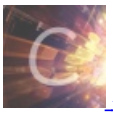
crypto实验报告

原创

临风而眠 于 2021-09-02 21:42:55 发布 51 收藏

分类专栏： # 密码学 文章标签： python crypto 古典密码 RSA加密 密码学



密码学实验报告

> 笔者大一小学期参加的一个网络攻防创新实践课的实验报告作业，纯入门级别

文章目录

- 密码学实验报告
  - 实验目标
    - 第一部分
    - 第二部分
  - 详细步骤
    - 第一部分
      - 1.栅栏密码基础型
      - 2.凯撒密码
      - 3.培根密码
    - 第二部分
      - 1.搜教程
      - 2.具体实现
      - 3.其他参考教程链接
  - 总结反思

实验目标

第一部分

下列密文均使用古典密码加密，请识别并写出下列密文的加密类型。

tsrlnhiafcisiee

wklvlvfdvhu

BAABBAABBBABAAABAABAABAAABAABAAAAABAAAAAAAABAABBBAABBABABAAAAAAAAABBAB

第二部分

下面是一次RSA加密中的数据，请尝试识别可用的攻击方式，写出解密思路，并写出对应的python脚本。

```
c = 112360596109401550405154208154590881085484386951111626769797403708036126824495959753155216154568359072461704976719389305216702402563680 63

e = 3

n = 123814470394550598363280518848914546938137731026777975885846733672494493975703069760053867471836249473290828799962586855892685902902050630018312939010564945676699712246249820341712155938398068732866646422826619477180434858148938235662092482058999079105450136181685141895955574548671667320167741641072330259009
```

详细步骤

第一部分

> 解密工具网站:http://ctf.ssleye.com/
> 网站2:http://www.metools.info/code/

1.栅栏密码基础型

先做的第三个培根密码，解密出来有this is ,然后盲猜这里面也有this is ， 去掉this is之后，还剩下：rlniafcee

然后发现thisis被分搁在这串字符的各个地方，那就猜是栅栏密码，然后因为t和h分别是第一个和第六个，相差了5，那我就猜是5栏，所以五个分为一组：

```
tsrln
hiafc
isiee
```

那么解密结果就是:

```
thisisrailfence
```

2.凯撒密码

先做的第三个培根密码，解密出来有this is

然后盲猜这里面也有this is ， 然后看到lvlv一样，那肯定就是th isis,那说不定这就是位移的密码,i到l是往后移动两位，s到v也是

那就解决了,移动两位得到答案：

```
thisiscaser
```

3.培根密码

根据其全为A、B的特征，可判断其为培根密码，对照规则表可知解密结果为:

```
thisisbaconian
```

第二部分

> RSA加密介绍:https://www.jianshu.com/p/9f7905f2c2a3

1.搜教程

先去不假思索地搜了很多教程，然后看到一个类似的题目，十分高兴，想直接粘代码□...

Known High Bits Message Attack

知道 N、c、e 以及明文 m 的高位，可以恢复出全部明文 m，由于算法的约束，只适用于 e 较小的情况。

Python

```
n = 223375098517794820019222119192313564325064519541078717034235090602460782501295801994519136733873431
e = 3
m = 705519548471280823325741526303717702300738529975629361855203926032785538185646434198894660591912685
c = pow(m, e, n)

beta = 1
epsilon = beta**2 / 7
nbits = n.nbits()
kbits = floor(nbits * (beta**2 / e - epsilon))
mbar = m & (2**nbits - 2**kbits)
# mbar = 0x37e34e4ec193004e9f4cbf4a2c1f9416f6ac425d365e44e431d8fbbea082825612bc3daeac837db750f2cc025744

PR.<x> = PolynomialRing(Zmod(n))
f = (mbar + x)**e - c
roots = f.small_roots(X=2**kbits, beta=1) # find root < 2^kbits with factor = n
print mbar + roots[0]
```

可是这个地方一直报错：

```
PR.<x> = PolynomialRing(Zmod(n))
```

可我网上也搜不到报错原因，而且看到了更多的教程都是这样子写的，问了学长才知道那是sage环境下的代码...

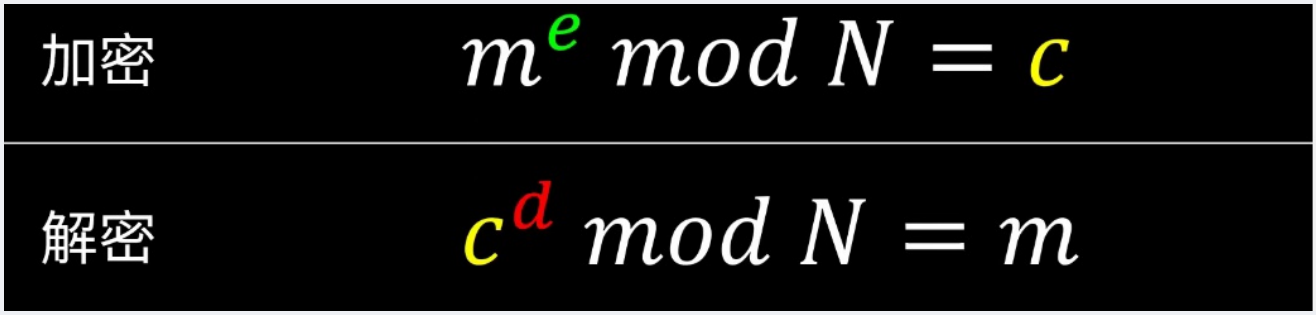
2.具体实现

简单了解RSA:

> □B站通俗讲解链接(感觉这个讲的挺好□□)

$m^e\quad mod\quad n\quad=\quad c $

原始数据m,求e次幂，e可看作加密时的密钥，将m 除以n并取余数得到密文c，正向算出c很容易，但是反向推出m很难



> d代表另一个用于解密的密钥

e很小,所以可以用低加密指数攻击

原理是已知$m^e\quad =\quad kn+ c $ ，在有限时间内枚举 k ，再对(kn + c) 开 e 次方，如果得到整数就很有可能是 m

```
import gmpy2

c = 112360596109401550405154208154590881085484386951111626769797403708036126824495959753155216154568359072461704976719389305216702402563680 63

e = 3

n = 123814470394550598363280518848914546938137731026777975885846733672494493975703069760053867471836249473290828799962586855892685902902050630018312939010564945676699712246249820341712155938398068732866646422826619477180434858148938235662092482058999079105450136181685141895955574545867166732016774164107233025900 09

for i in range(0,1000):
    # 返回一个元组，m[1] 的值表示开方结果是否为整数
    m = gmpy2.iroot(c+i*n,3)
    if(m[1]==1):
        print(m[0])
```

得到答案为:

```
2239776471831658234108454124349621200139611967
```

3.其他参考教程链接

廖雪峰RSA教程

python RSA解密：

https://www.cnblogs.com/lanston1/p/11875706.html

http://t.zoukankan.com/xautxuqiang-p-6067456.html

https://zhuanlan.zhihu.com/p/76228394

总结反思

- 密码学挺有意思的，上面古典密码的解密全都可以试试用python实现