基于模糊身份的加密(FIBE)算法实现

一、实验要求

1. 学习掌握基于模糊身份的加密(FIBE)算法原理。

2. 代码实现基于模糊身份的加密(FIBE)算法。

3. 随机指定明文数据，能够输出访问策略Access policy、属性集Attribute Set、密文Ciphertext、密钥Private key和解密结果Decryption results。

二、实验设备

主机：Windows10

工具：IntelliJ Idea

三、实验原理

1. 双线性映射

设G1，GT是素数阶p的群，设g是G1的生成元（即本原根，或称为阶），即G1为由p生成的循环加法群，0为G1的单位元。我们说G1有一个允许的双线性映射即e: G1×G1→GT，则GT为对称双线性群，即具有相同阶p的循环乘法群，1为GT的单位元。该映射当且仅当以下两个条件成立：① 映射是双线性的；② 对于所有的ab，我们有e（ga, gb）=e（g, g）ab。该映射具备非退化性则需满足e（g, g）≠1，即无法映射到GT的幺元。

本算法实现中使用到对称双线性映射即G1×G1→GT。由于双线性群现在的构造是基于椭圆曲线的，而椭圆曲线上的元素是由坐标(x, y)表示的，所以我们将G1的结果输出到Java的控制台，得到的是一个坐标。而GT是一个普通的Zn群，所以其元素的表示是一个数。

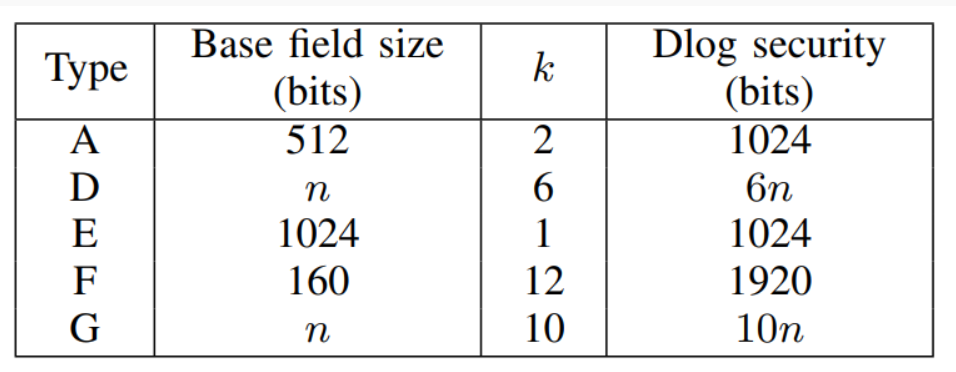
2. JPBC库

PBC库（pairing-based cryptography library）是斯坦福大学研究人员开发的一个免费可移植C语言库。它通过提供一个抽象的接口，使程序设计人员可以不必考虑具体的数学细节，甚至不必考虑椭圆曲线和数论的相关知识就可以实现基于配对的密码体制。JPBC库（Java Pairing-Based Cryptography Library）是对PBC库的Java封装，常用于基于配对的密码学算法仿真程序编写中。

JPBC库共提供四个循环群，其中G1，G2，GT均为阶为p的乘法循环群，而Zp为整数域上的加法循环群。乘法循环群上的点是z值为0的椭圆曲线上的点，而整数循环群上的点是数，二者均可抽象为Element数据类型并用于仿真中。G1，G2，GT中元素的模幂运算、倍乘运算以及相互之间的加法运算，运算结果均为对应群上的元素，Zp中元素的加减乘除运算以及乘方运算，运算结果为整数循环群上的元素。

需要注意的是，现在的密码学相关论文中，习惯将G1，G2设置为乘法循环群。但是基于椭圆曲线的双线性群构造中，G1，G2是加法循环群。所以在2005年以前的论文中，双线性群一般写成加法群的形式。JPBC库中将G1，G2表示成了乘法循环群，因此在加法循环群形式方案的仿真过程中，应特别注意将加法群改写为乘法群的写法再完成进一步仿真。由于加法群中的加法运算对应乘法群中的乘法运算，减法运算对应除法运算（即求逆元），乘法运算对应幂指数运算，而除法运算对应对数运算。故改写过程需要结合以上运算法则。

双线性群（即椭圆曲线）的初始化在JPBC中表现为对Pairing对象的初始化。JPBC库支持A、A1、D、E、F、G六种椭圆曲线，对比如下。我们可以通过代码动态产生和从文件中读取相关参数这两种方法完成上述初始化过程。



当确定椭圆曲线参数后重复调用getG1()，newElement()和newRandomElement()方法，可以得到使用PairingFactory.getPairing(filename)函数导入特定参数的椭圆曲线后，每次调用getG1()函数生成的循环群都是相同的，故可以通过保存椭圆曲线参数至xxx.properties文件并导入这一操作实现循环群的保存。对于群G1，每次调用G1.newElement()函数生成的生成元g都是相同的。然而调用G1.newRandomElement()函数随机获取的群上元素则是不同的。

