**《密码学》课程设计实验报告**

实验序号：05　　　　　　　　　　实验项目名称：公钥密码RSA

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学　　号 |  | 姓　　名 |  | | 专业、班 | 18信安 |
| 实验地点 | 新珈楼B310 | 指导教师 | 王后珍 | | 时间 | 2020.12.21 |
| 1. 实验目的及要求   实验目的：   1. 掌握公钥密码的概念和基本工作方式； 2. 掌握RSA密码、ElGamal密码和椭圆曲线密码的原理与算法； 3. 了解RSA密码、ElGamal密码和椭圆曲线密码的安全性； 4. 了解RSA密码、ElGamal密码和椭圆曲线密码的应用。   实验要求：   1. 掌握RSA密码的实现方案； 2. 掌握ElGamal密码的实现方案； 3. 掌握椭圆曲线密码的实现方案； 4. 了解公钥算法实现中的相关优化算法。   二、实验设备（环境）及要求  Windows操作系统，高级语言开发环境  三、实验内容与步骤  1. RSA密码  （1）RSA加解密算法  ①随机地选择两个大素数p和q，而且保密；  ②计算n=pq，将n公开；  ③计算φ(n)=(p-1)(q-1)，对φ(n)保密；  ④随机地选取一个正整数e，1<e<φ(n)且（e，φ(n)）=1，将e公开；  ⑤根据ed＝1 mod φ(n)，求出d，并对d保密；  ⑥加密运算：  C＝Me mod n （7-4）  ⑦解密运算：  M＝Cd mod n （7-5）  （2）求逆算法：  **欧几里得迭代求逆算法**  求  令计算：      ……    其中每步中的商为整数，余数满足  令计算：    即为  ——————————————————————————————————————————  参考实现（伪码）  **算法1** 利用扩展Euclidean算法求Fp上的逆  输入：素数p和a∈[0. P-1]；  输出：a-1 mod p；   1. u←a，b←p； 2. x1←1,x2←0； 3. 当u≠1重复进行    1. q←⎣v/u⎦ //商    2. r←v-qu //余数    3. x←x2 -qx1； //S[i]=S[i-2]-S[i-1]\*Qi    4. v←u //把除数作为新的被除数    5. u←r //把余数作为新的除数    6. x2←x1  //更新S[i-2]    7. x1←x； //更新S[i-1] 4. 返回（x1 mod p）。 //返回x1 mod p   （3）快速乘方运算  **反复平方乘算法（教材p221）**  设要计算c=ab mod n。    例如：  **算法2** 计算点乘的从右向左的二进制方法  输入：k=（kt-1,…,k2,k1,k0）2,P∈ E(Fq)；  输出：kP；   1. Q←∝； 2. 对于i从0到t-1重复执行   2.1 如果ki=1则Q←Q+P；  2.2 P←2P；   1. 返回（Q）。   **算法3** 计算点乘的从左向右的二进制方法  输入：k=（kt-1,…,k2,k1,k0）2,P∈ E(Fq)；  输出：kP；   1. Q←∝； 2. 对于i从t-1到0重复执行   2.1 Q←2Q；  2.2 如果ki=1则Q←Q+P；   1. 返回（Q）。   **实验（1）**令p=3,q=11,d=7,m=5,手工或编程计算密文C 。  **实验（2）**设RSA密码的 e=3,n=33,C=9, 手工或编程计算明文M 。  **实验（3）**令p=17,q=11, e=7,试计算RSA密码其余参数 。  进一步对于m=88, 计算密文C 。  2. ELGamal密码（参见教材p219）  **例：**设p=19，m=17,构造一个ELGamal密码，并用它对m加密。  **实验（4）**设p=5，m=3,构造一个ELGamal密码，并用它对m加密。  3.椭圆曲线密码（选作）  （1）GF(p)上的椭圆曲线  **实验（5）**取p=23,求出椭圆曲线 y2=x3+x+1的全部解点。（选作）  （2）椭圆曲线密码  理解并实现SM2算法加解密过程。