

CTF密码学--新手题--Normal_RSA--解题过程及总结

原创

hippotomons 于 2019-10-21 22:05:09 发布 6089 收藏 23

文章标签: [CTF 密码学](#) [RSA](#) [rsatool](#) [openssl](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/hippotomons/article/details/102672851>

版权



Normal_RSA

难度系数: ☆

题目来源: PCTF

题目描述: 你和小鱼走啊走走啊走, 走到下一个题目一看你又一愣, 怎么还是一个数学题啊 小鱼又一笑, hhhh数学在密码学里面很重要的! 现在知道吃亏了吧! 你哼一声不服气, 我知道数学 很重要了! 但是工具也很重要, 你看我拿工具把他解出来! 你打开电脑折腾了一会还真的把答案 做了出来, 小鱼有些吃惊, 向你投过来一个赞叹的目光

附件下载下来又是一阵懵逼, 一个enc文件, 一个是pem文件, 根据名字可以猜到一个是flag的密文, 一个是公钥 (pubkey), 但是这两种文件格式咱都没见过啊。。

 flag.enc	2016/4/29 17:56	PSENC File	1 KB
 pubkey.pem	2016/4/29 17:19	PEM 文件	1 KB

flag.enc打不开, pubkey.pem倒是可以打开, 里面也明确写了这是个public key, 而且中间有一段base64的编码

```
1 -----BEGIN PUBLIC KEY-----
2 MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMJjauXD2OQ/+5erCQKPGqxsC/bNPXDr
3 yigb/+l/vjDdAgMBAE=
4 -----END PUBLIC KEY-----
5
```

但是解码出来也是意义不明的乱码

```
PS D:\Note\Program_code\py> python -u "d:\Note\Program_code\py\demo.py"
+ 0(! ĀcjĀĀöā?ū« -lōī=pēĒ(ē%0Ÿ
                   öī=pēĒ(ē%0Ÿ
```

于是百度之

这个时候度娘就开始误导我了

误导1:

后缀ENC是什么文件

后缀ENC是什么文件

 我来答

 分享

 举报

浏览 11956 次

4个回答

#校园# 有一个矫情的舍友是什么样的体验?



jsntr88

来自电脑网络类芝麻团 推荐于2016-08-22

ENC格式是使用“Encore”软件制作的文件。可以使用Adobe Encore打开。

Adobe Encore 过去曾作为一款完全独立的软件存在，但从 CS3 开始，Adobe 将其划归 Premiere Pro 的附属组件，因为取消了 Premiere Pro 2.0 时代的 DVD 编码、设计与刻录集成，Encore 已成为 Premiere 必不可少的一个输出组件，但其更为专业与完善的设计功能，相对更独立的架构，又使其仍可以单独运行。Encore 更像一款为了 Premiere Pro 最终出版视频产品的打包终端，其支持硬件刻录的功能有着明显的物理意义特性。这也是目前为止，它与 Media Encoder 的一个很明显的区别。

👍 抢首赞



💬 评论

🔄 分享

🚩 举报

<https://blog.csdn.net/hippolomons>

神™midi乐谱文件

误导2:

视频加密后是enc格式怎么破解

👤 我来答

🔄 分享

🚩 举报

浏览 2759 次

2个回答

#热议# 本科应届生真的人均都月薪过万吗?



迪丽热巴like

来自电脑网络类芝麻团 2016-01-15

enc文件是一款叫神盾加密软件加密的文件。

具体网络资料:

<http://zhidao.baidu.com/question/377266003.html>

<http://zhidao.baidu.com/question/494896697.html>

我之前用的是文件夹保护3000，可以对文件夹进行加密码、隐藏和伪装保护，方便快捷地帮您解决重要文件夹的保密问题。

软件保密性好。文件夹加密码后，打开文件夹要输入正确密码，而且在任何环境下均不失效。文件夹隐藏后，在任何环境下不通过本软件无法找到。文件夹伪装后就变成了伪装的对象，即便打开也看不到文件夹里原有的文件。