3. 拉格朗日差值算法

给定n个点(x0,y0),(x1,y1),…,(xn-1,yn-1)，可以决定一个n- 1次多项式：

分别将这n个点代入P(x)中可以得到一组多项式，用矩阵乘法可得：

该组多项式可以表示为：

其中，拉格朗日因子：

4. Shamir秘密共享

即将秘密s分割后共享给n个⼈，⾄少k个⼈组合后才可以恢复秘密s。该分享⽅案具体内容为：

任意取k-1个随机数，构造如下k-1次随机多项式 q(x)：

并使得 q(0) = a0 = s ，取任意n个数xn分别代入多项式，计算 (x1,q(x1))，(x2,q(x2))，...，(xn,q(xn))，将对应的q(xi) 分享给第i个用户。（所有运算均在有限域中进行。）

k个用户通过其秘密分片通过拉格朗日插值算法求解多项式系数可以恢复 q(x)，进而计算出q(0)，也就是s = q(0) = a0。

5. FIBE算法原理

① setup 初始化

（1）生成pairing相关公共参数<e, g, G1, GT, Zp>。

（2）确定属性全集U为整数集合{1, 2, ... , |U|}，以及系统门限值d。

（3）针对每个属性i选择随机数ti∈Zp作为主密钥组件，计算Ti=gti作为对应的公钥组件。

（4）选取随机数y∈Zp，并计算 Y = e(g,g)y 。

（5）最终，系统主密钥 msk = < t1, t2, ..., t|U|, y >，公钥 pk = < T1, T2, ..., T|U|, Y >。

② keygen 密钥生成

（1）随机选择一个d-1次多项式q(x)，使得q(0) = y（即拉格朗日差值法中的a0）。

（2） 针对用户属性集合S中的每个属性i，计算q(i)，进一步计算 。

（3）用户私钥为 sk = {Di}（i∈S）.

③ encrypt 加密

（1）选取随机数s∈Zp，针对明文消息M∈GT，计算。

（2）针对明文属性集合W中的每个属性i，计算。

（3）密文为ct = < E’, {Ei}(i∈W) >。

④ decrypt 解密

（1）如果用户属性集合S和明文属性W重合属性个数不小于d，可按如下方法继续解密。

（2）从所有重合属性中选取d个构成属性集合I。

（3）针对I中的每个属性i，计算，其中是拉格朗日因子。

（4）。

（5）。

四、实验流程

本实验将复现基于模糊身份的加密(FIBE)算法，实现过程中有几点备注如下：

1. 实现过程中所有属性均用整数表示。在实际应用中是通过索引表将每一个整数和一个字符串属性对应起来。

2. 多项式求值和拉格朗日插值在群Zp（在JPBC库中表示为Zr）上进行，因此相应的int值在计算前要转换为Zr Element。

3. 对于重复使用的值一定要记得使用getImmutable()或者duplicate()。尤其是在for循环中。

4. 使用从文件pairingParametersFileName（实例中为a.properties文件）中读取相关参数的方法完成对双线性群（即椭圆曲线）的初始化即在JPBC中对Pairing对象初始化。

5. 生成的公钥pk、主密钥msk、用户私钥sk、密文ct集合也都放入对应的文件pk.properties、msk.properties、sk.properties、ct.properties中。

