实验吧——WriteUP&&涨姿势（6）

原创

浅零半泣 于 2017-06-14 12:32:25 发布 1266 收藏 1

分类专栏： CTF

本文链接：https://blog.csdn.net/sinat_34200786/article/details/73214450

WriteUp

1. RSA实践
2. 一串奇怪的数
3. 兔子你好
4. NSCTF crypto200

涨姿势

1. RSAROLL
2. 凯撒和某某加密

RSA实践

原题

在一次RSA密钥对生成中，假设p=473398607161，q=4511491，e=17
求解出d
将得到的d提交

解题思路

爆破爆破爆破

WriteUp

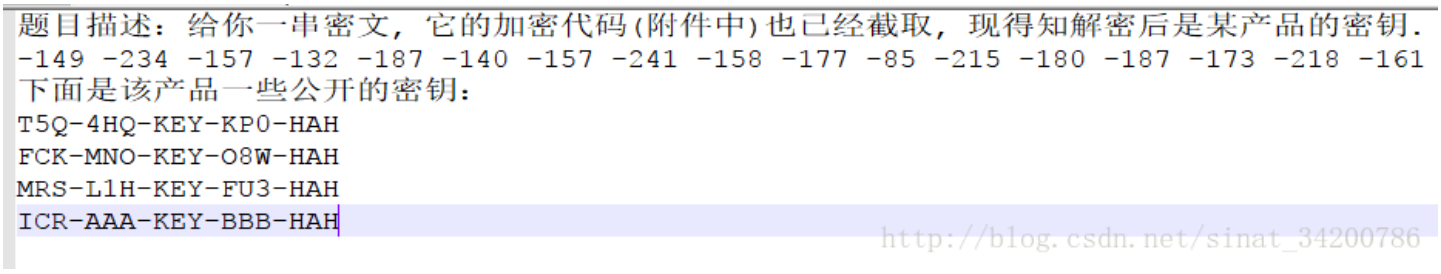
```cpp
#include<iostream>

long long int get(long long int a,long long int b,long long int &x,long long int &y);
int main()
{
    long long int  a,b,x,y;
    a = 473398607160;
    b = 4511490;
    get(17,a*b,x,y);
    std::cout<<x<<std::endl;
    return 0;
}

//扩展欧几里得算法
//返回a,b的最大公约数并求x,y使满足 ax + by = gcd(a,b)
long long int get(long long int a,long long int b,long long int &x,long long int &y)
{
    if(b == 0)
    {
        x = 1;
        y = 0;
        return a;
    }
    else
    {
        long long int r = get(b,a%b,y,x);
        y -= x*(a/b);
        return r;
    }
}
```

一串奇怪的数

原题



解题思路

```
代码审计，逆向解密
```

WriteUp

```
首先审计加密代码，找出加密规律
```

```python
#coding:utf-8     #加密代码

import hashlib

def md5(s):
    return hashlib.md5(s).hexdigest()

def evalCrossTotal(strMd5):
    r = 0
    for i in strMd5:
        r += int("0x%s" % i, 16)
    return r

def encryptString(strString, strPasswd):
    strPasswdMd5 = md5(strPasswd)
    intMd5       = evalCrossTotal(strPasswdMd5)

    r = []

    for i in range(len(strString)):
        r.append(
            ord(strString[i]) + \
            int("0x%s" % strPasswdMd5[i%32], 16) - \
            intMd5
        )
        intMd5 = evalCrossTotal(
            md5(strString[:(i+1)])[:16] + \
            md5(str(intMd5))[:16]
        )
    return " ".join(map(lambda x: str(x), r))
```

发现：

1. 如果strPassword已知则可确定第一个intMD5
2. 已知第一个intMD5可由第一个密文逆推第一个明文
3. 已知第一个明文和第一个intMD5可推出第二个intMD5
4. 已知第二个intMD5可由第二个密文逆推出第二个明文
5. 已知第N个明文和第N个intMD5可推出第N+1个intMD5
6. ...

关键在于strPassword是多少？我猜是空字符串你信不信？

```
import hashlib

def md5(s):
    return hashlib.md5(s.encode('utf-8')).hexdigest()

def evalCrossTotal(strMd5):
    r = 0
    for i in strMd5:
        r += int("0x%s" % i, 16)
    return r

key = ''   #关键在于密码为空
md5key = md5(key)
total = evalCrossTotal(md5key)
flag = ''

with open('miwen.txt','r') as mi:
    ss = mi.readline()
    ls = ss.split(' ')
    for n in range(len(ls)):
        flag += chr(int(ls[n]) + total - int("0x%s" % md5key[n%32], 16))
        total = evalCrossTotal(md5(flag[:n+1])[:16] + md5(str(total))[:16])

print(flag)
```

