

RSA原理和手工，工具解密

翻译

[cried_cat](#) 于 2018-04-01 09:06:37 发布 3201 收藏 2
2018/3/30

criedcat

密码学writeup

笔者借阅一些网络文献来总结笔者一周来对密码学以及rsa加密方法的认知。

本writeup提纲：1rsa加密理论2手工解密rsa算法3解析解密rsa题目4ctf的rsa

Rsa理论参考链接：

<https://blog.csdn.net/dbs1215/article/details/48953589>（链接中笔者发现了一些错误，读者请自行去辨别，看链接的文章中正确的内容）

文章已经说的很彻底了，笔者认为，如果说看过rsa加密后自认为解密思路也比较明朗的话，限制人类破解rsa加密的因素，主要是计算机的运行能力。所以才会说，对于计算机而言，找到一种更容易分解质数的算法，才会离破解rsa算法迈进一步。

这里就简单写一写rsa理论概括了。

公钥	(E, N)
私钥	(D, N)
密钥对	(E, D, N)
加密	密文 = 明文 $E \bmod N$ 密文 = 明文 $E \bmod N$
解密	明文 = 密文 $D \bmod N$ 明文 = 密文 $D \bmod N$

求N	$N = p * q$; p, q 为质数
求L	$L = \text{lcm}(p-1, q-1)$; L 为 $p-1, q-1$ 的最小公倍数
求E	$1 < E < L, \text{gcd}(E, L) = 1$; E, L 最大公约数为1 (E 和 L 互质)
求D	$1 < D < L, E * D \bmod L = 1$

通过上面的表格，容易知道，rsa加密紧紧依靠着质数分解。

首先笔者觉得有个比较使用的技术：

手工算小数值rsa的算法

题目意图：

RSA加密实例



加密：

$$19^5 \equiv 66 \pmod{119}, c = 66$$

解密：

$$66^{77} \pmod{119} = ?$$

平方乘算法：

平方乘算法 Square-and-Multiply

$$x \xrightarrow{SQ} x^2 \xrightarrow{MUL} x^3 \xrightarrow{SQ} x^6 \xrightarrow{SQ} x^{12} \xrightarrow{MUL} x^{13} \xrightarrow{SQ} x^{26}$$

https://blog.csdn.net/cried_cat

RSA解密过程

• RSA 算法加密/解密过程：

1. 密钥对 (KU, KR) ：

$$KU = \{e, n\}, KR = \{d, n\}$$

2. 加密过程：把待加密的内容分成 k 比特的分组， $k \leq \log_2 n$ ，并写成数字，设为 M ：

$$C = M^e \pmod{n}$$

3. 解密过程

$$M = C^d \pmod{n}$$

https://blog.csdn.net/cried_cat

比如，我们要加密的数字是十九；

现在我们想办法把密文66的77方还原回19；

而手工的数学关键就是，

还原这个数字。

平方乘算法



1. 将乘方数转换为二进制：

$$77 = (1001101)_2$$

2. 根据每一位为0还是1逐个做处理：

- 1：先平方，再乘 x
- 0：平方

3. 获得如下过程：

77：	Hi	1	0	0	1	1	0	1
	R	x	X ²	X ⁴	X ⁹	X ¹⁹	X ³⁸	X ⁷⁷
	Hi	1	0	0	1	1	0	1
	R	66	72	67	83	94	30	19

我们知道密文，大数，私钥：

数学表达式： $66^77 \bmod 119$

首先还原密钥为二进制： $77 = 0b1001101$

第一位的底数为66；66的平方为4396，然后模119，得到72；

$66^2 \bmod 119 = 72$

接着下一个数字是0，0还是做乘方，我们把72平方得5184，再模119，获得的数就是67；下一个数是1了，1和0的运算不一样，是先乘方再乘x，

所以67的平方再乘66（注意，要乘的不是进阶数，而是基数66），再模119，得到83；

$66^6 \bmod 119 = 83$

；再遇到一，83先做平方，再乘66，模119得到94；最后是0，直接94做平方模119得30；

最后做最后一位1，30方乘66模119最终得到了明文19.