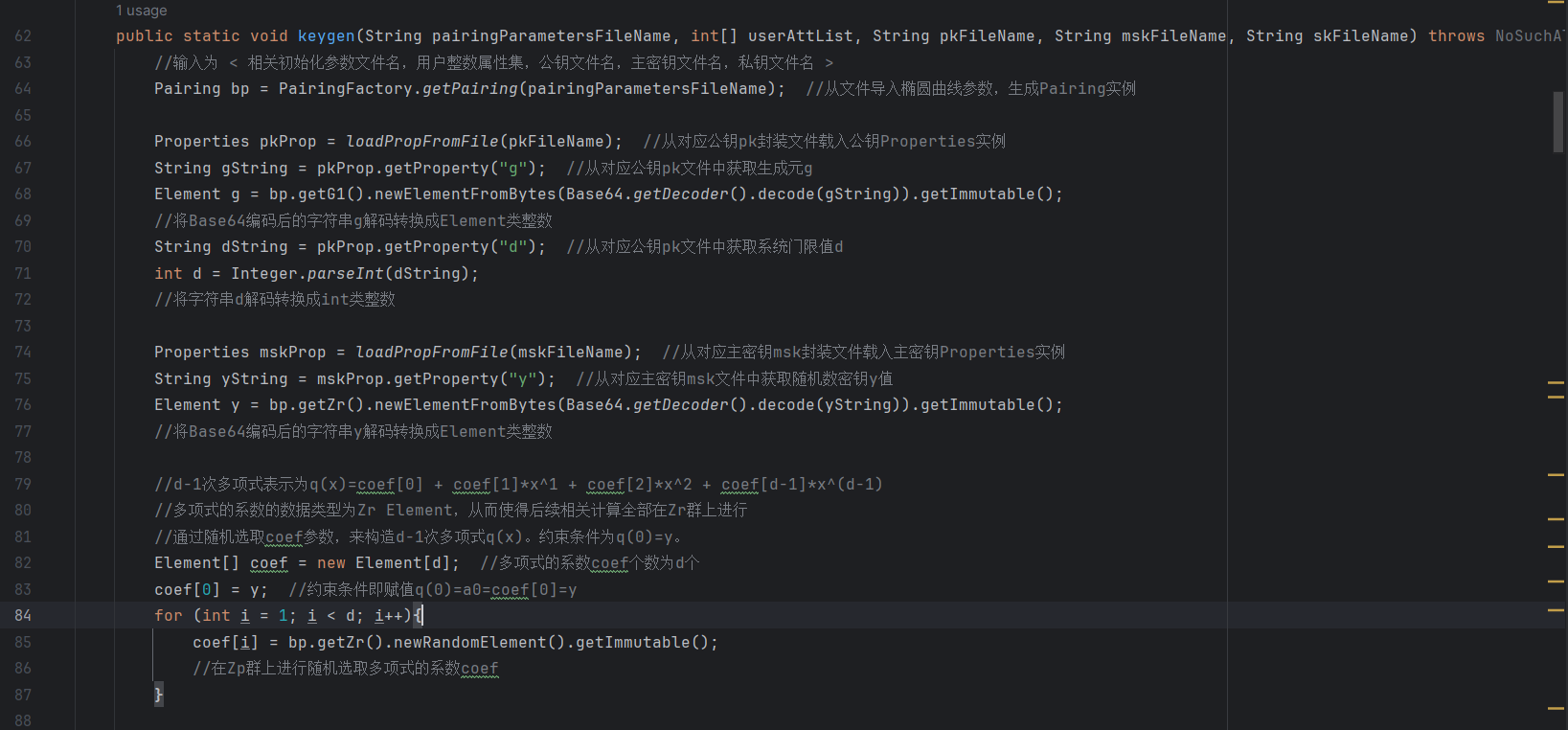
6. 实际应用中解密时只要通过对应属性值能够解出对应的明文点即可，无需将明文点再还原成明文消息串。因此选用的是单向函数SHA-256来对明文消息串进行转换。即需要加密的明文消息串从input.txt文件中读取，并通过SHA-256哈希为GT群上的点Element类，进而继续下一步加密。

