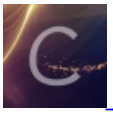
攻防世界-新手练习-Normal_RSA

原创

根本不是咖啡猫 于 2021-07-21 11:05:49 发布 388 收藏 3
分类专栏： 攻防答题 文章标签： 安全



【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！但是工具也很重要的，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】

两个文件，一个flag.enc，一个pubkey.pem。

根据题目，两文件应该是rsa的公钥加密过程。

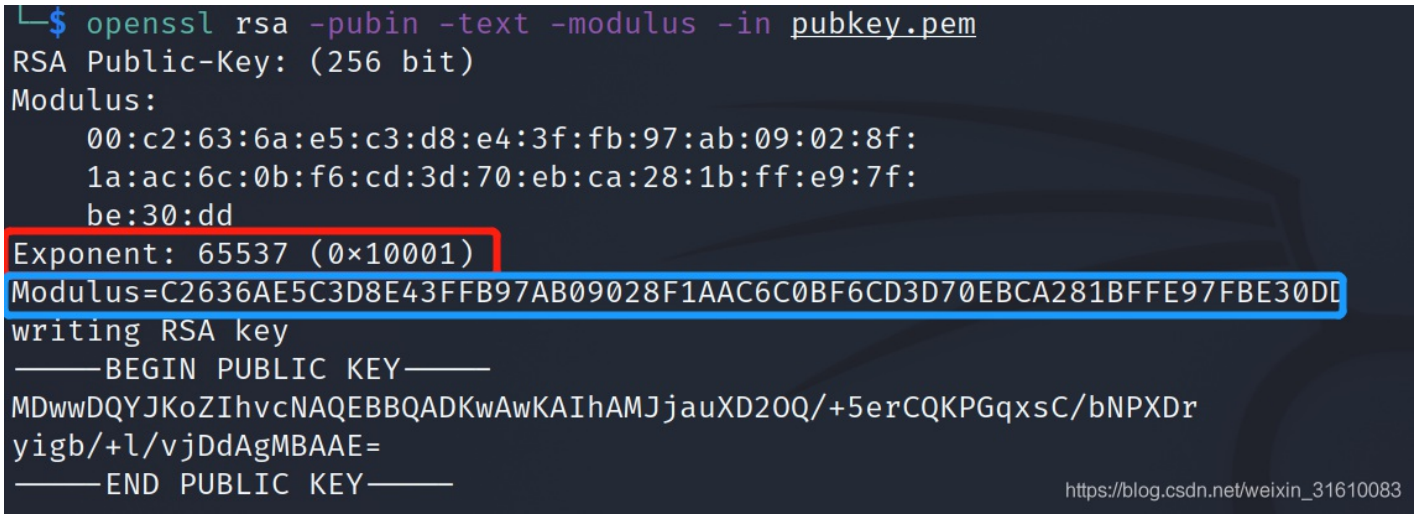
【思路】

根据公钥密码学知识，公私钥对中，私钥用于加密，公钥用于解密，这里给了pubkey即公钥，则flag.enc应是密文，使用工具[rsatool]对密文进行解密。

1. 用openssl从公钥中提取e和modulus（即大素数）的值。

```
└─$ openssl rsa -help
Usage: rsa [options]
Valid options are:
 -help               Display this summary
 -inform format      Input format, one of DER PEM
 -outform format     Output format, one of DER PEM PVK
 -in val             Input file
 -out outfile        Output file
 -pubin              Expect a public key in input file   //指定输入文件为公钥
 -pubout             Output a public key
 -passout val        Output file pass phrase source
 -passin val         Input file pass phrase source
 -RSAPublicKey_in    Input is an RSAPublicKey
 -RSAPublicKey_out   Output is an RSAPublicKey
 -noout              Don't print key out
 -text               Print the key in text  //以明文形式输出各参数
 -modulus            Print the RSA key modulus  //输出模数值
 -check              Verify key consistency
 -*                  Any supported cipher
 -pvk-strong         Enable 'Strong' PVK encoding level (default)
 -pvk-weak           Enable 'Weak' PVK encoding level
 -pvk-none           Don't enforce PVK encoding
 -engine val         Use engine, possibly a hardware device
```