上述是手工方法，接下来是编程思路，详见www.whaledu.com/course/76/task/680/show，个人理解是，把手工的方法编写为程序

Openssl简介



• Openssl中rsa算法用法：

- 指令可用：`genrsa` 生成并输入一个RSA私钥
- `Rsa`：处理RSA密钥的格式转换等问题
- `RSAutl`：使用RSA密钥进行加密、解密、签名和验证等算法

https://blog.csdn.net/cried_cat

• Openssl中rsa解密用法：

- `openssl rsautl -decrypt -in Cipher -inkey privkey -out Plaintext`
- 指定输入密文、私钥和解密后生成文件即可

https://blog.csdn.net/cried_cat

题目解析

- 题目给出了一个压缩文件rsa.zip
- 打开有两个文件发现其中的endata被加密了



- 使用的是zip的伪加密，也就是需要修改标记密码的位。



带有星号的就是加密的文件，要求我们输入密码，可能是伪加密。

一般有两种思路：1：伪加密，不需要真正输入密码进行解密；2：进行爆破：用进行爆破（只针对zip）；

Advanced Archive Password Recovery 这个支持字典自动生成

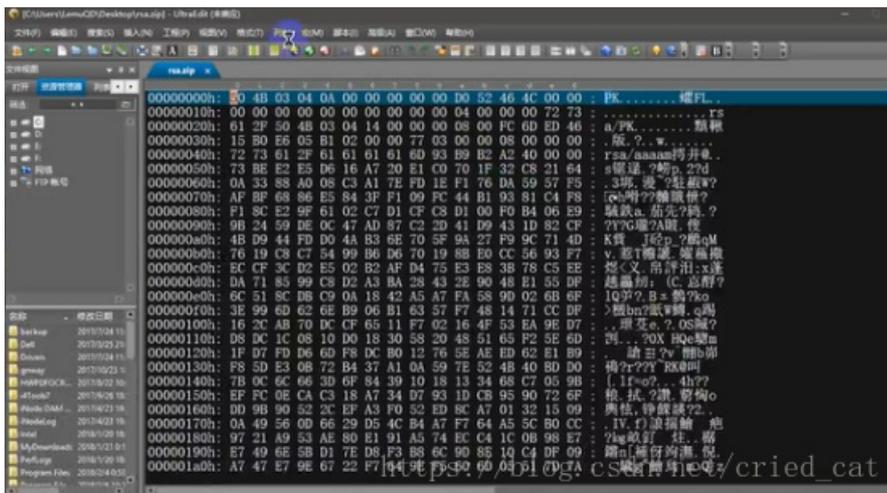
伪加密速度快；

以下介绍两种破解伪加密的算法：

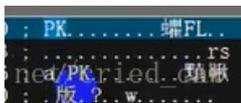
1手动

要掌握文件的

压缩包格式：用winhex来打开文件压缩包



找加密位，



带有pk字段，这就是加密的东西。文件夹，压缩文件本身，两个文件，在这道题中一共是四个pk位；

但是最终找到五个pk

找明显带有0的地方



这里的0014 (0x0014) 代表压缩方式是怎样压缩的

```
00000450h: AA 3A B4 C7 D0 01 50 4B 01 02 1F 00 14 00 09 00 : 7解?PK
00000460h: 08 00 8E 9F F3 4E 58 F3 CB C7 85 00 00 00 80 00 : /e解?PK: cat
00000470h: 00 00 0A 00 24 00 00 00 00 00 00 00 20 00 00 00 : $.
00000480h: 09 02 00 00 72 73 61 2E 65 6E 64 61 74 61 0A 00 : ? rsa/edata
```

我们要改动的就是0014后面的两个00; (之前的pk后是两个00)
这里表示了伪加密的位置, 把09 (加密) 变成00就可以了, 保存。
生成新文件, 我们可以发现星号不见了, 解压成功。

(百度: 文件头格式)

2

- 两个方法修改:
 - 使用ZipCenOp自动修复, 将ZipCenOp和压缩包放在同一目录下。
 - 打开cmd命令, 输入java -jar ZipCenOp.jar r rsa.zip

```
D:\课程平台\CTF打卡\第一期\day3>java -jar ZipCenOp.jar r rsa.zip
success 3 flag(s) found
```

- 即可解开伪加密。
- 解压得到两个文件 http://blog.csdn.net/cried_cat

把zip文件放到zipcenop的同目录下即可, 输入命令

```
D:\课程平台\CTF打卡\第一期\day3>java -jar ZipCenOp.jar r rsa.zip
```

