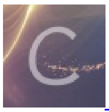
# 2021第五届强网杯网络安全挑战赛决赛-crypto writeup

ljahum 于 2021-07-15 12:02:44 发布 354 收藏 2

分类专栏： ctf

本文链接：https://blog.csdn.net/a_touhouer/article/details/118756480

版权


ctf 专栏收录该内容

4 篇文章 0 订阅

订阅专栏

## 2021第五届强网杯网络安全挑战赛决赛-crypto writeup

> 不垫底就算成功□□□

蛮恶心的，差点因为服务器太慢出不来

```
from sage.stats.distributions.discrete_gaussian_integer import DiscreteGaussianDistributionIntegerSampler
from random import randint, getrandbits
from secret import flag
import sys
import signal

q = 2 ^ 54
t = 83
T = 3
d = 1024
delta = int(q / t)
sigma = 2
P.<x> = PolynomialRing(ZZ)
f = x ^ d + 1
R.<X> = P.quotient(f)
D = DiscreteGaussianDistributionIntegerSampler(sigma=sigma)


def sample1():
    return R([D() for _ in range(d)])


def sample2():
    return R([randint(0, q - 1) for _ in range(d)])

def sample3(x):
    return [randint(0, T - 1) for _ in range(x)]


def Roundq(a):
    A = a.list()
    for i in range(len(A)):
        A[i] = A[i] % q
        if A[i] > (q / 2):
            A[i] = A[i] - q
    return R(A)


def Roundt(a):
    A = a.list()
    for i in range(len(A)):
        A[i] = A[i] % t
        if A[i] > (t / 2):
            A[i] = A[i] - t
    return R(A)


def keygen():
    s = sample1()
    a = Roundq(sample2())
    e = Roundq(sample1())
    pk = [Roundq(-(a * s + e)), a]
    return s, pk


def encrypt(m):
    u = sample1()
    e1 = sample1()
    e2 = sample1()
    return (Roundq(pk[0] * u + e1 + delta * m), Roundq(pk[1] * u + e2))


def baseT(n, b=T):
    v = []
    while True:
        x = n // b
        y = n % b
        v.append(y)
        if x == 0:
            break
        n = x
    v.reverse()
    return v

def mutual(k, c, s):
    tmp = t * Roundq(c[0] + c[1] * s)
    TMP = tmp.list()
    for i in range(len(TMP)):
        TMP[i] = round(TMP[i] / q)
    tmp2 = Roundt(R(TMP))
    if tmp2[min(k, d)] == 0:
        print(True)
    else:
        print(False)

signal.alarm(1024)
sk, pk = keygen()
print(f"public key:{pk[0].list()}, {pk[1].list()}")

namelist = ["admin", "Adam", "Bruce", "Chris", "David"]
users = dict()
for i in namelist:
    users[i] = getrandbits(32)
    users[i] = getrandbits(32)

menu = '''
1.Add friends
2.find friends
3.Send Message
4.Regist'''

friends = set()
while 1:
    print(f"Current number of users: {len(users)}")
    print(menu)
    op = int(input(">").strip())
    if op == 1:
        name = input("name:").strip()
        id_num = int(input("id:").strip())
        if name in users.keys():
            if id_num == users[name]:
                friends.add(name)
            else:
                print("failed")
        else:
            print("failed")
    elif op == 2:
        op2 = input("recv ct?(Y/N)").strip()
        if op2.upper() == "Y":
            for name in users.keys():
                id_num = users[name]
                x = baseT(id_num)
                y = x + sample3(d - len(x))
                ct = encrypt(R(y))
                print(ct[0].list(), ct[1].list())
                op3 = input("continue?(Y/N)")
                if op3.upper() == "N":
                    break
                elif op3.upper() != "Y":
                    sys.exit(1)
        elif op2.upper() != "N":
            sys.exit(1)

        for i in range(len(users)):
            c1 = input("c1:").strip().split(" ")
            c2 = input("c2:").strip().split(" ")
            cc1 = list(map(int, c1))
            cc2 = list(map(int, c2))
            mutual(i, [R(cc1), R(cc2)], sk)

    elif op == 3:
        name = input("name:").strip()
        message = input("message:").strip()
        if name not in friends:
            print("failed")
        else:
            if name == "admin":
                if message == "give me the flag":
                    print(flag)
            else:
                print(f"send '{message}' to {name}")
    elif op == 4:
        name = input("name:").strip()
        if name not in users.keys():
            users[name] = getrandbits(32)
            print("succeeded")
        else:
            print("failed")
    else:
        sys.exit(1)
```

