primegame

发表于 2021-05-25 更新于 2021-05-26 分类于 Challenge， 2021， 第四届红帽杯网络安全大赛， Crypto
Challenge | 2021 | 第四届红帽杯网络安全大赛 | Crypto | primegame

点击此处获得更好的阅读体验

WriteUp来源

来自Bellin的博客

解题思路

这次的crypto题目简直离谱，虽然有难度，但是都是前面比赛出现过的原题的一丢丢魔改就拿来出题。所以基本上你会Google基本就能AK。

这个的原题writeup链接：Pokajeon / Kapojeon 2020

server 源代码

```
#!/usr/bin/env python3

from decimal import *
import math
import random
import struct
# from flag import flag

flag=b"123456"*6
assert (len(flag) == 48)
msg1 = flag[:24]
msg2 = flag[24:]
primes = [2]
for i in range(3, 90):
    f = True
    for j in primes:
        if i * i < j:
            break
        if i % j == 0:
            f = False
            break
    if f:
        primes.append(i)

print(primes)

getcontext().prec = 100
keys = []
for i in range(len(msg1)):
    keys.append(Decimal(primes[i]).ln())

print(keys)

sum_ = Decimal(0.0)
for i, c in enumerate(msg1):
    # print(i,c)
    sum_ += c * Decimal(keys[i])

ct = math.floor(sum_ * 2 ** 256)
print(ct)

sum_ = Decimal(0.0)
for i, c in enumerate(msg2):
    sum_ += c * Decimal(keys[i])

ct = math.floor(sum_ * 2 ** 256)
print(ct)
```

类似0/1背包问题，将[3,89]的素数的自然对数值作为密钥，然后和flag的每个byte乘后累加，最后再乘以2256。

恢复方法是低密度攻击。然后不管这些，其实魔改writeup的代码即可

```
# sagemath
import math
from decimal import *
import random
import struct

getcontext().prec = int(100)

primes = [2]
for i in range(3, 90):
    f = True
    for j in primes:
        if i * i < j:
            break
        if i % j == 0:
            f = False
            break
    if f:
        primes.append(i)

keys = []
for i in range(len(primes)):
    keys.append(Decimal(int(primes[i])).ln())

arr = []
for v in keys:
    arr.append(int(v * int(16) ** int(64)))

ct = 42598547504778133678996330091044685278303271259857188534566055054637206341058991 8
# ct = 59795204366044624902018477323298397401778025588194237904445467698064641708751545 3
def encrypt(res):
    h = Decimal(int(0))
    for i in range(len(keys)):
        h += res[i] * keys[i]

    ct = int(h * int(16)**int(64))
    return ct

def f(N):
    ln = len(arr)
    A = Matrix(ZZ, ln + 1, ln + 1)
    for i in range(ln):
        A[i, i] = 1
        A[i, ln] = arr[i] // N
        A[ln, i] = 64

    A[ln, ln] = ct // N

    res = A.LLL()

    for i in range(ln + 1):
        flag = True
        for j in range(ln):
            if -64 <= res[i][j] < 64:
                continue
            flag = False
            break
        if flag:
            vec = [int(v + 64) for v in res[i][:-1]]
            ret = encrypt(vec)
            if ret == ct:
                print(N, bytes(vec))
            else:
                print("NO", ret, bytes(vec))

for i in range(2, 10000):
    print(i)
    f(i)
```

Flag

```
1 flag{715c39c3-1b46-4c23-8006-27b43eba2446}
```


 # Challenge # 2021 # Crypto # 第四届红帽杯网络安全大赛
virtual_world
colorful code