命令：openssl rsa -pubin -text -modulus -in pubkey.pem



2. 参数处理

上一步中得modulus(十六进制)=

C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD

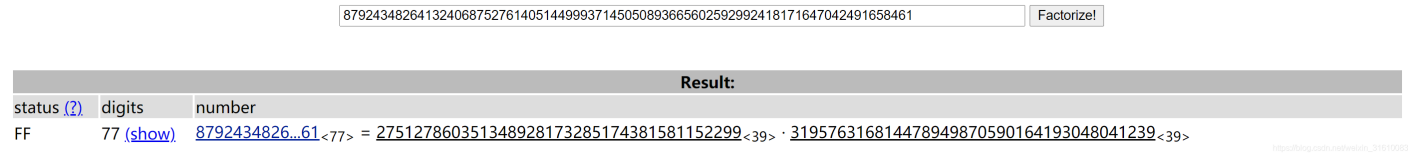
a.进制转换

把模数转换为10进制，得到：modulus(十进制)=

879243482641324068752761405144999371450508936656025929924181716470424916584 61

b.素数分解

在线大数分解工具：http://www.factordb.com/



所以，目前我们有参数：

p=275127860351348928173285174381581152299

q=319576316814478949870590164193048041239

e=65537

3. 生成私钥

在rsatool的目录下，

安装rsatool的问题，参考 https://blog.csdn.net/jcbx_/article/details/97250664

命令：

```
python3 rsatool.py -o prikey.pem -e 65537 -p 275127860351348928173285174381581152299 -q 3195763168144789498
```

```
└─$ python3 rsatool.py -o prikey.pem -e 65537 -p 275127860351348928173285174381581152299
 -q 319576316814478949870590164193048041239
Using (p, q) to initialise RSA instance

n =
c2636ae5c3d8e43ffb97ab09028f1aac6c0bf6cd3d70ebca281bffe97fbe30dd

e = 65537 (0x10001)

d =
1806799bd44ce649122b78b43060c786f8b77fb1593e0842da063ba0d8728bf1

p = 275127860351348928173285174381581152299 (0xcefbb2cf7e18a98ebedc36e3e7c3b02b)

q = 319576316814478949870590164193048041239 (0xf06c28e91c8922b9c236e23560c09717)

Saving PEM as prikey.pem
```


私钥生成在当前目录下（即rsatool目录）。

1. 用私钥解密flag

命令：openssl rsautl -decrypt -in flag.enc -inkey prikey.pem

```
└─$ openssl rsautl -decrypt -in flag.enc -inkey prikey.pem
PCTF{256b_i5_m3dium}
```

【答案】

PCTF{256b_i5_m3dium}

【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！但是工具也很重要的，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】

两个文件，一个flag.enc，一个pubkey.pem。

根据题目，两文件应该是rsa的公钥加密过程。

【思路】

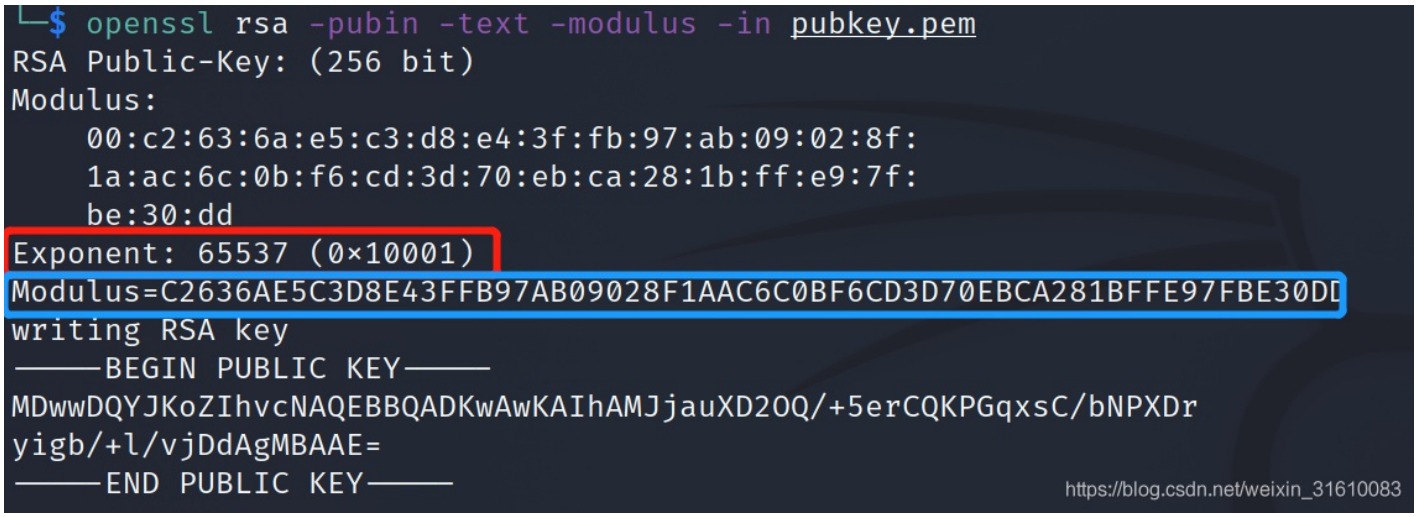
根据公钥密码学知识，公私钥对中，私钥用于加密，公钥用于解密，这里给了pubkey即公钥，则flag.enc应是密文，使用工具[rsatool]对密文进行解密。

