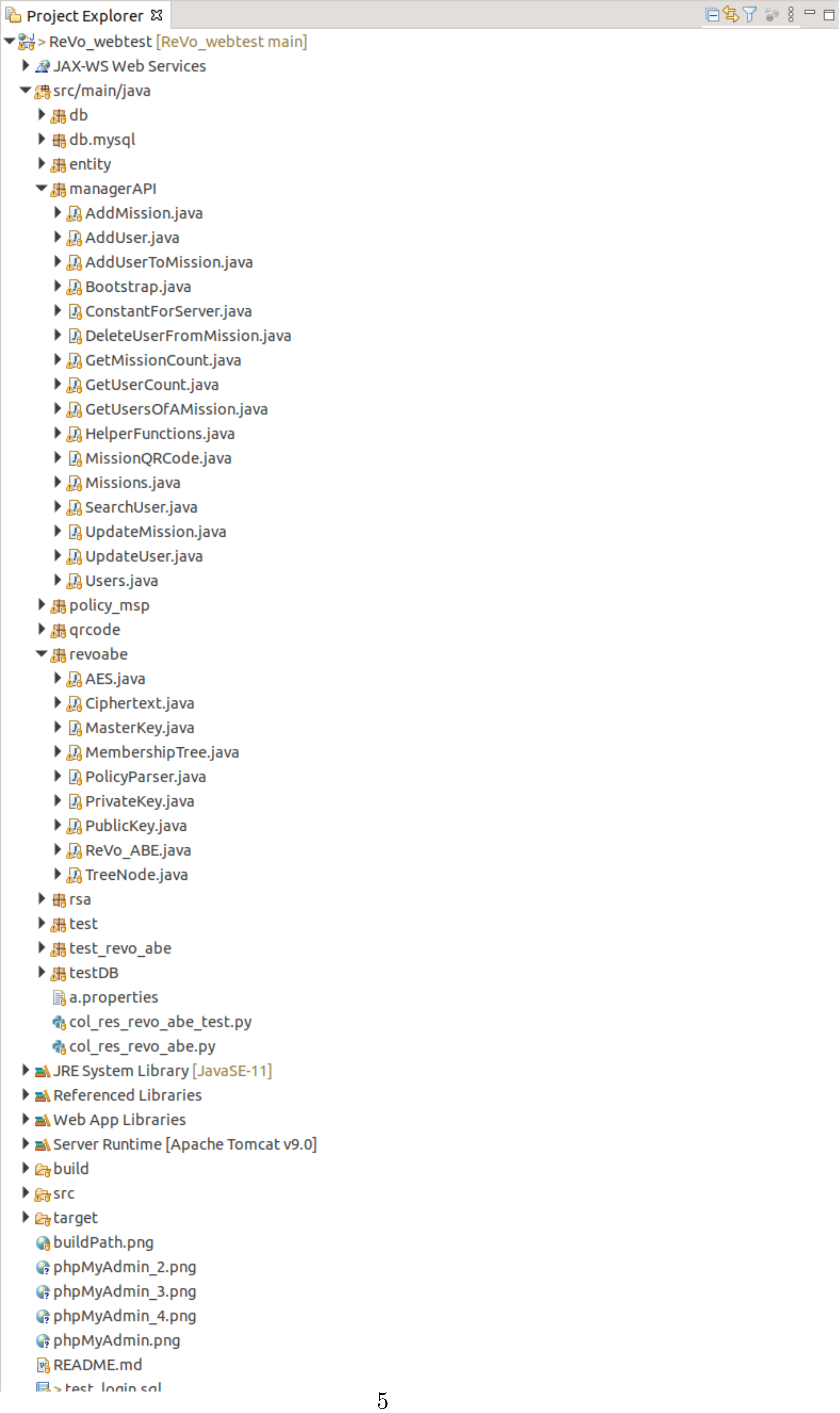
https://documentation.mamp.info/en/MAMP-Windows/Installation/

