**深 圳 大 学 实 验 报 告**

**课程名称： 计算机安全导论**

**实验项目名称­： 非对称密码实验**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 林秋镇**

**报告人： 学号： 班级：**

**实验时间： 2024年10月18日**

**实验报告提交时间： 2024年10月27日**

**教务部制**

|  |
| --- |
| 实验目的与要求：   1. 理解公钥密码的基本思想 2. 理解RSA算法的原理 3. 掌握RSA算法的输入输出格式和密钥格式 4. 掌握RSA算法实现加解密过程、数字签名 |
| 实验环境：  客户端  Microsoft Visual Studio 2005等 |
| 实验原理：  RSA原理  RSA算法由Rivest, Shamir & Adleman于1977年在MIT提出，是目前最广泛接受和实现的通用公钥加密方法。    RSA属于分组密码，基于大整数运算，对于某个***n***，明文跟密文都在***0 ~ n-1***之间的整数，加解密方法如下：  ***C* = *Me* mod *n*, 这里 0≤*M*< *n***  ***M* = *Cd* mod *n* = *Med* mod *n***  其中***M***是明文，***C***是密文，公钥 ***PU =｛e, n｝***, 私钥 ***PR =｛d, n｝***。  为了使RSA能够用于公钥加密，必须满足下列要求：   * 找到***e, d, n***的值，使得对所有的***M* < *n*, *Med* mod *n* = *M***。 * 对所有满足***M* < *n***的值，计算***Me***和***Cd***相对容易。 * 给定***e***和***n***，不可能推出***d***。   RSA密钥设置如下：   * 随机选择两个大素数: ***p, q***，并计算他们的乘积 ***n = p × q*** 作为加密和解密时的模。 * 接着计算***n***的欧拉函数值 ***ø(n) = (p - 1)(q - 1)***。 * 选择与**n**互素的加密密钥 **e** , 满足***1 < e < ø(n), gcd( e, ø(n) )=1***。 * 计算**e**关于模***ø(n)***的乘法逆元***d***，作为解密密钥。 * 公布公钥: ***PU = {e, n}*** ，保留私钥: ***PR = {d, n}***。   RSA安全分析：   * 穷举搜索攻击：试遍所有可能的私钥，所以***e***和***d***的比特数越大，算法越安全，同时运算更复杂，系统运行更慢。 * 分析RSA密码算法：主要重点在于如何分解***n***为两个素数。由于大数***n***具有很大的素因子，因式分解问题非常困难。目前1024比特的密钥安全强度足够了。   数字签名的方法：   * 签名生成：   + 使用哈希函数（如***SHA-256***）计算消息***m***的哈希值***h***。   + 使用私钥***d***计算签名***s***, ***s = hd mod n*** 。   + 签名 ***s***附加在消息***m***上发送。  签名验证：接收消息*m*和签名*s*。再次计算消息*m*的哈希值*h’*。使用公钥*e*验证签名, *h = se mod n*。  * + 如果***h'***与***h''***相同，说明签名有效，消息未被篡改。 |
| 实验内容：   1. 使用自己熟悉的语言实现RSA加密算法 2. 通过自己编写的RSA算法实现对文档进行加解密、数字签名方法 |
| 实验步骤与结果：   1. 使用自己熟悉的语言实现RSA加密算法   下面将在Eclipse平台用Java语言实现RSA加密算法。   1. 密钥相关变量声明与初始化     这段代码声明了 RSA 算法所需的关键变量，用于生成和存储公钥及私钥。n 表示 RSA 模数，是两个大素数的乘积；d 是私钥，用于解密；e 是公钥，用于加密。通过设定 bitlen 为 2048 位，保证密钥长度符合现代安全标准；nlen 存储模数的位数，用于后续读取二进制明文的的位数。   1. RSA 密钥生成构造函数     此构造函数用于生成 RSA 算法的公钥和私钥。首先，通过 SecureRandom 生成器生成两个大素数 p 和 q。接着计算模数 n 和欧拉函数 ，然后选择公钥 e，确保其与 互质。最后，通过模逆计算出私钥 d，使得加密与解密过程相互关联。   1. RSA 加密与解密函数     这两个函数实现了 RSA 算法中的加密和解密过程。encrypt 函数接收一个明文消息，将其通过公式转换为密文。相应地，decrypt 函数接收密文，并通过公式 进行解密，恢复出原始的明文。   1. 文件加密方法     该方法实现了将指定文件中的内容进行加密并保存到新文件的功能。