兔子你好

原题

U2FsdGVkX197ihEWFWSF8qzdJ/Y1GS6pieLsbQHFUA==

解题思路

```
兔子的加密
```

WriteUp

```
看上去有点像Base64，不过解出来是乱码，所以用Rabbit解密
```

NSCTF crypto200

原题

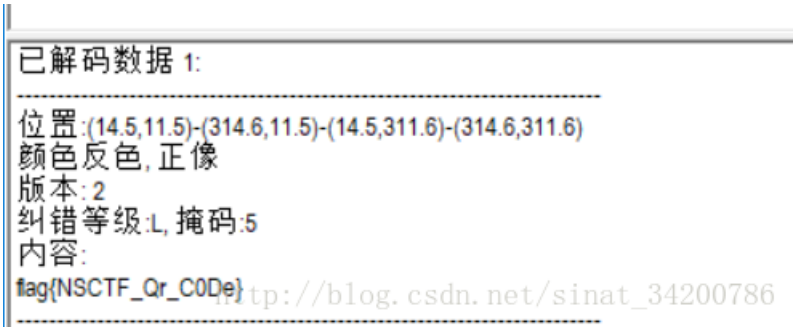
解题思路

```
Stegsovle直接搞定
```

WriteUp

```
Stegsolve载入后左右翻看，发现两张二维码
```

只有第二张可以扫描，轻松得到flag



RSAROLL

原题

```
{920139713,19}

704796792
752211152
274704164
18414022
368270835
483295235
263072905
459788476
483295235
459788476
663551792
475206804
459788476
428313374
475206804
459788476
425392137
704796792
458265677
341524652
```

解题思路

爆破密钥

WriteUp

简单了解Rsa算法后可知
n = 920139713，e = 19，由于n较小，所以直接爆破(分解）得到素数
p = 18443 q = 49891

fn = (p-1)(q-1)
这时 e 对于 fn 的模反元素有关系：ed - 1 = k*fn
等价于求解方程： ex + fn*y = 1
代入可得 : 19e + 920071380y = 1
此式可用扩展欧几里得算法求解,x 即为所求密钥

```
#include<iostream>

int get(int a,int b,int &x,int &y);
int main()
{
    int a,b,x,y;
    a = 18442;
    b = 49890;
    get(19,a*b,x,y);
    printf("%d %d",x,y);
    return 0;
}

//扩展欧几里得算法
//返回a,b的最大公约数并求x,y使满足 ax + by = gcd(a,b)
int get(int a,int b,int &x,int &y)
{
    if(b == 0)
    {
        x = 1;
        y = 0;
        return a;
    }
    else
    {
        int r = get(b,a%b,y,x);
        y -= x*(a/b);
        return r;
    }
}
```

加密过程：m 原文，c 密文

$$m^e \equiv c \pmod{n}$$

解密过程：d 密钥

$c^d \equiv m \pmod{n}$

此时上脚本解密即可

```
d = 96849619
n = 920139713
s = ''
with open('Rsa.txt','r') as R:
    for line in R:
        s += chr(pow(int(line),d,n))

print(s)
```

涨姿势点

对于Rsa算法的较深入了解：
1. 完整的加解密过程
2. 大素数分解困难保证算法安全性
扩展欧几里得算法还可以解特定关系的二元一次方程
pow(x,y,z)快速求解 x的y次方模z（秒解）而x**y%z则需要长时间运算

备注

参考资料：
[Rsa]　　'http://www.ruanyifeng.com/blog/2013/06/rsa_algorithm_part_one.html'
[扩展欧几里得] 'http://www.cnblogs.com/frog112111/archive/2012/08/19/2646012.html'

凯撒和某某加密

原题

解题思路

整个ASCii表的凯撒以及脑洞栅栏

WriteUp

直接凯撒解密发现都是无用信息

解出来的信息也不像是某种加密，猜测可能是在整个ASCii表中进行凯撒加密，试试

```
s = "aZZg/x\ZbavpZiEZp+n)o+"
a = '''!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~'''

for n in range(0,26):
    b = a[n:] + a[:n]
    t = str.maketrans(a,b)
    print(s.translate(t))

print('End')
```

结果中有这样一个出现flag字样的字符串

```
d]]j2{_]edys]lH]s.q,r.
e^^k3|^^fezt^mI^t/r-s/
f__l4}a_gf{u_nJ_u0s.t0
g``m5~b`hg|v`oK`v1t/u1
```

这个栅栏需要脑洞了，解法如下：

```
f__
l4}
a_
gf
{u
_n
J_
u0
s.
t0

key = 'flag{_Just_4_fun_0.0_}'
```

涨姿势点