

Computer Security Hw0x0A Writeup

- Realname: 胡安鳳
- ID on course web: alfons0329
- Student ID: R08922024



tags: Computer Security NTU CS CS CTF Writeup

Mandalorian (crypto)



A classical RSA LSB attack.



Step 1: Get c, e, n in RSA from the server.

The c, e, n are the cipher and public key pair (e, N) in RSA encryption.

(N, e) 是公鑰, (N, d) 是私鑰

加密消息 [编辑]

假设Bob想给Alice送一个消息
小于 N 的非负整数 n , 比如他
数字。假如他的信息非常长的
以将 n 加密为 c :

$$c \equiv n^e \pmod{N}$$

Step 2: LSB attack as the Cryptography 110 ppt shown.

LSB Oracle Attack - 解法二

Oracle

$$c \xrightarrow{\text{oracle}} m$$

推論

$$y_1 \quad x_0$$

$$\begin{aligned} m &= x_0 + 2y_1 \\ r &\equiv \lfloor x_0 + 2y_1 \rfloor_n \pmod{2} \\ &\equiv x_0 \pmod{2} \\ \Rightarrow x_0 &\equiv r \pmod{2} \end{aligned}$$

Navigation icons and page number 45/100

LSB Oracle Attack - 解法二

Oracle

$$(2^{-1})^e c \xrightarrow{\text{oracle}} 2^{-1} m$$

推論

$$y_2 \quad x_1 \quad x_0$$

$$\begin{aligned} 2^{-1} m &= 2^{-1} x_0 + x_1 + 2y_2 \\ r &\equiv \lfloor 2^{-1} x_0 + x_1 + 2y_2 \rfloor_n \pmod{2} \\ &\equiv \lfloor 2^{-1} x_0 \rfloor_n + x_1 \pmod{2} \\ \Rightarrow x_1 &\equiv r - \lfloor 2^{-1} x_0 \rfloor_n \pmod{2} \end{aligned}$$

Navigation icons and page number 46/100

LSB Oracle Attack - 解法二

Oracle

$$(2^{-2})^e c \xrightarrow{\text{oracle}} 2^{-2} m$$

推論

$$y_3 \quad x_2 \quad x_1 \quad x_0$$

$$\begin{aligned} 2^{-2} m &= 2^{-2} x_0 + 2^{-1} x_1 + x_2 + 2y_3 \\ r &\equiv \lfloor 2^{-2} x_0 + 2^{-1} x_1 + x_2 + 2y_3 \rfloor_n \pmod{2} \\ &\equiv \lfloor 2^{-2} x_0 + 2^{-1} x_1 \rfloor_n + x_2 \pmod{2} \\ \Rightarrow x_2 &\equiv r - \lfloor 2^{-2} x_0 + 2^{-1} x_1 \rfloor_n \pmod{2} \end{aligned}$$

Navigation icons and page number 47/100

LSB Oracle Attack - 解法二

碎碎念

- x_i 代表 m 的第 i 個 bit
- y_i 代表 m 整段從最高位的 bit 到最低位數來第 i 個 bit
- 每次可以推論一個 bit，需要 $O(\log(n))$ 次 oracle

Navigation icons and page number 48/100

The flag will be $(xn, xn - 1, \dots, x1, x0)$, with $c \rightarrow m$, and MOD being 16.

So the overall exploitation will be.

$$((base^{-ie}))c \rightarrow (base^{-i})m$$

The i above is equivalent to `pow_cnt` shown below and the LSB message forging will be

```
1 multiplied_c = pow(base, pow_cnt * e, n) * c % n # ((base ^ {-ie}))c
2 rem.sendline(str(multiplied_c))
3 multiplied_m = int(rem.recvline().split()[-1]) # (base ^ {-i})m
```