（教材p239）  四、实验结果与数据处理  ***RSA代码整体说明***  相关文件(rsa.py)  接口设计如下图，一共包含四个类，分别为   * ***PublicKey***：公开密钥类 * ***PrivateKey***：私有密钥类 * ***KeyManager***：密钥管理类 * ***RSA***：RSA加解密类   其中每个类的接口定义分别如下(**蓝色框标出为公开方法**)    ***PublicKey***类    ***PrivateKey***类    ***KeyManager***类    ***RSA***类  程序的执行流程可以用下图来表示    ***RSA***流程  ***RSA核心代码说明***  在介绍RSA的加解密之前，我们首先介绍一下**大素数的生成**。其过程可以用以下流程图来表示    大素数生成  这里我们进行素数判定的方法是***miller-rabin***算法，其核心思想是不断地选取在中地基，并检验是否每次有  其代码如下所示    ***miller-rabin***素性测试  有了***miller-rabin***测试，我们的大素数产生算法就容易完成了，其内容如下    素数产生算法  由于在RSA算法中还涉及到求逆的运算，因此**扩展欧几里得算法**也极为重要，其算法流程如下图所示  Euclidean Algorithm | SpringerLink  扩展欧几里得算法  结合上述算法，我们可以容易得到其对应的代码如下，指得一提的是，这里我们使用**栈**来模逆递归函数以提高运行的效率    扩展欧几里得算法  最后一步，我们只差**快速幂**算法了，这里我们以2使用以2为第的快速幂，其对应的代码如下    快速幂  RSA的加解密算法都非常简单，有了前面的铺垫，我们的加解密实现均只需要一行  其中加密算法如下    ***RSA***加密算法    解密算法如下    ***RSA***解密算法  ***RSA实验结果演示***  我们首先生成1024比特长的密钥，其中公钥如下    公钥  私钥如下    私钥  在这里我们对整数***1231231235313123523113***进行加解密测试，可以发现能够成功的进行加解密    加解密测试  ***ECC代码整体说明***  相关文件(ecc.py)  接口设计如下图，一共包含四个类，分别为   * ***EllipticCurvePoint***：椭圆上的点类，实现了点加和乘法 * ***EllipticCurve***：椭圆曲线类 * ***ECCParameters***：椭圆曲线参数类 * ***ECC***：ECC加解密类   其中每个类的接口定义分别如下(**蓝色框标出为公开方法**)    ***EllipticCurve***类    ***EllipticCurvePoint***类    ***ECCParameters***类    ***ECC***类  ***ECC核心代码说明***  在椭圆曲线中，首先一个比较重要的步骤就是**点加**，指得一提的是，这里我们需要分两种情况来讨论，当这两个点**不相等且不互逆**时，我们有    ***ECC***加法  当这两个点**相等**时，我们有    ***ECC***加法  由此可以得到点加的代码如下图所示    点加代码  定义了点加之后，我们就可以完成椭圆曲线上的点与有限域上点的乘法。实际上，**有限域上点的乘法运算类似于快速幂运算**，其代码如下    椭圆曲线点倍乘  接下来可以得到ELGamal型椭圆曲线的密码加解密过程如下  **加密**   1. 用户A选择一个随机数 2. 用户A计算点 3. 用户A计算点 4. 用户A计算 5. 用户A发送加密数据给用户B   其对应的代码如下所示    椭圆曲线加密    **解密**   1. 用户B用自己的私钥求出点 2. 求出分量的逆 3. 对解密，得到明文数据   其对应的代码如下所示    椭圆曲线解密  ***ECC实验结果演示***  我们对明文***1231231235123***进行加解密的测试，其中椭圆曲线的参数参照**SM2椭圆曲线公钥密码算法推荐曲线参数**，如下所示    椭圆曲线参数  可以发现能够成功的进行加解密    加解密结果  五、分析与讨论   1. 在RSA算法中，我们通常需要产生大素数来生成公私钥。而如何进行快速的素性判定是能够高效生成素数的关键，因此我们通常需要在安全性和实用性之间进行权衡。这也是使用非确定性素数判定算法Miller-Rabin的原因 2. 与RSA相比，ECC基于椭圆曲线上的离散对数问题。由于目前还没有亚指数级求解椭圆曲线上离散对数问题的算法，因此同等密钥长度下，ECC较为安全 3. 为了确保椭圆曲线密码的安全，我们需要注意不能够使用弱的椭圆曲线。 | | | | | | |
| 六、教师评语  签名：  日期： | | | | 成绩 | | |