具体实现如下。

1. setup初始化

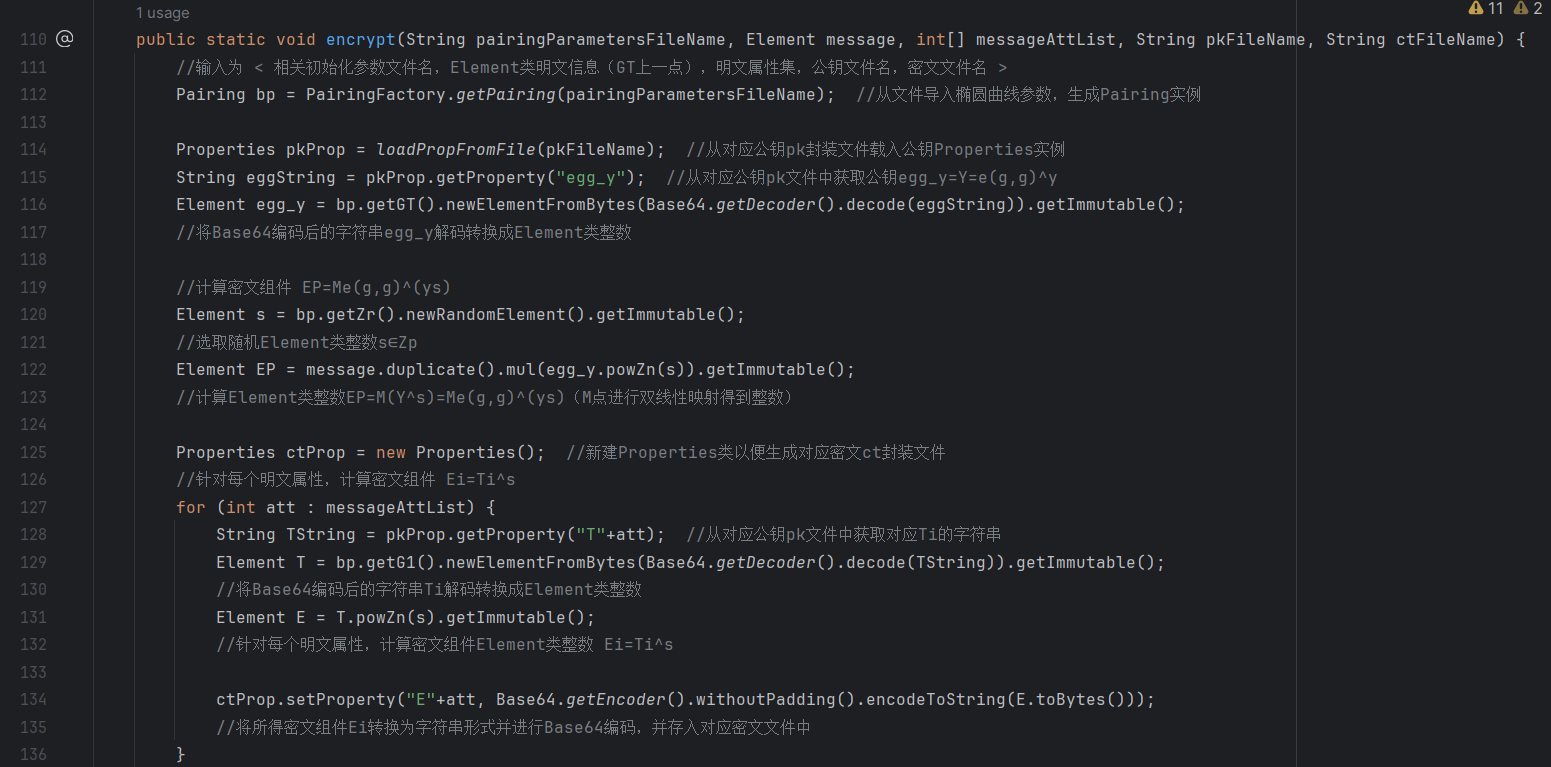


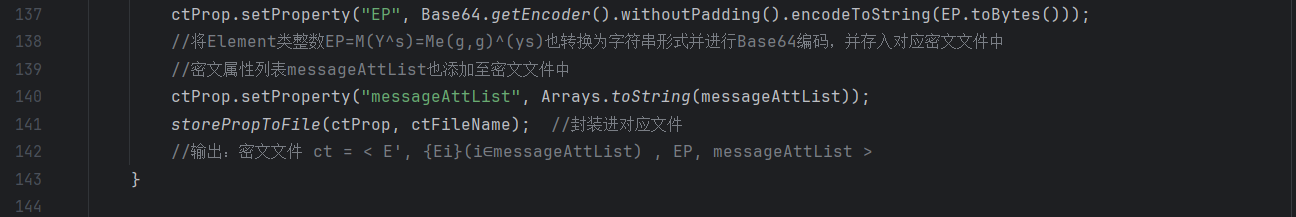
2. keygen密钥生成



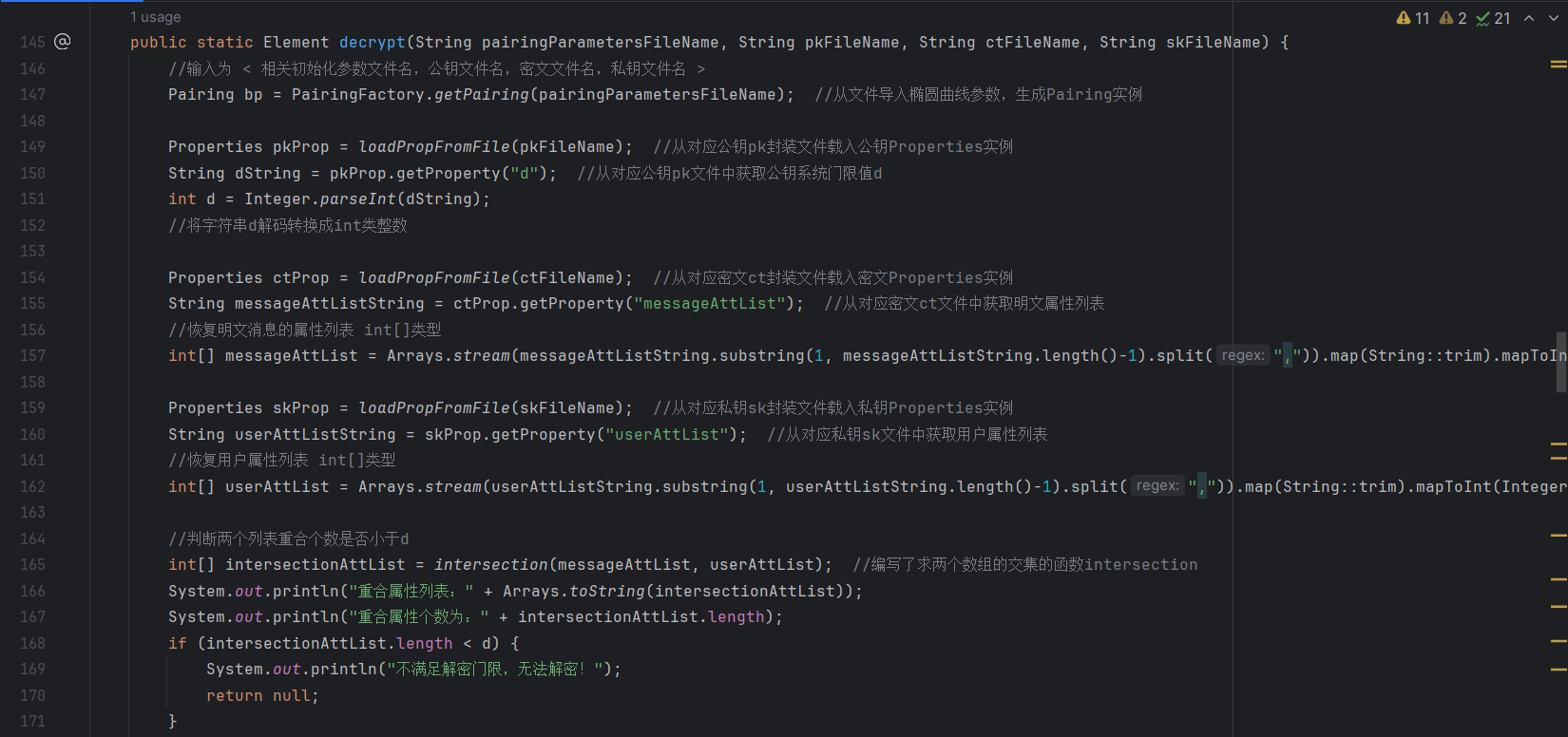


3. encrypt 加密

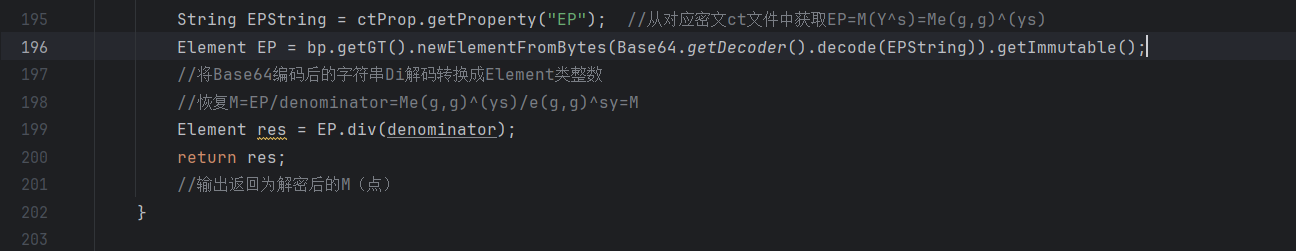




4. decrypt 解密

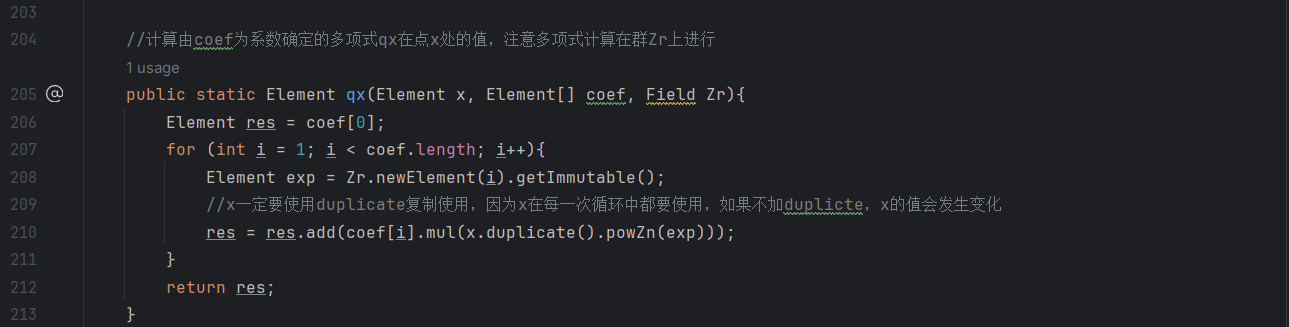






5. 相关函数

