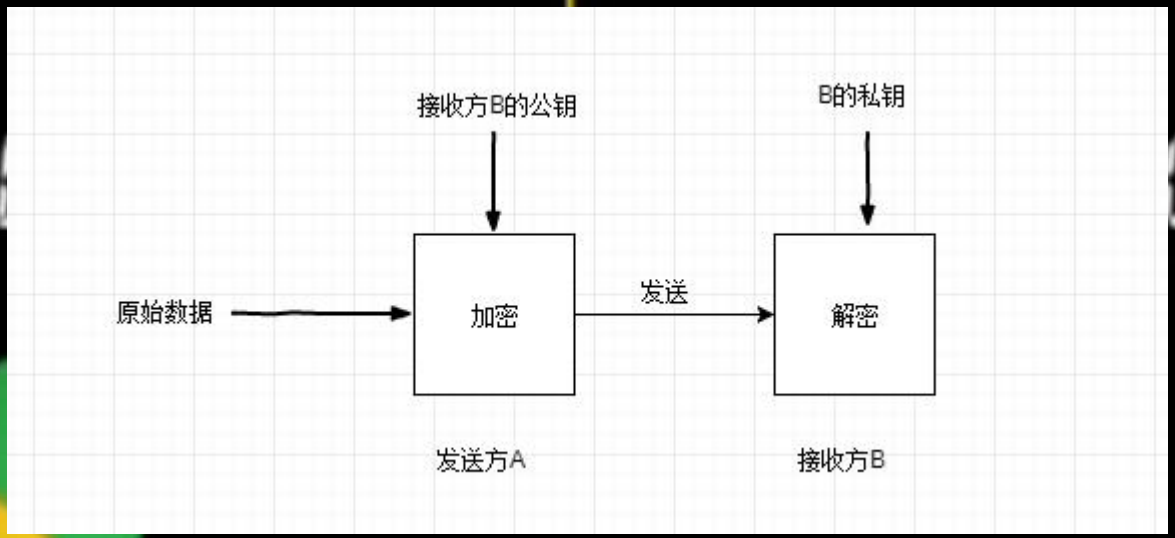
RSA_初级

——Islab_Winter　by Ⅰ Fpop

内容

- 引述
- 数字签名
- 攻击方法
- 工具
- 常见题型

引述



这将解除公开色的作用

This undoes the effect of her public color

数字签名

- 在真实场景中，我们签名/画押/盖章是为了保证什么

- 为什么要盖章
- 为什么要大写
- 为什么要在拾和万前加/
- 为什么要写￥

签名$sig(m) = m^d (mod\ n)$，将其发个收信方，
他用公钥进行运算可以得到：
$(sig(m))^e = m^{ed} (mod\ n) \equiv m (mod\ n)$

- 但是这会出现一个问题？？如果数据太大

但是如果源数据太大，用私钥进行运算花时间很长，所以现在一般会用hash函数进行处理

签名$sig(m) = hash(m)^d (mod\ n)$，将其发个收信方，
他用公钥进行运算可以得到：
$(sig(m))^e = hash(m)^{ed} (mod\ n) \equiv hash(m) (mod\ n)$

用下唐老板讲的故事

假如你去玩，花了500订酒店，花了300定车票，于是你发送了两条消息300，500，并且发送了签名$300^d \pmod n$和$500^d \pmod n$，并且用你的公钥e可以验证签名，银行名正言顺的扣了你800。但是如果银行比较鸡贼，它把你的两个签名乘了起来
$300^d \pmod n \times 500^d \pmod n = 150000^d \pmod n$，并且伪造了你消费15万的记录，用你的公钥通过了验证，名正言顺的扣了你15万

