NPUCTF 2020 Crypto

原创

Lan_Magnolia 于 2020-05-23 20:34:19 发布

文章标签： 算法 安全

本文链接：https://blog.csdn.net/cwr1499640048/article/details/105753567

NPUCTF 2020 Crypto

认清形势，建立信心

```
p = getPrime(25)
e = # Hidden
q = getPrime(25)
n = p * q
m = bytes_to_long(flag.strip(b"npuctf{").strip(b"}"))

c = pow(m, e, n)
print(c)
print(pow(2, e, n))
print(pow(4, e, n))
print(pow(8, e, n))
'''
169169912654178
128509160179202
518818742414340
358553002064450
'''
```

首先，这里展开一个公式：

((a mod x) ^ b) mod x = (a ^ b) mod x

作为小白，接下来是推导证明:

设 a = kx + d

(a mod x) ^ b = d ^ b

a ^ b = (kx + d) ^ b,此处二项式展开得知共b+1项 前b项都有x这个因数 最后一个为d ^ b.

则根据题目，

2 ^ e mod n = a

4 ^ e mod n = 2 ^ 2e mod n = (2 ^ e mod n) ^ 2 mod n = a ^ 2 mod n = b

a ^ 2 - b = kn(k=1,k=2…) 同理，a ^ 3 - c = kn(k=1,k=2…). n为两式的公因数，由此根据gcd（ a ^ 2 - b，a ^ 3 - c）求出n

```
from Crypto.Util.number import *

a = 128509160179202
b = 518818742414340
c = 358553002064450
n = GCD(a**2-b, a**3-c)
print n
#1054494004042394
```

$1054494004042394_{<16>} = 2 \cdot 18195301 \cdot 28977097$

p = 18195301 q = 28977097

接下来求取e，这里可以进行穷举，使用C++很快可以跑出e

```
#include<iostream>
#include<math.h>
#define LONG long long
using namespace std;

int main(){
    LONG e=1,c=2,n=527247002021197,a=128509160179202;
    while(e<100000000000){
        e++;
        c=c*2%n;
        if(c==a){
            cout<<e<<endl;
            return 0;
        }
    }
    return 0;
}
//808723997
```

接下来就是正常的RSA求解

```
from Crypto.Util.number import *
from gmpy2 import *

e = 808723997
p = 18195301
q = 28977097
c = 169169912654178
m = pow(c,invert(e,(p-1)*(q-1)),p*q)
flag = long_to_bytes(m)
print flag
```