CTF4y 講義資料

yoshiking(@y05h1k1ng)

December 13, 2019

問題

[TokyoWesterns CTF 4th 2018] Crypto - mixed cipher (233points/39solves)

いろいろな攻撃手法が一度に学べる！！おいしい！！

やってみそ

（スコアサーバーの）8000 番が開いてるはず

```
yoshiking@yoshikingdom:~/Documents/sakumon/mixed_cipher/doc$ nc localhost 8000
Welcome to mixed cipher :)
I heard bulldozer is on this channel, be careful!
1: encrypt
2: decrypt
3: get encrypted flag
4: get encrypted key
```

概要

４つのメニューがある

- encrypt
- decrypt
- get encrypted flag
- get encrypted key

encrypt

入力した値を RSA と AES で暗号化してくれる

```
42 def encrypt():
43     p = raw_input('input plain text: ').strip()
44     print('RSA: {}'.format(pubkey.encrypt(p, 0)[0].encode('hex')))
45     print('AES: {}'.format(aes_encrypt(p).encode('hex')))
```

decrypt

RSA で復号してくれる

ただし、bulldozer が最後の 1byte 以外を#に置き換える

⇒ 復号結果の最後の 1byte しかわからない！！

```
47 def decrypt():
48     c = raw_input('input hexencoded cipher text: ').strip().decode('hex')
49     print('RSA: {}'.format(bulldozer(privkey.decrypt(c)).encode('hex')))
```

get encrypted flag

aes で暗号化した flag をくれる

ただ、iv がわからない（#に置き換えられている）

```
51 def print_flag():
52     print('here is encrypted flag :)')
53     p = flag
54     print('another bulldozer is coming!')
55     print(('#'*BLOCK_SIZE+aes_encrypt(p)[BLOCK_SIZE:]).encode('hex'))
```

get encrypted key

RSA で暗号化した aes 鍵をくれる

```
57 def print_key():
58     print('here is encrypted key :)')
59     p = aeskey
60     c = pubkey.encrypt(p, 0)[0]
61     print(c.encode('hex'))
```

ざっと読んだ感じ...

- decrypt
  - 自明に LSB Decryption Oracle Attack ですね
- get encrypted flag
  - iv 消されてるのどうしようか...

とりあえず LSB Decryption Oracle Attack やろうぜ！！

LSB Decryption Oracle Attack

任意の暗号文を復号した結果の最下位ビットを得ることができるとき、与えられた暗号文に対応する平文を求める攻撃

$$2^e * c \equiv (2 * m)^e \bmod n$$

復号すると、

$$(2 * m)^{e*d} \equiv 2 * m \bmod n$$

となり、結果 $2 * m$ について

- 最下位ビットが 1 であれば、$2 * m > n$
- 最下位ビットが 0 であれば、$2 * m < n$

もうちょい詳しく...

n は奇数

LSB Decryption Oracle Attack でやること

- print key で入手できる rsa で暗号化された aes 鍵を復元する

さあ、攻撃！！！の前に...

n 持ってないじゃん！！！

n の取得

- これはそんなに難しくない
- (LSB Decryption Oracle Attack に比べて) よくやる手法な気がする

n の取得

encrypt で好きな平文を暗号化できる。

- -1 を入れるパターンもあるけど、今回は入力できない

適当な平文 m_i について、rsa の式をいじってみる。

$$m_i^e \equiv c_i \bmod n$$

より、商と余りの関係から、

$$\begin{aligned} m_i^e &= c_i + k_i * n \\ m_i^e - c_i &= k_i * n \end{aligned}$$

が求められる。

n の取得

異なる入力 m_1, m_2 を与えたとき、

$$\begin{cases} m_1^e - c_1 &= k_1 * n \\ m_2^e - c_2 &= k_2 * n \end{cases}$$

となる。ここで、この 2 つの最大公約数を取ると n が求めることができる。

$$\begin{aligned} gcd(m_1^e - c_1, m_2^e - c_2) &= gcd(k_1 * n, k_2 * n) \\ &= n \end{aligned}$$

とりあえず、nの取得、LSB Decryption Oracle Attackまで書いてaes鍵を復元してみよう

現状

- aes 鍵は手に切れた
- ただ、iv がない。

iv の入手

aes encrypt を見てみると、iv の生成は genrandbits を使っている
*Rightarrow*python は Mersenne Twister を使って random を生成する...

Mersenne Twister

Mersenne Twister の内部状態の復元