首先，使用 FileInputStream 读取输入文件的内容，每次读取 nlen - 1 位，因为M转为数字后的位数小于n，所以保证了M转为数字后必然小于n。读取的字节被转换为 BigInteger 格式后，通过 RSA 的 encrypt 方法进行加密。加密后的结果以字符串形式写入输出文件，并在每个加密数据后添加换行符，以便于后续处理。最后，使用 Arrays.fill 方法清空缓冲区。   1. 文件解密方法     该方法用于将加密文件中的内容逐行解密并保存到新文件。首先，使用 BufferedReader 读取指定的输入文件，每次读取一行。读取到的每行内容被转换为 BigInteger 格式，随后使用 RSA 的 decrypt 方法进行解密。解密后的数据以字节数组形式写入到输出文件中，确保原始数据得到恢复。最后，使用 Arrays.fill 方法清空解密后的字节数组。   1. 数字签名方法     该方法首先读取指定文件的所有内容，将其拼接为一个字符串。接着，使用 SHA-256 算法计算该字符串的摘要值，以确保文件内容的唯一性和完整性。最后，将计算得到的摘要值转换为 BigInteger，并利用 RSA 的私钥对其进行加密，生成文件的数字签名。通过这种方式，发送者可以证明文件内容未被篡改，并且仅由持有私钥的发送者签署。   1. 签名验证方法     该方法首先读取指定文件的全部内容，将其拼接为一个字符串。然后，使用 SHA-256 算法计算该字符串的摘要值，以确保文件内容的完整性。接着，使用 RSA 的公钥对传入的数字签名进行解密，获取发送者在签名时生成的摘要值。最后，通过比较解密得到的摘要值与当前计算的摘要值，验证文件的签名是否有效。这一过程确保了文件未被篡改，且确认了签名的有效性。   1. RSA 加密与解密演示     这段代码实现了 RSA 加密和解密的基本功能。首先，通过创建 RSA 类的实例，生成所需的公钥和私钥，并输出模数 n、公钥 e 和私钥 d。然后，定义了三个文件路径，分别表示输入的明文文件、加密后的文件和解密后的文件。在 try 块中，首先调用 encryptFile 方法将明文文件进行加密并保存到指定路径，然后打印加密成功的消息。接着，调用 decryptFile 方法对加密文件进行解密并保存为另一个文件，最后打印解密成功的消息。如果在过程中出现 IO 异常，将捕获并输出异常信息。   1. 数字签名演示     该代码实现了对指定文件内容的数字签名和验证过程。首先，定义了待签名文件的路径，并调用 signFile 方法对该文件内容进行签名，生成一个数字签名。随后，输出生成的签名值。接着，通过调用 verifyFile 方法，将文件路径和签名传入，以验证该签名是否有效。最后，输出验证结果，确保文件内容的完整性和签名的可信性。如果在过程中发生异常，将捕获并打印相关的错误信息。此流程确保了数据的安全性和完整性。  2、通过自己编写的RSA算法实现对文档进行加解密、数字签名方法   1. 加解密   待加密的明文    加密后的密文    解密后的明文  可以看到，解密后的明文与加密前的明文一致，说明明文被完美还原，编写的RSA算法代码正确性得以验证。    加密和解密运行结果     1. 数字签名   待签名文件      数字签名的生成和验证  可以看到，数字签名验证成功，说明数字签名生成和验证的代码正确。 |
| 实验结论：  在完成了RSA算法的实验之后，我有了很多收获。首先，我理解了公钥密码的基本思想，即加密和解密使用不同的密钥，发送方使用接收方的公钥来加密信息，而接收方则用自己的私钥来解密，这保证了通信的安全性。其次，在学习RSA算法的过程中，我掌握了其背后的数学原理，包括大数质因数分解的难度以及欧拉定理的应用，这些是RSA安全性的基础。最后，通过实践操作，我学会了如何使用RSA算法进行数据的加密与解密，并且实现了基本的数字签名功能，加深我对RSA算法理解的同时，还巩固了我的代码能力。  在实验中，我也对比了非对称密码与之前实验做的对称密码之间的差异，发现了它们各有优势和局限。  非对称密码的优点主要体现在以下几个方面：   * 非对称加密技术中，每个用户拥有一对密钥，即公钥和私钥，公钥可以公开，而私钥则需要保密。这种机制使得密钥的分配变得容易，因为不需要像对称加密那样在通信之前秘密地交换密钥。 * 非对称加密支持数字签名，可以用来验证消息的来源及其完整性，这是对称加密所不具备的功能。 * 由于使用的是公钥加密，只有拥有对应私钥的人才能解密，因此即便是在不安全的网络环境中传输数据，也能保证数据的安全性。   然而，非对称密码也有其不足之处：   * 非对称加密计算复杂度高，处理速度通常比对称加密慢得多。 * 非对称加密算法通常需要更多的计算资源和存储空间，特别是在处理大量数据的情况下。   相比之下，对称密码的优势在于其加密解密速度快、效率高，且所需的计算资源较少。但是，它的一个主要缺点就是密钥管理较为复杂，尤其是在开放网络环境中，确保密钥的安全交换是一个挑战。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。