## 分析

需要我们泄露admin的id来添加一个friend来get massage

这个题是一个经典的 CCA attack on FPSI

针对全同态的一个攻击，但和常规情况不同的是在生成密钥时并未像标准加密系统中为了方便硬件运算使用0 1序列生成的多项式

而是使用 [-7,-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6,7] 得序列随机生成的私钥

```
from sage.stats.distributions.discrete_gaussian_integer import DiscreteGaussianDistributionIntegerSampler
D = DiscreteGaussianDistributionIntegerSampler(sigma=sigma)

def sample_2():
    return R([randint(0,1) for _ in range(d)])
def sample1():
    return R([D() for _ in range(d)])

def keygen():
    s = sample1()
    a = Roundq(sample2())
    e = Roundq(sample1())
    pk = [Roundq(-(a * s + e)), a]
    return s, pk
```

原版payload只需要给$t = m$

可以恢复出密钥

payload:

```
M=delta//4+50
Recoverd_key=[]
for i in range(d):
    Recoverd_key.append(recover_key(i))

def recover_key(i):
    t1=[0 for _ in range(d)]
    t1[i]=M
    t2=M
    cc0=pk[0]+R(t1)
    cc1=pk[1]+R(t2)
    ans = decrypt([cc0,cc1]).list()
    return ans[i]
```

但题目情况密钥并不为0 1 序列而且Oracle attack只能判断为该位上数字是否为0

在改变t1[i]=M M的个数后发现以下性质：

| 私钥\ 返回的数据 | M | 2M | 7M | 5M |
|---|---|---|---|---|
| 7 | 2 | 2 | 3 | 0 |
| 6 | 2 | 2 | 3 | 0 |
| 5 | 2 | 2 | 3 | 0 |
| 4 | 1 | 2 | 3 | 0 |
| 3 | 1 | 1 | 3 | -1 |
| 2 | 1 | 1 | 2 | -1 |
| 1 | 1 | 1 | 2 | -1 |
| 0 | 0 | 1 | 2 | -1 |
| -1 | 0 | 0 | 2 | -2 |
| -2 | 0 | 0 | 1 | -2 |
| -3 | -1 | 0 | 1 | -2 |
| -4 | -1 | -1 | 1 | -2 |
| -5 | -1 | -1 | 1 | -3 |
| -6 | -1 | -1 | 0 | -3 |
| -7 | -1 | -1 | 0 | -3 |

只要按顺序 n = [8,7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5,-4] t1 = n*M 就可以对密钥进行一个padding Oracle

