BUUCTF Crypto RSA & what writeup

原创

Slightwindsec 于 2020-04-18 02:32:35 发布 1717 收藏 1

分类专栏： CTF 文章标签： 密码学 python base64

本文链接：https://blog.csdn.net/qq_41956187/article/details/105592471

我的博客：https://blog.slight-wind.com/

RSA & what writeup

RSA共模攻击 + Base64隐写

在 buu 刷到的一题，看到 N 用了两次，但 RSA 共模攻击解完发现还没结束…

```python
from Crypto.Util.number import*
import base64

def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

def CMA(n,e1,e2,c1,c2):
    s = egcd(e1, e2)
    s1 = s[1]
    s2 = s[2]
    if s1<0:
        s1 = - s1
        c1 = inverse(c1, n)
    elif s2<0:
        s2 = - s2
        c2 = inverse(c2, n)
    m = pow(c1,s1,n)*pow(c2,s2,n) % n
    return m

f1=open("HUB1")
f2=open("HUB2")
N=f1.readline()
N=f2.readline()
e1,e2=f1.readline(),f2.readline()
f1.readline()
f2.readline()
c1,c2=f1.readline(),f2.readline()
ans=b''
cnt=0
while len(c1)!=0:
    cnt+=1
    ans+=long_to_bytes(CMA(int(N),int(e1),int(e2),int(c1),int(c2)))
    #print(base64.b64decode(temp))
    c1,c2=f1.readline(),f2.readline()
temp=b''
M=b''
print(ans)
for i in ans:
    k=long_to_bytes(i)
    #print(i," ",end="")
    if k==b'\n':
        M+=base64.b64decode(temp)
        temp=b''
        continue
    temp+=k
print(M)
```

到这里可以解出来 base64 编码和解码后的明文。

```
b'VEhJUz==\nRkxBR3==\nSVN=\nSElEREVOLo==\nQ0FO\nWU9V\nRklORM==\nSVT=\nT1VUP4==\nRE8=\nWU9V\nS05PV9==\nQkFTRTY0P5
==\nWW91bmdD\nVEhJTku=\nWU9V\nQVJF\nTk9U\nVEhBVE==\nRkFNSUxJQVI=\nV0lUSO==\nQkFTRTY0Lh==\nQmFzZTY0\naXO=\nYW==\n
Z3JvdXA=\nb2b=\nc2ltaWxhcn==\nYmluYXJ5LXRvLXRleHR=\nZW5jb2Rpbme=\nc2NoZW1lc0==\ndGhhdD==\ncmVwcmVzZW50\nYmluYXJ5
\nZGF0YW==\naW5=\nYW6=\nQVNDSUl=\nc3RyaW5n\nZm9ybWF0\nYnk=\ndHJhbnNsYXRpbmd=\naXS=\naW50b1==\nYT==\ncmFkaXgtNjQ=
\ncmVwcmVzZW50YXRpb24u\nVGhl\ndGVybc==\nQmFzZTY0\nb3JpZ2luYXRlc8==\nZnJvbd==\nYY==\nc3BlY2lmaWN=\nTUlNRT==\nY29u
dGVudI==\ndHJhbnNmZXI=\nZW5jb2Rpbmcu\nVGhl\ncGFydGljdWxhct==\nc2V0\nb2b=\nNjR=\nY2hhcmFjdGVyc5==\nY2hvc2Vu\ndG+=
\ncmVwcmVzZW50\ndGhl\nNjQ=\ncGxhY2UtdmFsdWVz\nZm9y\ndGhl\nYmFzZd==\ndmFyaWVz\nYmV0d2Vlbt==\naW1wbGVtZW50YXRpb25z
Lp==\nVGhl\nZ2VuZXJhbI==\nc3RyYXRlZ3n=\naXO=\ndG9=\nY2hvb3Nl\nNjR=\nY2hhcmFjdGVyc5==\ndGhhdA==\nYXJl\nYm90aN==\n
bWVtYmVyc5==\nb2a=\nYS==\nc3Vic2V0\nY29tbW9u\ndG8=\nbW9zdM==\nZW5jb2RpbmdzLA==\nYW5k\nYWxzb8==\ncHJpbnRhYmxlLg==
\nVGhpc9==\nY29tYmluYXRpb25=\nbGVhdmVz\ndGhl\nZGF0YW==\ndW5saWtlbHk=\ndG/=\nYmV=\nbW9kaWZpZWS=\naW5=\ndHJhbnNpdE
==\ndGhyb3VnaN==\naW5mb3JtYXRpb26=\nc3lzdGVtcyw=\nc3VjaN==\nYXM=\nRS1tYWlsLD==\ndGhhdA==\nd2VyZQ==\ndHJhZGl0aW9u
YWxseQ==\nbm90\nOC1iaXQ=\nY2xlYW4uWzFd\nRm9y\nZXhhbXBsZSw=\nTUlNRSdz\nQmFzZTY0\naW1wbGVtZW50YXRpb24=\ndXNlcw==\n
QahDWiw=\nYahDeiw=\nYW5k\nMKhDOQ==\nZm9y\ndGhl\nZmlyc3Q=\nNjI=\ndmFsdWVzLg==\nT3RoZXI=\ndmFyaWF0aW9ucw==\nc2hhcm
U=\ndGhpcw==\ncHJvcGVydHk=\nYnV0\nZGlmZmVy\naW4=\ndGhl\nc3ltYm9scw==\nY2hvc2Vu\nZm9y\ndGhl\nbGFzdA==\ndHdv\ndmFs
dWVzOw==\nYW4=\nZXhhbXBsZQ==\naXM=\nVVRGLTcu'
```