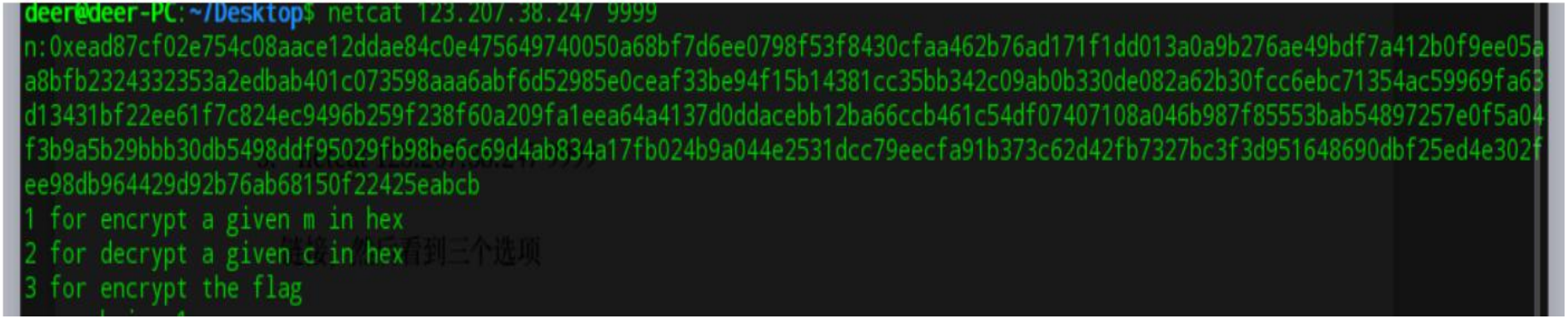
攻击方法

选择密文攻击

百度百科是这样写的：

密码分析者事先任意搜集一定数量的密文，让这些密文透过被攻击的解密算法解密，透过未知的密钥获得解密后的明文。由此能够计算出加密者的私钥或者分解模数，运用这些信息，攻击者可以恢复所有的明文

- 给你解密和加密选择
- 给你被加密的密文



分析过程：

用加密算法加密一个较为简单的数，这里我选择的是 x = 2,密文记为r,由RSA原理可知:

$$x^e (mod\ n) = r$$
$$r^d (mod\ n) = x$$

同理，它所给出的密文c和我们需要求的明文m也有下面式子:

$$m^e (mod\ n) = c$$
$$c^d (mod\ n) = m$$

将两个密文相乘,令其等于y:

$$y = r * c$$

分析过程：

令u为其明文,由欧拉定理可知:

$$u = y^d (mod\ n)$$

将y带入可得:

$$(r * c)^d (mod\ n) = u$$
$$r^d (mod\ n) * c^d (mod\ n) = u$$
$$x * m = u$$
$$m = u//x$$

就可以求出明文

基于分解模数的RSA攻击

- yafu(yafu 基本命令：factor(n))
- factordb
  - http://www.factordb.com/index.php
- n比较小的话，可以选择暴力破解
- sage有专门因式分解的函数
  - http://sagecell.sagemath.org

公共因子攻击

- 一般有两个n值，求出其公共因子
- 如果n1 = p*q1,n2 = p*q2，可以求出gcd(n1,n2)的公因子
- 明文都没什么联系，e也一般取65537

栗子：


```
'''
例子: factor跑不出来, 通过求公因子, 然后分别求出p,q
'''
n1=90510139654040844828700878648214555351590086960429
53021965631089095795348830954383127323853272528967729
31104517960540769359266568331166058120488657114632772
02884558749272811281211173235796912047923999131066275
43274457036172455814805715668293705603675386878220947
72218691411299045272217436371363029768515966932895152
08919384034527976506858495236581919474114290688297340
53745180460758604283051344339641429819373112365211739
21616042049416707199643850685052616838938685049979610
20036254042456457962716903107488043 27
n2=13225948396179603816062046418717214792668512413625
09156999752436424399599196101889415005920782409383742
04513752405503100502093989645063185189916201425759266
23780411532257230701985821629425722030608722035570690
47417125923815394709531030352283197166466606754264903
44616217256562348690055012934239751847019297291700772
80251436216167293058560030089006140224 3Q7542567957118
17872069827124772614325795379812780557553445737670769
51793312062480275004564657590263719816033564139497109
94207370175501187315320536623858566574 3
```

低加密指数攻击

- 特点：m和e都比较小，e一般取3

$$m^e < n, m = \sqrt[3]{c}$$

有可能稍微大一点，$m^e = k * n + c, m = \sqrt[e]{c + kn}$，
枚举k进行求解

- 工具：

```
#gmpy2的应用
#!usr/env/python3
import gmpy2
for k in range(10):
    gmpy2.iroot(c+k*n,e)
```

Rabin算法

- 辨别特点，e = 2
- 加密 $c = m^2 (mod\ n)$
- 这里只关注解密方法,具体看这个
  - https://en.wikipedia.org/wiki/Rabin_cryptosystem

```python
def rabin_decrypt(c, p, q, e=2):
    n = p * q
    mp = pow(c, (p + 1) / 4, p)  #整除 用//
    mq = pow(c, (q + 1) / 4, q)
    yp = gmpy2.invert(p, q)
    yq = gmpy2.invert(q, p)
    r = (yp * p * mq + yq * q * mp) % n
    rr = n - r
    s = (yp * p * mq - yq * q * mp) % n
    ss = n - s
    return (r, rr, s, ss)
```