## exp

```
from pwn import *
# from icecream import *
from tqdm import tqdm

from time import *
p1 = time()

# -----------------------------------
# get pk
# io = remote('0.0.0.0',10001)
io = remote('172.20.5.23',8001)

q = 2 ^54
t = 83
T = 3
d = 1024
delta = int(q / t)
sigma = 2
P.<x> = PolynomialRing(ZZ)
f = x ^ d + 1
R.<X> = P.quotient(f)

def Roundt(a):
 A = a.list()
 for i in range(len(A)):
  A[i] = A[i] % t
  if A[i] > (t / 2):
   A[i] = A[i] - t
 return R(A)


def Roundq(a):
 A = a.list()
 for i in range(len(A)):
  A[i] = A[i] % q
  if A[i] > (q / 2):
   A[i] = A[i] - q
 return R(A)
def mutual2(k, c, s):
 tmp = t * Roundq(c[0] + c[1] * s)
 TMP = tmp.list()
 for i in range(len(TMP)):
  TMP[i] = round(TMP[i] / q)
 tmp2 = Roundt(R(TMP))
 return tmp2



io.recvuntil('public key:[')
pk1 = io.recvuntil(']')[:-1]
io.recvuntil('[')
pk2 = io.recvuntil(']')[:-1]
# print(pk1)
# print(pk2[:100])
pk2 = [int(i) for i in pk2.split(b',')]
# print(pk2[:10])
pk1 = [int(i) for i in pk1.split(b',')]
pk=[R(pk1),R(pk2)]
for i in range(1024-5):
 print(io.recvuntil('>'))
 io.sendline('4')
 io.sendline(str(i))




M=delta//4+50
padding = [8,7, 6, 5, 4, 3, 2, 1,  0, -1, -2, -3, -4, -5,-4]
sks=      [-7,-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6,7]
rk=[100 for i in range(1024)]

for x in range(len(sks)):

 print(io.recvuntil('>'))
 io.sendline('2')
 print(io.recvuntil('recv ct?(Y/N)'))
 io.sendline('Y')
 sleep(1)
 print(io.recvuntil('['))
 adminC1 = io.recvuntil(']')[:-1]
 print((io.recvuntil('[')))
 adminC2 = io.recvuntil(']')[:-1]
 adminC1 = [int(i) for i in adminC1.split(b',')]
 # print(pk2[:10])
 adminC2 = [int(i) for i in adminC2.split(b',')]

 print(io.recvuntil('continue?(Y/N)'))
 print(io.recvuntil('continue?(Y/N)'))
 io.sendline('N')
 pad = padding[x]
 print(f'第{x}个了')
 for k in tqdm(range(1024)):
  t1=[0 for _ in range(d)]
  t1[k]=pad*M
  t2=M
  # ================================
  cc0=(pk[0]+R(t1)).list()
  payload_c1 = ''
  for i in cc0:
   payload_c1 += str(i)
   payload_c1+=' '
  payload_c2 = ''
  cc1=(pk[1]+R(t2)).list()
  for i in cc1:
   payload_c2 += str(i)
   payload_c2+=' '

  io.recvuntil('c1:')
  io.sendline(payload_c1)
  # sleep(1)
  io.recvuntil('c2:')
  io.sendline(payload_c2)
  # =====================================

  fb = io.recvline()
  if(b'True' in fb and rk[k]==100):
   rk[k] = sks[x]
 # input()
for i in range(len(rk)):
    if(rk[i]==100):
        rk[i]=7

sk = R(rk)
adminCT=[R(adminC1),R(adminC2)]
# cc0=R(ct[0].list())
# cc1=R(ct[1].list())

ans = mutual2(0,adminCT,sk)

x = ans.list()[:25]
admin_id =0
use=[]
ids=[]
for i in x:
 admin_id *= T
 admin_id += i
 use.append(i)
 ids.append(admin_id)

print(rk)
print(ids)
# io.interactive()
sleep(1)
for id in ids:
 print(io.recvuntil('>'))
 io.sendline('1')
 print(io.recvuntil('name:'))
 io.sendline('admin')
 print(io.recvuntil('id:'))
 io.sendline(str(id))
 print(io.recvline())
print(io.recvuntil('>'))
io.sendline('3')
io.sendlineafter('name:','admin')
io.sendlineafter('message:','give me the flag')
sleep(1)
print(io.recv(2048))
p2 = time()
print(p2-p1)
```

## FLAG

```
b'name:'
b'id:'
b'failed\n'
b'Current number of users: 1024\n\n1.Add friends\n2.find friends\n3.Send Message\n4.Regist\n>'
b'name:'
b'id:'
b'failed\n'
b'Current number of users: 1024\n\n1.Add friends\n2.find friends\n3.Send Message\n4.Regist\n>'
b'flag{CCA_attack_BFV_123698745}\n'
[*] Closed connection to 172.20.5.23 port 8001
/mnt/c/U/1/De/qwb决赛/bfv
```

> 服务器OI速度及其慢 等了半天才打通