解码后的明文：（出题人科普了一遍 base64？？）

> THIS FLAG IS HIDDEN.
> CAN YOU FIND IT OUT?
> DO YOU KNOW BASE64?
> Young C THINK YOU ARE NOT THAT FAMILIAR WITH BASE64.
> Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation.
> The term Base64 originates from a specific MIME content transfer encoding.
> The particular set of 64 characters chosen to represent the 64 place-values for the base varies between implementations.
> The general strategy is to choose 64 character sthatare both members of a subset common to most encodings,
> and also printable.
> This combination leaves the data unlikely to be mod if iedintransit through in formation systems,
> such as E-mail,that were tradition all ynot 8-bit clean.
> [1]Forexample,MIME's Base64 implement ation uses A\xa8CZ,a\xa8Cz,and 0\xa8C9 for the first 62 values.
> Other variations share this property butdiffer in the symbols chosen for the last two values;an example is UTF-7.

明文里显然不能获得更多信息了，只能看那段 base64 编码，它的特别之处在于分了很多很多小段，不难想到（之前做过）base64隐写。

于是在网上嫖一段 base64 隐写脚本改一改用了。

exp：

```
from Crypto.Util.number import*
import base64
c = b'VEhJUz==\nRkxBR3==\nSVN=\nSElEREVOLo==\nQ0FO\nWU9V\nRklORM==\nSVT=\nT1VUP4==\nRE8=\nWU9V\nS05PV9==\nQkFTRTY0P5==\nWW91bmdD\nVEhJTku=\nWU9V\nQVJF\nTk9U\nVEhBVE==\nRkFNSUxJQVI=\nV0lUSO==\nQkFTRTY0Lh==\nQmFzZTY0\naXO=\nYW==\nZ3JvdXA=\nb2b=\nc2ltaWxhcn==\nYmluYXJ5LXRvLXRleHR=\nZW5jb2Rpbme=\nc2NoZW1lc0==\ndGhhdD==\ncmVwcmVzZW50\nYmluYXJ5\nZGF0YW==\naW5=\nYW6=\nQVNDSUl=\nc3RyaW5n\nZm9ybWF0\nYnk=\ndHJhbnNsYXRpbmd=\naXS=\naW50b1==\nYT==\ncmFkaXgtNjQ=\ncmVwcmVzZW50YXRpb24u\nVGhl\ndGVybc==\nQmFzZTY0\nb3JpZ2luYXRlc8==\nZnJvbd==\nYY==\nc3BlY2lmaWN=\nTUlNRT==\nY29udGVudI==\ndHJhbnNmZXI=\nZW5jb2Rpbmcu\nVGhl\ncGFydGljdWxhct==\nc2V0\nb2b=\nNjR=\nY2hhcmFjdGVyc5==\nY2hvc2Vu\ndG+=\ncmVwcmVzZW50\ndGhl\nNjQ=\ncGxhY2UtdmFsdWVz\nZm9y\ndGhl\nYmFzZd==\ndmFyaWVz\nYmV0d2Vlbt==\naW1wbGVtZW50YXRpb25zLp==\nVGhl\nZ2VuZXJhbI==\nc3RyYXRlZ3n=\naXO=\ndG9=\nY2hvb3Nl\nNjR=\nY2hhcmFjdGVyc5==\ndGhhdA==\nYXJl\nYm90aN==\nbWVtYmVyc5==\nb2a=\nYS==\nc3Vic2V0\nY29tbW9u\ndG8=\nbW9zdM==\nZW5jb2RpbmdzLA==\nYW5k\nYWxzb8==\ncHJpbnRhYmxlLg==\nVGhpc9==\nY29tYmluYXRpb25=\nbGVhdmVz\ndGhl\nZGF0YW==\ndW5saWtlbHk=\ndG/=\nYmV=\nbW9kaWZpZWS=\naW5=\ndHJhbnNpdE==\ndGhyb3VnaN==\naW5mb3JtYXRpb26=\nc3lzdGVtcyw=\nc3VjaN==\nYXM=\nRS1tYWlsLD==\ndGhhdA==\nd2VyZQ==\ndHJhZGl0aW9uYWxseQ==\nbm90\nOC1iaXQ=\nY2xlYW4uWzFd\nRm9y\nZXhhbXBsZSw=\nTUlNRSdz\nQmFzZTY0\naW1wbGVtZW50YXRpb24=\ndXNlcw==\nQahDWiw=\nYahDeiw=\nYW5k\nMKhDOQ==\nZm9y\ndGhl\nZmlyc3Q=\nNjI=\ndmFsdWVzLg==\nT3RoZXI=\ndmFyaWF0aW9ucw==\nc2hhcmU=\ndGhpcw==\ncHJvcGVydHk=\nYnV0\nZGlmZmVy\naW4=\ndGhl\nc3ltYm9scw==\nY2hvc2Vu\nZm9y\ndGhl\nbGFzdA==\ndHdv\ndmFsdWVzOw==\nYW4=\nZXhhbXBsZQ==\naXM=\nVVRGLTcu'

def get_base64_diff_value(s1, s2):
    base64chars = b'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    res = 0
    for i in range(len(s2)):
        if s1[i] != s2[i]:
            return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
    return res

def solve_stego():
    line=b''
    bin_str=''
    for i in c:
        k=long_to_bytes(i)
        if k==b'\n':
            steg_line = line
            norm_line = base64.b64encode(base64.b64decode(line))
            diff = get_base64_diff_value(steg_line, norm_line)
            #print(diff)
            pads_num = steg_line.count(b'=')
            if diff:
                bin_str += bin(diff)[2:].zfill(pads_num * 2)
            else:
                bin_str += '0' * pads_num * 2
            print(goflag(bin_str))
            line=b''
            continue
        line+=k

def goflag(bin_str):
    res_str = ''
    for i in range(0, len(bin_str), 8):
        res_str += chr(int(bin_str[i:i + 8], 2))
    return res_str


if __name__ == '__main__':
    solve_stego()
```

最终得到字符串：7c86d8f7d6de33a87f7f9d6b005ce640 套上 flag{} 就可以了。