1.2.2 MacOS:

https://documentation.mamp.info/en/MAMP-Mac/Installation/

2 Use the API:

The API has two parts, the RestFul API that manage mission, User, and Match which resides in manageAPI package and the attribute based encryption API which resides in revoabe package. The structure of the project are as follows:



```
▼ > ReVo_webtest [ReVo_webtest main]
  ▶ JAX-WS Web Services
  ▼ src/main/java
    ▶ db
    ▶ db.mysql
    ▶ entity
    ▼ managerAPI
      ▶ AddMission.java
      ▶ AddUser.java
      ▶ AddUserToMission.java
      ▶ Bootstrap.java
      ▶ ConstantForServer.java
      ▶ DeleteUserFromMission.java
      ▶ GetMissionCount.java
      ▶ GetUserCount.java
      ▶ GetUsersOfAMission.java
      ▶ HelperFunctions.java
      ▶ MissionQRCode.java
      ▶ Missions.java
      ▶ SearchUser.java
      ▶ UpdateMission.java
      ▶ UpdateUser.java
      ▶ Users.java
    ▶ policy_msp
    ▶ qrcode
    ▼ revoabe
      ▶ AES.java
      ▶ Ciphertext.java
      ▶ MasterKey.java
      ▶ MembershipTree.java
      ▶ PolicyParser.java
      ▶ PrivateKey.java
      ▶ PublicKey.java
      ▶ ReVo_ABE.java
      ▶ TreeNode.java
    ▶ rsa
    ▶ test
    ▶ test_revo_abe
    ▶ testDB
      a.properties
      col_res_revo_abe_test.py
      col_res_revo_abe.py
  ▶ JRE System Library [JavaSE-11]
  ▶ Referenced Libraries
  ▶ Web App Libraries
  ▶ Server Runtime [Apache Tomcat v9.0]
  ▶ build
  ▶ src
  ▶ target
    buildPath.png
    phpMyAdmin_2.png
    phpMyAdmin_3.png
    phpMyAdmin_4.png
    phpMyAdmin.png
    README.md
    > test_login.sql
```

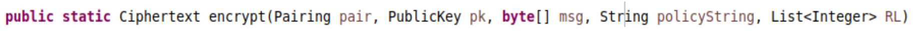
2.1 To use the manageAPI:

All API are java servlet. To use the API, in eclipse, select server and run. Note your IP. To call the API check the report file.

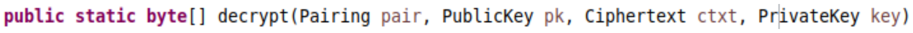
https://github.com/cxfcdcpu/ReVo_webtest/blob/main/August_Report%20.docx

2.2 To use the revo-ABE:

revo-ABE is a powerful attribute based ecryption algorithm which is originally written in python using charm library. In this API, it is rewritten using java with JPBC and antlr library. The implementation is in the revoabe package. To use the library, user need to create a revo-abe instance first. Then generate private key and public key for a user in the bootstrap stage. Then we can use the encryp and decrypt function to encrypt any byte array to ciphertext and decrypt any ciphertext to byte array. Both encrypt and decrypt function has static and non-static implementation as following.



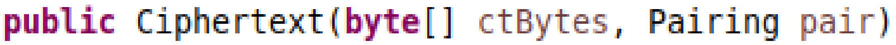
```
public static Ciphertext encrypt(Pairing pair, PublicKey pk, byte[] msg, String policyString, List<Integer> RL)
```



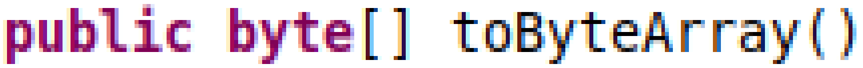
```
public static byte[] decrypt(Pairing pair, PublicKey pk, Ciphertext ctxt, PrivateKey key)
```

(Note: pairing can be generated using JPBC pairing. The buildpath setup are shown in previous of this document)

Also, the ciphertext can be generated from byte array and convert to byte array with the class function as follows:



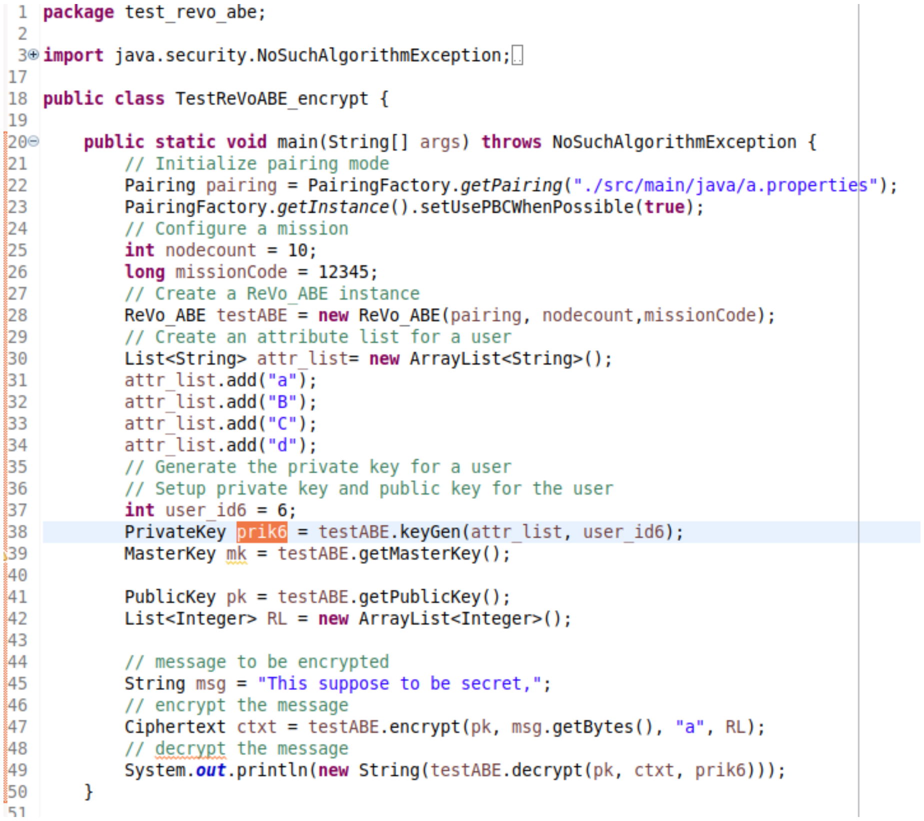
```
public Ciphertext(byte[] ctBytes, Pairing pair)
```



```
public byte[] toByteArray()
```

To test the Revo-abe, just use the test code in:

https://github.com/cxfcdcpu/ReVo_webtest/blob/main/src/main/java
/test_revo_abe/TestReVoABE_encrypt.java



```java
package test_revo_abe;

import java.security.NoSuchAlgorithmException;

public class TestReVoABE_encrypt {

    public static void main(String[] args) throws NoSuchAlgorithmException {
        // Initialize pairing mode
        Pairing pairing = PairingFactory.getPairing("./src/main/java/a.properties");
        PairingFactory.getInstance().setUsePBCWhenPossible(true);
        // Configure a mission
        int nodecount = 10;
        long missionCode = 12345;
        // Create a ReVo_ABE instance
        ReVo_ABE testABE = new ReVo_ABE(pairing, nodecount,missionCode);
        // Create an attribute list for a user
        List<String> attr_list= new ArrayList<String>();
        attr_list.add("a");
        attr_list.add("B");
        attr_list.add("C");
        attr_list.add("d");
        // Generate the private key for a user
        // Setup private key and public key for the user
        int user_id6 = 6;
        PrivateKey prik6 = testABE.keyGen(attr_list, user_id6);
        MasterKey mk = testABE.getMasterKey();

        PublicKey pk = testABE.getPublicKey();
        List<Integer> RL = new ArrayList<Integer>();

        // message to be encrypted
        String msg = "This suppose to be secret,";
        // encrypt the message
        Ciphertext ctxt = testABE.encrypt(pk, msg.getBytes(), "a", RL);
        // decrypt the message
        System.out.println(new String(testABE.decrypt(pk, ctxt, prik6)));
    }
```