So the overall LSB attack part

```

1 pow_cnt = 0
2 try_cnt = 0
3 flag_res = 0
4 flag_list = []
5
6 while try_cnt < 1024 // 4:
7     rem.sendlineafter('>', '2')
8
9     # under the mod n cycle
10    multiplied_c = pow(base, pow_cnt * e, n) * c % n
11    rem.sendline(str(multiplied_c))
12    multiplied_m = int(rem.recvline().split()[-1])
13    r = multiplied_m % n % MOD
14
15    tmp = 0
16    start_pow = len(flag_list)
17    for i in range(len(flag_list)):
18        tmp += pow(base, start_pow, n) * flag_list[i]
19        start_pow -= 1
20
21    xi = (r - ((tmp) % n)) % MOD
22
23    # check if the received bit(ot byte) is the padding zero
24    # (continuously padding zero), break as the threshold meets
25    if xi == 0:
26        if continuous_padding >= 10:
27            break
28        else:
29            continuous_padding += 1
30    else:
31        continuous_padding = 0
32
33    flag_list.append(xi)
34    # make flag_res grow bigger, ex:
35    # x2x1x0 will be mod ** 2 * x2 + mod ** 1 * x1 + mod ** 0 8 x0
36    # (just like naive method for turning 0x7FB to 2043)
37    flag_res = (((MOD ** pow_cnt) % n * xi) % n + flag_res) % n
38    print(hex(flag_res))
39    pow_cnt += 1
40    try_cnt += 1

```

Gradually decrypting to have a flag.

```

0x6f4b454e7d
0x506f4b454e7d
0x3506f4b454e7d
0x73506f4b454e7d
0x573506f4b454e7d
0x6573506f4b454e7d
0x66573506f4b454e7d
0x766573506f4b454e7d
0x1766573506f4b454e7d
0x61766573506f4b454e7d
0x861766573506f4b454e7d
0x4861766573506f4b454e7d
0x94861766573506f4b454e7d
0x1061766573506f4b454e7d

```

```
0x494861f766573506f4b454e7d
0x7494861766573506f4b454e7d
0x47494861766573506f4b454e7d
0x447494861766573506f4b454e7d
0x3447494861766573506f4b454e7d
0xc3447494861766573506f4b454e7d
0x6c3447494861766573506f4b454e7d
0x66c3447494861766573506f4b454e7d
0x466c3447494861766573506f4b454e7d
0x3466c3447494861766573506f4b454e7d
0x33466c3447494861766573506f4b454e7d
0x833466c3447494861766573506f4b454e7d
0x4833466c3447494861766573506f4b454e7d
0x44833466c3447494861766573506f4b454e7d
0x544833466c3447494861766573506f4b454e7d
0x4544833466c3447494861766573506f4b454e7d
0x74544833466c3447494861766573506f4b454e7d
0x74544833466c3447494861766573506f4b454e7d
0x3074544833466c3447494861766573506f4b454e7d
0x73074544833466c3447494861766573506f4b454e7d
0x673074544833466c3447494861766573506f4b454e7d
0x5673074544833466c3447494861766573506f4b454e7d
0xf75673074544833466c3447494861766573506f4b454e7d
0xf75673074544833466c3447494861766573506f4b454e7d
0x6f75673074544833466c3447494861766573506f4b454e7d
0x96f75673074544833466c3447494861766573506f4b454e7d
0x596f75673074544833466c3447494861766573506f4b454e7d
0xb596f75673074544833466c3447494861766573506f4b454e7d
0x7b596f75673074544833466c3447494861766573506f4b454e7d
0x77b596f75673074544833466c3447494861766573506f4b454e7d
0x477b596f75673074544833466c3447494861766573506f4b454e7d
0x1477b596f75673074544833466c3447494861766573506f4b454e7d
0x41477b596f75673074544833466c3447494861766573506f4b454e7d
0xc41477b596f75673074544833466c3447494861766573506f4b454e7d
0x4c41477b596f75673074544833466c3447494861766573506f4b454e7d
0x64c41477b596f75673074544833466c3447494861766573506f4b454e7d
0x464c41477b596f75673074544833466c3447494861766573506f4b454e7d
0x464c41477b596f75673074544833466c3447494861766573506f4b454e7d
0x464c41477b596f75673074544833466c3447494861766573506f4b454e7d
0x464c41477b596f75673074544833466c3447494861766573506f4b454e7d
0x464c41477b596f75673074544833466c3447494861766573506f4b454e7d
0x464c41477b596f75673074544833466c3447494861766573506f4b454e7d
0x464c41477b596f75673074544833466c3447494861766573506f4b454e7d
0x464c41477b596f75673074544833466c3447494861766573506f4b454e7d
0x464c41477b596f75673074544833466c3447494861766573506f4b454e7d
FLAG[Youg@tH3Ft4GIHavesPoKEN] *] Closed connection to edu-ctf.csie.org port 10192
```