软件对文件夹加密码、隐藏或伪装时速度特别快，无论文件夹大小。软件采用的是成熟、优秀的**数据保护技术**，安全性高。加密码文件夹使用完毕后，会自动恢复到加密码状态，无须再次加密码。

误导2让我去下了一个神盾xxxx的软件，下载下来之后发现这玩意貌似只能生成一个加密的磁盘分区，于是又去百度pem是啥，这次度娘终于靠谱了

PEM是OpenSSL和许多其他SSL工具的标准格式，OpenSSL 使用PEM 文件格式存储证书和密钥。这种格式被设计用来安全的包含在asci甚至富文本文档中，如电子邮件。这意味着您可以简单的复制和粘贴pem文件的内容到另一个文档中。

PEM文件是Base64编码的证书。PEM证书通常用于web服务器，因为他们可以通过一个简单的文本编辑器，很容易地转换成可读的数据。通常当一个PEM编码在文本编辑器中打开文件,它会包含不同的页眉和页脚。

于是去下了个openssl，然后从pubkey.pem中提取信息

```

C:\Users\Lenovo\Desktop\5a8da6f6bea6b4ec79884e9a736774d77
λ openssl
OpenSSL> rsa -pubin -text -modulus -in warmup -in pubkey.pem
RSA Public-Key: (256 bit)
Modulus:
  00:c2:63:6a:e5:c3:d8:e4:3f:fb:97:ab:09:02:8f:
  1a:ac:6c:0b:f6:cd:3d:70:eb:ca:28:1b:ff:e9:7f:
  be:30:dd
Exponent: 65537 (0x10001)
Modulus=C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD
writing RSA key
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMJjauXD20Q/+5erCQKPGqxsC/bNPXDr
yigb/+1/vjDdAgMBAAE=
-----END PUBLIC KEY-----
OpenSSL>

```

https://blog.csdn.net/hippotomons

里面modulus就是N，exponent就是e

这里显示

Modulus (hex)

=C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD

明显是个16进制数，先转换成10进制的

Modulus (dec)

=87924348264132406875276140514499937145050893665602592992418171647042491658461

对其在线质因数分解，得到p、q

Result:		
status (?)	digits	number
FF	77 (show)	8792434826...61<77> = 275127860351348928173285174381581152299<39> · 319576316814478949870590164193048041239<39>