回车, 注意备份, 这回直接修改文件

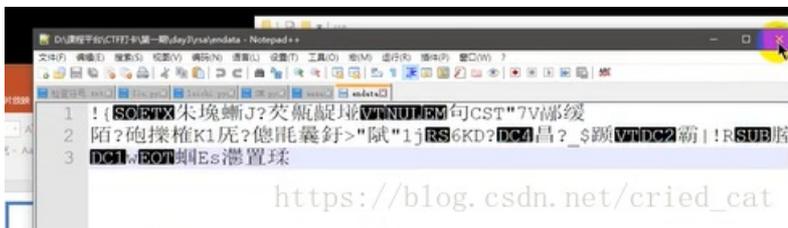
```
D:\课程平台\CTF打卡\第一期\day3>java -jar ZipCenOp.jar r rsa.zip
success 3 flag(s) found
D:\课程平台\CTF打卡\第一期\day3>
```

获取三个加密标志位, 已去除加密标志。

把aaa打开,

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQQA4GKtDqB1D2hTQVm9oMyp7w3C8i4tCE0zsWTWtV0gVaoi
idBeZR3AX/crKn1X1QC6YD/pUJJRPzoKY+bk0YFT64mca7oW2fP719LFiQRei
fs4n9mvIYeBx1TtHfFwWBDeIHcjP33eR1hue69Dq5tZYR12e3SrUJXvRFwIDf
AoGAMUQcF1KdHOv5wkweXg/4eIpJHJe2nuLkgL26P5FD9D/1r9ZAsKNKmo/Vc
8fDRfQoBOueFwJAZ8qRUsWCT+/0ZDs0xCKrQ7YuxO2p9HV1sMQF74D2TVcoI
P5sjTvs0MutaoTdU0YDNO/ssqk3We2e11tr6ii3HmHtquwkCQQDezreUOIjWl
ndoGwYf9LsXoEZVMSI6vw/SqiqOvagN3mufApNfj+JrZ6LvV0hHbYfaVKEUvc
BKAQapNFakeAyz2RlugQ20fVzUkzDCSF8ByWjK4GgAqQ/qioXJ9tSPcsgV1yl
WM7rTBDWaohHT3N+vhAcsszQ2VJZy6vKqWJBAL2liH7CLD79Uwswgg70FfM8k
1UfMDp+vFIdA4JiDjRX2JUNFTHm/9tZ6Eb+rQgXQ+ZlOpoUtKz85tqCihl0Ck
R16MyChIRRR/LMizVPer6dkJJWff97LebL150cxwzcpQtet2svTDIRLiJ3H
QWsq6hudCk3tNrRQqb8CQCCTcs0uWBe6klDKWLCPEYxuTqB9xksQtm1qvdfc
BWaxHtc/ByfAisj9cfq2CY/fEoeGqLagZ5tG5G81V9VZ
-----END RSA PRIVATE KEY-----
og.csdn.net/cried_cat
```

这是rsa私钥文件



这是密文

接下来，我们用openssl解密

再kali中输入：

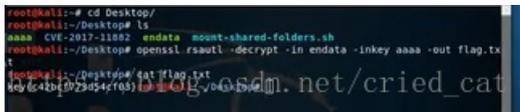


则，再当前文件夹下生成了解密后的文件。



把题库的文件拖进kali中，，

其实和上二图中的命令一样；



视频中用cat命令读取了解密文件内容

嗯，以上主要是是rsa手工解密的算法了。

密钥生成

题目链接：

<http://daka.whaledu.com:9999/challenges#密钥生成>

Challenge

54 Solves



密钥生成

4

在一次RSA密钥对生成中, 假设 $p=473398607161$, $q=4511491$,
 $e=17$ 求解出 d ,格式: `key{d}`

Flag

Submit

https://blog.csdn.net/cried_cat

用rsa-tool算算就知道答案了;

Challenge

38 Solves

RSA解密

4

加密方式好假, rsa直接解密吧~

📄 rsa.zip

Flag

https://blog.csdn.net/cried_cat

直接把本篇writeup上文的操作转化为行动就出来了题, 塔主亲讲过;

咳咳, 本周时间仓促, 未能做出几道rsa的题目。只能下周来上交解出rsa类的题目来弥补此周的缺陷。希望下周的writeup的rsa题目总结可以弥补此周写作的质量不佳。