① 计算由coef（即an）为系数确定的多项式q(x)在点x处的值



② 求两个数的交集



③ 拉格朗日因子计算



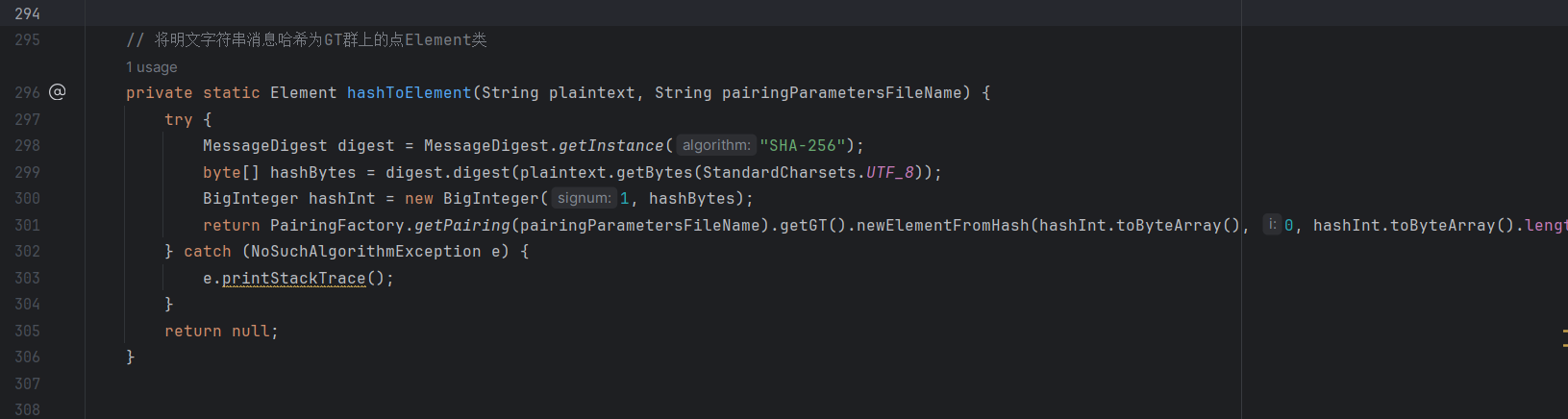
④ 保存和载入参数文件



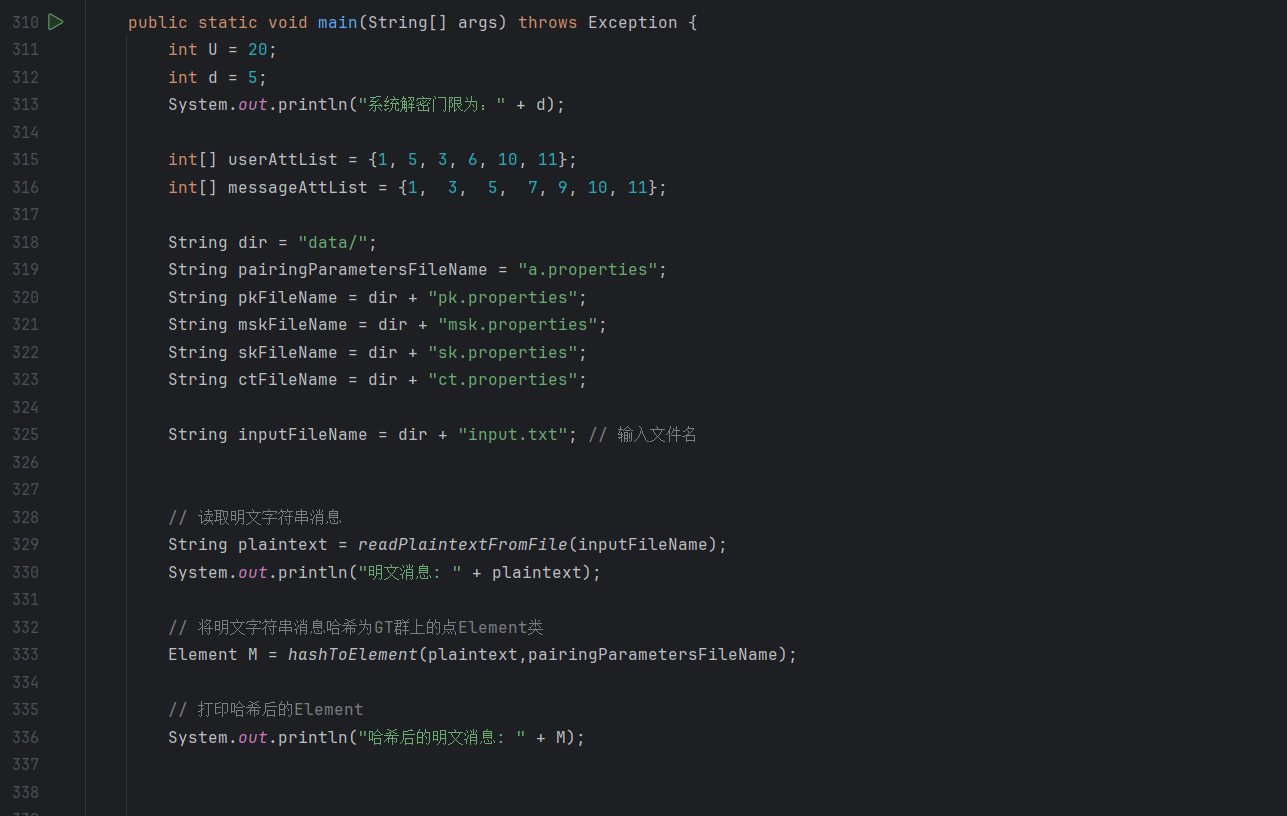
⑤ 从文件中读取字符串消息

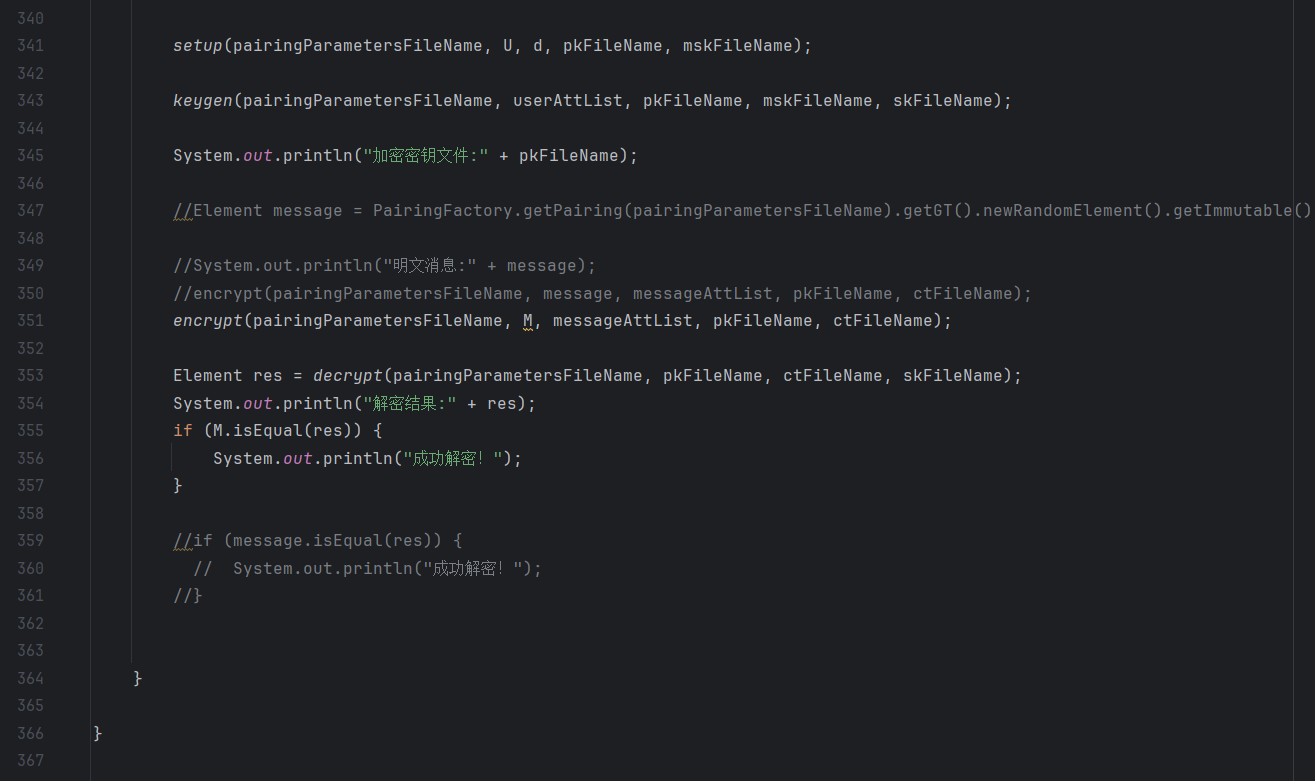


⑥ 将明文字符串消息使用SHA-256哈希为GT群上的点Element类



6. 主函数





7. 运行检验

① 测试数据：

属性集为{1, 2, ..., 20}，即U=20。

