Github 账号： https://github.com/yikesoftware

个人博客关于密码学大作业的链接：
https://github.com/yikesoftware/cryptography_assignment/blob/main/README.md

题目：RSA 大礼包

摘要：

对题目给出的 21 个片段的加密数据进行了细致的观察与分析，采用 Fermat 分解法和 p-1 分解法成功分解了 Frame2、Frame6、Frame10 和 Frame19 的模数并由此得到了正确的明文消息；使用公共模数攻击法和低加密指数攻击法找到了存在某些安全缺陷的消息片段，成功破译了 Frame0、Frame3、Frame4、Frame8、Frame12、Frame16 和 Frame20 的明文消息；借鉴因数碰撞的思想，用欧几里德算法遍历所有模数，求出 Frame1 和 Frame18 的模数的公因数，进而成功分解了 Frame1 和 Frame18 的模数得到正确的加解密参数，破译了明文消息。

题目描述

有人制作了一个 RSA 加解密软件。题目给出了该软件发送某个明文的所有参数和加密过程的全部数据（附件 3-1）。

Alice 使用该软件发送了一个通关密语，且所有加密数据已经被截获，尝试分析从加密数据恢复该通关密语及 RSA 体制参数——如能则给出原文和参数；如不能则给出已恢复部分并说明剩余部分不能恢复的理由。

过程

1. 攻击思路

考虑了一些 RSA 加密系统可能存在的一些隐藏缺陷：如模数生成时选择了不安全的素数，明文片段之间可能存在的联系等等。对此，可以用一些针对特定系统缺陷的攻击方法，尝试对模数进行分解，对明文消息进行破译。

1) 费马分解法

观察题目，可以发现一句提示“如果参数选取不当，同样存在被破译的可能”，由此猜测，在 21 组模数中，必然有存在安全缺陷的参数，故可以尝试整数分解法进行攻击。

费马分解法在两个素数 p 、q 十分相近的情况下，可以有效地对模数 N 进行分解。所以，可以考虑采用费马分解法对所有的模数进行攻击，若存在满足条件的模数，就可以成功破译。

2) Pollard p-1 分解法

思路同上，若某个模数 N 的素因数为 p ，1p - 存在小因数，则可以用 Pollard p-1 分解法破解该模数 N 。根据算法原理，通过编程实现，对每个模数进行攻击尝试，若存在满足攻击条件的数，就能够有效破解。

3) 低加密指数攻击

查阅相关文献后了解到：相同的明文消息 m 采用同一公钥 e 和不同模数 N 加密后得到根据不同的密文 c ，应用中国剩余定理求解同余式组，可以有效地恢复明文消息。其中，对于不同的模数 N ，要求它们的最大公因数为 1，且模数 N 的个数与公钥 e 的值相同。

通过对所截获加密帧数据的观察发现：在 21 个片段中，有些片段中的加密密钥是相同的，具体如下：Frame7、Frame11 和 Frame15 这 3 个分片均使用“3”作为加密密钥；Frame3、Frame8、Frame12、Frame16 和 Frame20 这 5 个分片均使用“5”作为加密密钥，通过验证，这两组消息分片中的模数各自互素。如果这两组数据是由相同的明文片段加密得到的，那么它们就完全符合低加密指数攻击条件。所以，假设这两组数据均符合条件，尝试低加密指数攻击法进行破译。

下面，分别对两组消息片段进行破译并验证。

(1)对 Frame7、Frame11 和 Frame15 的攻击

① 初次尝试

假设 Frame7、Frame11 和 Frame15 三个片段存在相同的明文，基于该攻击方法的原理，可以对其进行有效攻击，得到有意义的明文消息。

然而在攻击过程中，通过 Python 语言进行编程实现，最终得到的结果并不符合题目中对明文消息格式的要求，且毫无语义。因此，本次攻击没有成功。说明这三组消息的明文可能并不完全相同，才导致攻击失败。

② 算法改进

仔细研究该攻击方法的原理发现：其所需的消息个数并非一定要等于公钥值的大小，即不要求模数 N 的个数与公钥 e 的值相同；起决定作用的是 m^e 与 N_1,…,N_k 的大小关系（N_1,…,N_k 为加

密时所用消息的模数）。当m^e小于N_1,…,N_k时，可以对其进行有效攻击。

观察题目中明文格式和模数格式的要求：明文长度固定为512bit，模数长度固定为 1024bit，针对 e=3 的攻击情形，|m^3|=1536 远小于任意两个模数的乘积，因此针对 Frame7、Frame11 和 Frame15 的攻击，只要其中任意两个消息存在相同明文，即可进行有效攻击。

