无需JCE用底层API实现开发RSA PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/273/2021_2022__E6_97_A0_E9_9C_80JCE_E7_c104_273302.htm 若要自己开发RSA的话那都是用底层API实现的，自然是无需JCE。但有一个问题须说明，那就是你所提到的JDK1.1.8，是否可以正确执行我不敢确定，因为我手边没有1.1.8的文档，而我使用的所有API均来自1.2.2.至于1.1.8中是否都一样，我不知道，但想必没什么问题。 还有一个问题：由于RSA的实现均是纯粹的数学原理，故其算法当然也都是针对数字的。至于文本或二进制代码当然也可以，比如转换为字节数组或纯二进制等，具体使用什么方法最好最快我还没找到。所以这就留给你自己解决好了。不过RSA主要是理解算法，搞明白了这些其余不在话下。 这是第一个代码RSAGenerator，用于生成RSA中的p,q,n,m,e,d并把n,e,d写入磁盘中的RSAKey.ser文件。 import java.security.*. import java.math.*. import java.io.*. class RSAInfo implements Serializable { BigInteger e. BigInteger d. BigInteger n. } public class RSAGenerator { RSAInfo info=new RSAInfo(). public static void main(String[] args) { RSAGenerator obj=new RSAGenerator(). try{ obj.getParameter(). obj.writeState(). }catch(NoSuchAlgorithmException ex) { System.out.println("NoSuchAlgorithmException"). } catch(IOException ex) { System.out.println("IOException"). } } public void getParameter() throws NoSuchAlgorithmException { int bitlength=100. int certainty=50. SecureRandom sRandom=SecureRandom.getInstance("SHA1PRNG"). BigInteger

one=new BigInteger("1"). BigInteger p=new BigInteger(bitlength,certainty,sRandom). BigInteger q=new BigInteger(bitlength,certainty,sRandom). BigInteger n=p.multiply(q). BigInteger m=p.subtract(one).multiply((q.subtract(one))). int len=m.bitLength(). BigInteger e. while(true) { e=new BigInteger(len,sRandom). if(m.gcd(e).equals(one))break. } BigInteger d=e.modInverse(m). info.e=e. info.d=d. info.n=n. } public void writeState() throws IOException { FileOutputStream fos=new FileOutputStream("RSAKey.ser"). ObjectOutputStream oos=new ObjectOutputStream(fos). oos.writeObject(info). oos.flush(). fos.close(). oos.close(). } } 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com