系统解密门限值d=5。

明文消息串为“hello”。

② 用户属性与明文属性重合个数大于等于系统解密门限值的情况

测试数据：

用户属性 = {1, 5, 3, 6, 10, 11}；明文属性 = {1, 3, 5, 7, 9, 10, 11}。

运行得到：

系统解密门限为：5

明文消息: hello

哈希后的明文消息:

{x=7010616012506829908431641030078315303461952979236585186069808229003380475397080277444627692448087281027712466890367522595037118951522164295759347826276873,y=1283258037552552373079318247796877001039554043948196141833424327012374268409426519013368452398672774957831046422879020345771774716084020498112008034393769}

加密密钥文件:data/pk.properties

重合属性列表：[1, 3, 5, 10, 11]

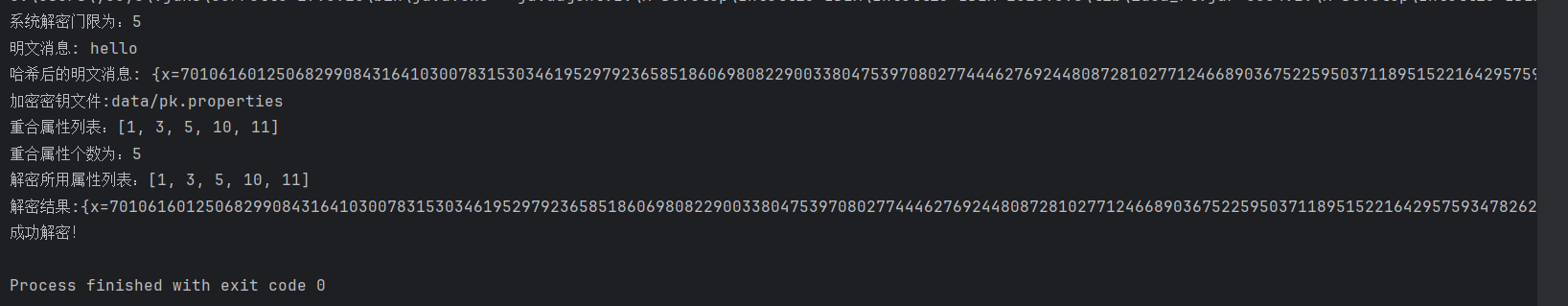
重合属性个数为：5

解密所用属性列表：[1, 3, 5, 10, 11]