通过改进后的攻击算法，依然未能获得正确的明文消息，因此可得出结论：这三个消息的明文片段均不相同。 (2)对 Frame3、Frame8、Frame12、Frame16 和Frame20 的攻击。

(2)对Frame3、Frame8、Frame12、Frame16 和Frame20 的攻击

借鉴上述改进后的算法思想，先对m^e与N_1,…,N_k的大小关系做出判断。通过计算可得知：当 e=5 时，只需存在 3 个消息拥有相同的加密明文即可实施有效攻击。

通过编程实现，对任意 3 个消息片段进行计算均可得到有效的明文消息，且得到的这些明文均相同。由此，破解得到 Frame3、Frame8、Frame12、Frame16

和Frame20 这5个消息的明文片段，并证明这 5 个消息由同一明文片段加密所得。

4) 公共模数攻击

当系统中不同的消息共用一个模数 N ，只有 e 和 d 不同，系统将是危险的，此时，攻击者可能无需分解N就能够恢复明文。

通过观察加密帧数据已经知道 Frame0 与 Frame4 中的模数N 是相同的，若这两个消息存在相同的明文，则可以使用共模攻击的方法进行有效攻击。

根据共模攻击的原理，通过编程实现对 Frame0 和 Frame4 进行攻击测试，最终得到了符合明文格式要求，且具有语言意义的明文消息。

2. 因数碰撞法求两个模数的最大公因数

通过查阅资料发现了一个巧妙的想法——从求任意两个模数的最大公因数入手，实现对大整数的因数分解。若某两个数的最大公因数为 1，则说明这两个数互素；若最大公因数大于 1，则说明该最大公因数同为这两个模数的一个因数，可以分别进行除法运算，进一步求出两个模数的另一个因数，即间接实现了大数分解。与费马分解法和 p-1 分解法相比，这种方法在运算效率方面有显著优势，计算可行性更强。

在实现过程中，采用欧几里得算法，对 21 个模数 N 两两求最大公因数，需要计算 210 次，其中，Frame0 与 Frame4 中的模数 N 是相同的，不予考虑，计算余下的 209 组模数 N ，可得到结果：Frame1 与 Frame18 中的两个模数 N 存在不为 1 的最大公因数，进而成功地对 Frame1 的模数与 Frame18 的模数进行分解，得到重要参数 p 和 q ，并依此计算出 phi(N) ，再由公钥 e 计算得到私钥 d 。最终，使用私钥 d ，根据解密算法得到明文消息，即可最终实现对密文的完全破解。对得到的明文消息进行加密验证，与所截获密文消息完全相同。

至此，完成了对 Frame1 与 Frame18 的完全破解，得到了有意义的明文消息与 RSA 体制参数 p 和 q 。

3. 猜测明文攻击

目前已经将现有条件下所能实现的全部常见攻击方法进行了试验，共得到了 13 个分片的明文，不考虑重复发送的消息，有 8 个片段的明文数据。根据已有的关键信息，通过查阅相关文献资料和互联网搜索，采用语义分析与加密验证相结合的方法，最终恢复了全部明文信息。

4. 寻找随机数生成规律

仔细阅读题目发现，题目中提到“素数 p 由某一随机数发生器生成”和“素数 q 可以随机选择，也可以由 2)中的随机数发生器产生”这两条重要信息。这意味着，在现已分解得到的 Frame1、Frame2、Frame6、Frame10、Frame18 和 Frame19 这六个分片中的 12 个素数中，至少有 6 个素数是由同一个随机数发生器生成的。于是考虑，能否通过已有的素数，找出随机数的生成规律，进而破解所有的素数参数呢？下面将研究的重心转移到如何还原随机数发生器的问题上。

随机数发生器生成的随机数是伪随机的，也就是说，根据特定的数学函数、利用计算机强大的计算功能生成的数，其实是有内在规律可循的，并不是真正意义上的随机。伪随机与真随机的区别在于，对于给定的初值，一个伪随机生成器产生的随机数是能够完全确定的，而真随机数是完全无法预测的。从信息论的角度来说，在信息的传递过程中，信息量只能保持不变或减少。所以当有限的比特信息（在随机数发生器中称输入的初值为“种子”）生成更多、更长的伪随机序列时，其信息量并没有增加。因此，随机数发生器生成的伪随机序列一定会在某些方面呈现出相关特征。利用这些特征可以进行利用已知的六组素数对随机数的生成规律进行探究，并还原随机数发生器。