还有好多其他攻击方法，可以自行百度看看

工具

gmpy和gmpy2

```
#gmpy2  在python3中安装:
sudo apt-get install libmpfrc++-dev
sudo apt-get install libmpc-dev
sudo pip3 install gmpy2
#gmpy2  在python2.7中安装:
sudo pip install python-gmpy2
#如果还是不行，请用aptitude
```

- 安装不成功的看这个
  - https://www.cnblogs.com/pcat/p/5746821.html
- 使用教程：
  - https://gmpy2.readthedocs.io/en/latest/

分解大整数n

- yafu
- factordb
  - http://www.factordb.com/index.php

libnum

- 安装教程：
  - https://www.cnblogs.com/pcat/p/7225782.html

```
git clone https://github.com/hellman/libnum
cd libnum
python setup.py install
```

我们在RSA中常用函数有：

```
import libnum
libnum.n2s(n)   #Number to string
libnum.b2s(b)   #Binary to string
libnum.gcd(num1,num2) #求最大公约数
libnum.lcm(num1,num2) #最小公倍数
libnum.s2n(s)   #string to Number
libnum.s2b(s)   #string to Binar
```

CTF_RSA_tool

- 一个集成工具
- 下载：https://github.com/lFpop/CTF-RSA-tool

RSAtools

- 已知n,e,p,q，可以求出d

pycrypto

- 使用：
  - https://pypi.org/project/pycrypto/

安装：

```
deepin:
sudo pip3 install pycrypto
```

- 读取公钥内容，然后打印

```
from Crypto.PublicKey import RSA
key = RSA.importKey(open('./public.pem', 'r').read())
print(key.n, key.e)
```

- 读取flag信息：

```
from Crypto.Util import number
with open('./flag.enc', 'rb') as f:
    data = f.read()
    print(number.bytes_to_long(data))
```

openssl

- 使用方法：
  - https://www.jianshu.com/p/15b1d935a44b
- linux里面应该是自带的

```
#提取公钥信息
openssl rsa -pubin -text -modulus -in [公钥.pem]
#用私钥进行解密
openssl rsautl -decrypt -in [密文.enc]-inkey [私钥.pem]
#其他命令可以看看openssl文档，点击上面那个openssl
```

rsatool

- https://github.com/ius/rsatool
- 另一款强大的rsa解密工具，主要用来生成私钥，具体用法看writeup

rsa

- 这是一个纯python实现的库，不依赖底层文件，优点是部署容易，缺点是速度比较慢
- 其他快速使用方法：
  - https://www.sha256.cc/2015/12/09/4-python%E5%AE%9E%E7%8E%B0rsa%E7%9A%84%E5%87%A0%E7%A7%8D%E6%96%B9%E6%A1%88/

CTF中常见RSA题型

- 已知p、q、e求解d
- 已知c、n、e求解明文
- 已知p、q、e、c求解明文
- 已知c、e，求解明文

已知p、q、e求解d

由于

$$e * d \equiv 1 \ (mod \ \phi(n))$$

所以要求d，我们可以先求

$$\phi(n) = (p-1) * (q-1)$$

方法:

```
'''
第一种gmpy2解法
mpz:Multiple-precision Integers（多精度型整数）
gmpy2.invert():求模逆
'''


'''
直接用扩展欧几里得法
'''


'''
用工具解密——RSAtools
ps:用这个工具记得将e转化为hex形式
'''
```

已知c、n、e求解明文

已知c、n、e求解明文/已知c,q,p,e

根据上述分解大数工具，我们可以求出p,q

利用$\phi(n) = (p-1) * (q-1)$求出$\phi(n)$

之后求出私钥信息d，原理同上，

利用$m \equiv c^d (mod\, n)$ 求出m

一个可能会用到的东西

当你知道了 n ,e,d,q,p这些东西和flag.enc文件，这个时候怎么利用工具解密？

1．rsa库

2．openssl方法

3．其他

作业：

- RSA实践
  - http://www.shiyanbar.com/ctf/1828
  - hint:所讲求 d 三种方法
- Fake_rsa
- RSA
  - http://www.shiyanbar.com/ctf/1772