所以现在知道了

p=275127860351348928173285174381581152299

q=319576316814478949870590164193048041239

e=65537

然后，事情就陷入了僵局。。之后看了一下writeup，

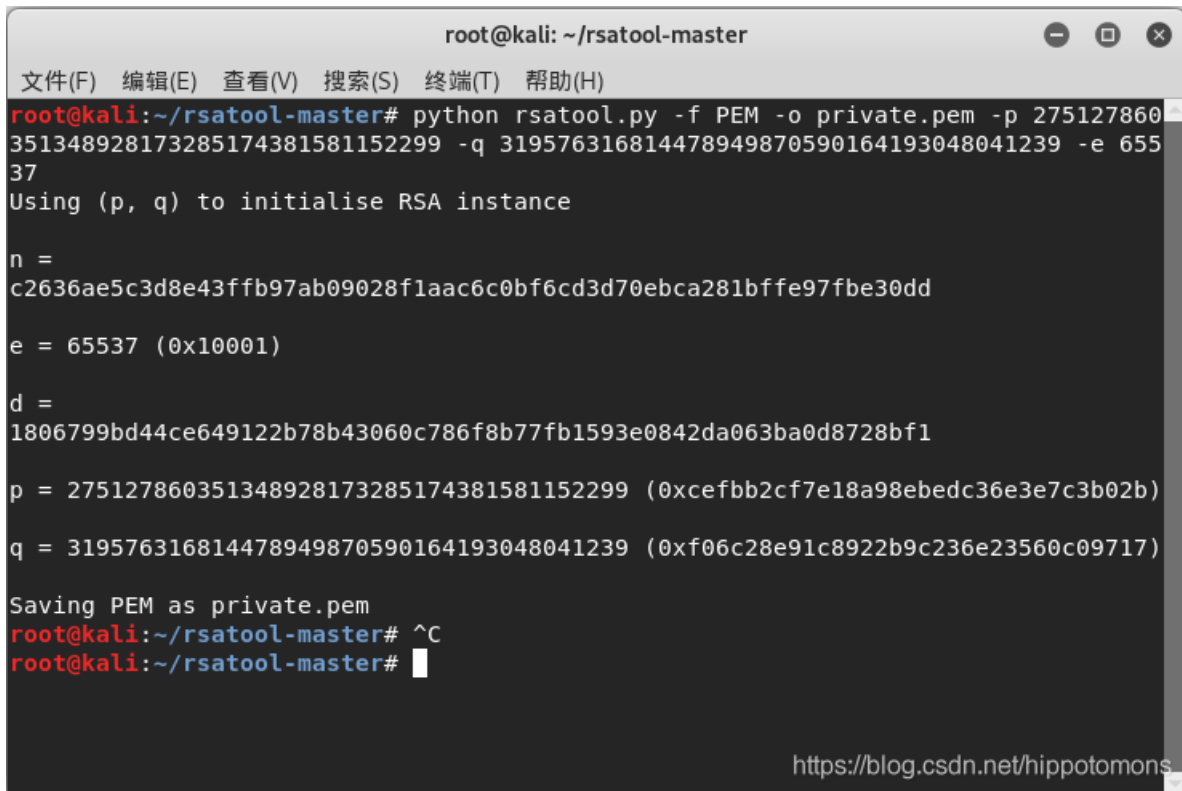
发现是用了一个叫rsatool的工具，可以由p、q、e计算d，并生成pem文件

然后就是想办法去安装rsatool，但是发现安装它还需要安俩东西，pyasn1和gmpy，其中pyasn1顺利安上了，但是gmpy死活安不上，鼓捣了3个晚上，发现好像gmpy太老了，不兼容现在的各种东西，但是好像在kali上能装成，于是按照教程，成功在kali上装上了

然后按照用法，把pubkey.pem和rsatool.py放一块，从终端输入命令：

```
python rsatool.py -f PEM -o private.pem -p 275127860351348928173285174381581152299 -q 319576316814478949870590164193048041239 -e 65537
```

成功生成private.pem文件



```
root@kali: ~/rsatool-master
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali:~/rsatool-master# python rsatool.py -f PEM -o private.pem -p 275127860351348928173285174381581152299 -q 319576316814478949870590164193048041239 -e 65537
Using (p, q) to initialise RSA instance
n =
c2636ae5c3d8e43ffb97ab09028f1aac6c0bf6cd3d70ebca281bffe97f30dd
e = 65537 (0x10001)
d =
1806799bd44ce649122b78b43060c786f8b77fb1593e0842da063ba0d8728bf1
p = 275127860351348928173285174381581152299 (0xcefbb2cf7e18a98ebcdc36e3e7c3b02b)
q = 319576316814478949870590164193048041239 (0xf06c28e91c8922b9c236e23560c09717)
Saving PEM as private.pem
root@kali:~/rsatool-master# ^C
root@kali:~/rsatool-master#
```

然后把private.pem拖回Windows，和pubkey.pem、flag.enc放一个文件夹里，打开cmd，使用openssl用private.pem解密flag.enc文件并将明文生成txt文件

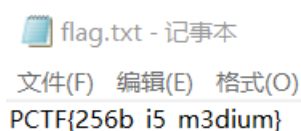
输入命令：

```
rsautl -decrypt -in flag.enc -inkey private.pem -out flag.txt
```

成功生成flag.txt

flag.enc	2016/4/29 17:56	PSENC File	1 KB
flag.txt	2019/10/19 21:27	文本文档	1 KB
private.pem	2019/10/19 21:18	PEM 文件	1 KB
pubkey.pem	2016/4/29 17:19	PEM 文件	1 KB

打开，得到flag



```
flag.txt - 记事本
文件(F) 编辑(E) 格式(O)
PCTF{256b_i5_m3dium}
```

收获：
Kali牛批