分别检测这 12 个素数序列的 0-1 分布特征、游程分布特征、移位相加特征、采样特征，在获得的数字特征中观察分析其共性特点，去除非生成器产生的素数的干扰，然后再与已知的随机数发生器的特征特点进行比较，猜想得到可能使用的随机数发生器结构。

对于可能使用的移存器结构和同余式结构，通过不断的尝试，对猜想的随机数发生器结构进行实验验证。最终根据实验结果和数据特征，联想经典的 RC4 与 BBS 生成器的级联结构，确定了真实的随机数发生器的结构。利用获得的随机数发生器遍历初值，将产生的序列与模数 N 计算最大公因数，确定是否为模数 N 的因数，得到 RSA 重要加密参数，破解加密数据帧内容。

总结

1. 参数的选取

(1)　关于素数 p、q 的选取

首先我们采用了两种常见的整数分解法——费马分解法和 Pollard p-1 分解法，成功地分解出了四组模数。这两种整数分解法都是确定性的算法，通过寻找特定的条件，使得素因数在某区间分布的概率大大提高，这样一来，算法的实现过程就变得简单可行。只要选取的素数符合某些规律，其生成的模数就存在用该类方法破解的安全威胁。

然后，我们利用已知的六组素数参数，分析它们之间的关系，找到了随机数产生的规律，最终还原了随机数发生器，进而求出了所有的素数参数。这也向我们证明：如果采用数学方法生成伪随机数，就存在随机数发生器被还原的风险。所以，

我们在选择随机数时，应尽量减弱随机数之间的相关关系，可以采用增大随机函数的复杂性的方法，例如增加代数次数、线性复杂度等。

(2)　关于模数 N 的选取

根据公共模数攻击的原理，我们成功破译 Frame0 和 Frame4 的过程，耗时仅 0.03 秒，由此可知，将同一个明文信息发送给不同的人时，尽量不要选取相同的模数 N。否则，当窃听者截获加密后的不同明文后，仅根据已知的公钥，就能够恢复明文消息。

此外，不同的用户应该选用不同的模数 N，，用户之间不能共享。这是因为，当某中心选择公用的模数 N，然后把(e, d)分发给众多用户，任何一对(e, d)都能分解模数 N，从而，本质上来讲，任何用户都可以求出共享该模数的每个用户的解密密钥 d。

(3)　关于加密指数 e 的选取

为了提高运算效率，RSA 算法设计者常常选取较小的加密指数 e 。然而，利用很小的 e ，不同的模数，去加密同样一段明文信息时，系统将是非常危险的。若密文不慎被攻击者截获，仅根据加密指数 e ，就可以有效地恢复明文信息。

所以，在选择加密指数 e 时，应尽量选择 16bit 以上的数，即不影响计算效率，又保证了算法的安全性；另外，当明文信息很短时，可以使用独立随机值填充的方法，降低明文信息的相关性，使得攻击者无法用低加密指数攻击法破译算法系统。

2. 对明文的要求

在本题的破译过程中，我们用到了猜测明文攻击的方法，根据已知的 8 个明文片段，通过互联网搜索，结合语义、语境的判断，最终恢复了全部明文信息。这也证实了，当明文空间较小时，猜测明文攻击是一种有效的攻击方法。

要想避免以这种方式破译出明文，我们在与特定对象进行通信交流时，可以提前商定某种语法规则，使得语言表达的意思不是其本身的意思，这样，即使消息被中途截获并破译，攻击者也无法理解明文所表达的意义；或尽可能地增大明文空间，并提前商定哪些字符不具有语义，也能有效防止攻击者对明文消息的恢复；还可以将明文分片加密后打乱顺序随机发送，采用只有合法接收者才能还原的发送顺

序，即使消息全部被破译也难以恢复成完整的具有明确语义的语句，降低猜测明文攻击的可行性。

参考文献

[1] Whitfield Diffie, Martin Hellman. New Directions in Cryptography[J]. IEEE Transactions on Information Theory, 1976, 22(6): 644-654.

[2] Ronald Rivest, Adi Shamir, Leonard Adlemen. A Method for Obtaining Digital Signatures and Public-key Cryptosystems[J]. Communications of the ACM, 1978, 21(4): 120-126.

[3] 谢健全, 杨春华. RSA 中几种可能泄密的参数选择[J]. 计算机工程, 2006, 32(16): 118-119.

[4] 王小云, 王明强, 孟宪萌. 公钥密码学的数学基础[M]. 科学出版社, 2013.

[5] GMP package, GNU Multiple Precision Arithmetic Library, https://gmplib.org/.

[6] Miracl package, http://www.shamus.ie/.