解密结果:

{x=7010616012506829908431641030078315303461952979236585186069808229003380475397080277444627692448087281027712466890367522595037118951522164295759347826276873,y=1283258037552552373079318247796877001039554043948196141833424327012374268409426519013368452398672774957831046422879020345771774716084020498112008034393769}

成功解密！



③ 用户属性与明文属性重合个数小于系统解密门限值的情况

测试数据：

用户属性 = {1, 5, 3, 6, 10, 11}；明文属性 = {1, 3, 5, 7, 9}。

运行得到：

系统解密门限为：5

明文消息: hello

哈希后的明文消息:

{x=7010616012506829908431641030078315303461952979236585186069808229003380475397080277444627692448087281027712466890367522595037118951522164295759347826276873,y=1283258037552552373079318247796877001039554043948196141833424327012374268409426519013368452398672774957831046422879020345771774716084020498112008034393769}

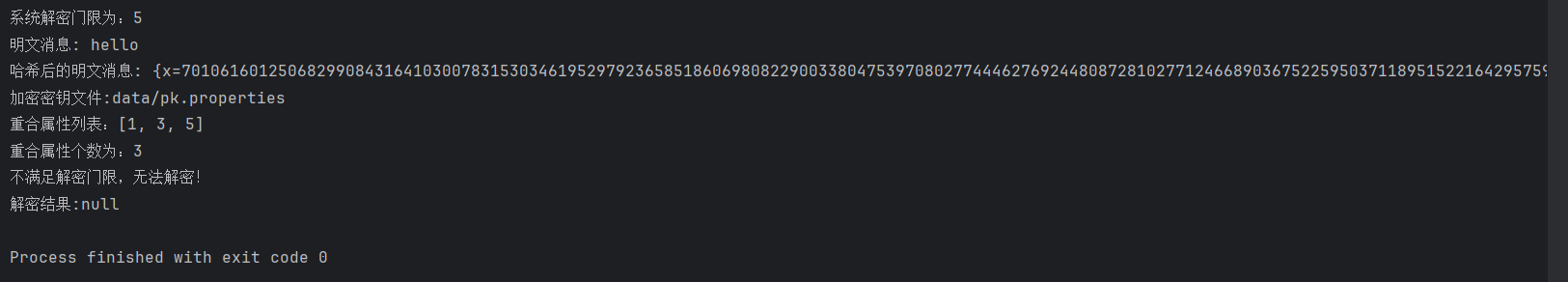
加密密钥文件:data/pk.properties

重合属性列表：[1, 3, 5]

重合属性个数为：3

不满足解密门限，无法解密！

解密结果:null



五、实验结论

在FIBE中，将身份看作是一组描述性属性。它允许具有身份w的用户去解密用身份w’加密的密文，当|w∩w’|≥d，即在一定的度量下，他们的身份重叠大于等于门限值d，也就是他们之间的属性重合大于等于门限值时，所以FIBE具有一定的容错特性，可以使用生物识别特征作为属性输入。

同时，它还具有抗共谋的特性，对于不同用户组合他们的属性，从而解密他们各自不能解密的密文，这在FIBE中是不允许的，主要是使用生成随机多项式来实现抗共谋攻击。FIBE的提出产生了两个新的新应用，第一个就是可以使用生物识别身份的身份基加密系统。例如{身高，体重，血型}等生物特征作为身份。第二个就是属性基加密系统。例如{本科生，研究生，教师}等属性作为身份。

FIBE就是最基本的属性基加密系统，将系统中的属性映射到中，密文和用户的密钥都与属性相关。该机制仅支持基于属性的门限策略，即当用户属性集合与密文属性集合相交的属性数量达到系统规定的门限值以上才允许该用户对密文进行解密。例如，图书馆中某论文的属性集合为{计算机，安全，研究生，英文}，且该论文属性加密门限值为3，则属性基为{计算机，安全，研究生}可以访问该论文，而属性集合为{计算机，安全，本科生}的用户就无法访问它。

该机制中，公钥与系统属性数目线性相关，幂运算和双线性对数目较多，系统越大，计算复杂。

参考文献

[1]Sahai, A., Waters, B. (2005). Fuzzy Identity-Based Encryption. In: Cramer, R. (eds) Advances in Cryptology–EUROCRYPT 2005. EUROCRYPT 2005. Lecture Notes in Computer Science, vol 3494. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11426639\_27（pdf地址：https://eprint.iacr.org/2004/086.pdf）

[2]十陆.（2022）.模糊身份基加密 (Fuzzy Identity based Encryption)算法及JPBC实现. CSDN. 检索于2024年4月2日, https://blog.csdn.net/weixin\_44960315/article/details/122394033.

[3]Intuzgm.（2022）.属性基加密——模糊身份基加密（FIBE）. CSDN. 检索于2024年4月2日, https://blog.csdn.net/m0\_52322997/article/details/124997902.