

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

```
1 #!/usr/bin/env python3
2
3 from decimal import *
4 import math
5 import random
6 import struct
7 # from flag import flag
8
9 flag=b"123456"*6
10 assert (len(flag) == 48)
11 msg1 = flag[:24]
12 msg2 = flag[24:]
13 primes = [2]
14 for i in range(3, 90):
15     f = True
16     for j in primes:
17         if i * i < j:
18             break
19         if i % j == 0:
20             f = False
21             break
22     if f:
23         primes.append(i)
24
25 print(primes)
26
27 getcontext().prec = 100
28 keys = []
29 for i in range(len(msg1)):
30     keys.append(Decimal(primes[i]).ln())
31
32 print(keys)
33
34 sum_ = Decimal(0.0)
35 for i, c in enumerate(msg1):
36     # print(i,c)
37     sum_ += c * Decimal(keys[i])
38
39 ct = math.floor(sum_ * 2 ** 256)
40 print(ct)
41
42 sum_ = Decimal(0.0)
43 for i, c in enumerate(msg2):
44     sum_ += c * Decimal(keys[i])
45
46 ct = math.floor(sum_ * 2 ** 256)
47 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

```
1 # sagemath
2 import math
3 from decimal import *
4 import random
5 import struct
6
7 getcontext().prec = int(100)
8
9 primes = [2]
10 for i in range(3, 90):
11     f = True
12     for j in primes:
13         if i * i < j:
14             break
15         if i % j == 0:
16             f = False
17             break
18     if f:
19         primes.append(i)
20
21 keys = []
22 for i in range(len(primes)):
23     keys.append(Decimal(int(primes[i])).ln())
24
25 arr = []
26 for v in keys:
27     arr.append(int(v * int(16) ** int(64)))
28
29 ct = 425985475047781336789963300910446852783032712598571885345660550546372063410589918
30 # ct = 597952043660446249020184773232983974017780255881942379044454676980646417087515453
31 def encrypt(res):
32     h = Decimal(int(0))
33     for i in range(len(keys)):
34         h += res[i] * keys[i]
35
36     ct = int(h * int(16) ** int(64))
37     return ct
38
39 def f(N):
40     ln = len(arr)
41     A = Matrix(ZZ, ln + 1, ln + 1)
42     for i in range(ln):
43         A[i, i] = 1
44         A[i, ln] = arr[i] // N
45         A[ln, i] = 64
46
47     A[ln, ln] = ct // N
48
49     res = A.LLL()
50
51     for i in range(ln + 1):
52         flag = True
53         for j in range(ln):
54             if -64 <= res[i][j] < 64:
55                 continue
56             flag = False
57             break
58         if flag:
59             vec = [int(v + 64) for v in res[i][:ln]]
60             ret = encrypt(vec)
61             if ret == ct:
62                 print(N, bytes(vec))
63             else:
64                 print("NO", ret, bytes(vec))
65
66 for i in range(2, 10000):
67     print(i)
68     f(i)
```

Flag

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

- 本文作者: CTFHub
- 本文链接: <https://writeup.ctfhub.com/Challenge/2021/第四届红帽杯网络安全大赛/Crypto/wAXpivG6T6i6pi17rkUPjo.html>
- 版权声明: 本博客所有文章除特别声明外, 均采用 [BY-NC-SA](#) 许可协议。转载请注明出处!

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