1. 用openssl从公钥中提取e和modulus（即大素数）的值。

```
└─$ openssl rsa -help
Usage: rsa [options]
Valid options are:
 -help               Display this summary
 -inform format      Input format, one of DER PEM
 -outform format     Output format, one of DER PEM PVK
 -in val             Input file
 -out outfile        Output file
 -pubin              Expect a public key in input file   //指定输入文件为公钥
 -pubout             Output a public key
 -passout val        Output file pass phrase source
 -passin val         Input file pass phrase source
 -RSAPublicKey_in    Input is an RSAPublicKey
 -RSAPublicKey_out   Output is an RSAPublicKey
 -noout              Don't print key out
 -text               Print the key in text  //以明文形式输出各参数
 -modulus            Print the RSA key modulus  //输出模数值
 -check              Verify key consistency
 -*                  Any supported cipher
 -pvk-strong         Enable 'Strong' PVK encoding level (default)
 -pvk-weak           Enable 'Weak' PVK encoding level
 -pvk-none           Don't enforce PVK encoding
 -engine val         Use engine, possibly a hardware device
```

命令：

```
openssl rsa -pubin -text -modulus -in pubkey.pem
```



2. 参数处理

上一步中得modulus(十六进制)=

C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD

a. 进制转换

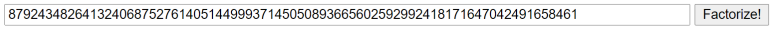
把模数转换为10进制，得到：

modulus(十进制)=

87924348264132406875276140514499937145050893665602592992418171647042491658461

b. 素数分解

在线大数分解工具：http://www.factordb.com/



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

所以，目前我们有参数：

p=275127860351348928173285174381581152299

q=319576316814478949870590164193048041239

e=65537

3/ 生成私钥

在rsatool的目录下，输入参数生成私钥。

（安装rsatool的问题，参考 https://blog.csdn.net/jcbx_/article/details/97250664）

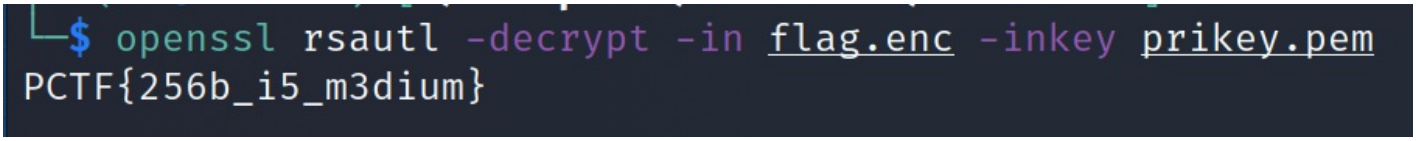
```
python3 rsatool.py -o prikey.pem -e 65537 -p 275127860351348928173285174381581152299 -q 3195763168144789498
```

私钥生成在当前目录下（即rsatool目录）。

4. 用私钥解密flag

命令：

```
openssl rsautl -decrypt -in flag.enc -inkey prikey.pem
```



【答案】

PCTF{256b_i5_m3dium}

如果文章对你有帮助，麻烦动动手指点赞、喜欢、支持一下咖啡猫，谢谢！