

# 密码学-仿射加密 一谈

原创

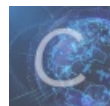
飞鸿踏雪 (蓝屏选手) 于 2019-08-10 10:58:20 发布 117 收藏

分类专栏: [密码学](#) 文章标签: [CTF](#)、[Crypto](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_41252520/article/details/98963957](https://blog.csdn.net/qq_41252520/article/details/98963957)

版权



[密码学](#) 专栏收录该内容

0 篇文章 0 订阅

订阅专栏

## 原理

- 将字母编码成一个表, 规定 **A=0**, 一直到 **Z=25**, 共 **26** 个字母

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

## 加密

- 给出一个仿射函数, 形如  $Y = (aX + b) \bmod 26$ , 且 **a**, **26** 互质。其中 **X** 表示明文输入, **Y** 表示密文输出。

## 解密

- 将加密函数进行求逆, 即对 **a** 求  $G(n-1)$ ,  $n=26$  的乘法逆元, 令  $E(a)$  表示 **a** 的乘法逆元, 则解密函数为  $X = (E(a)(Y-b)) \bmod 26$ , 其中 **Y** 为密文输入, **X** 为明文输出。
- 还有种情况就是只给了 **b**, 这样的话, 就要找出 1~25 中所有与 **26** 互质的数, 并求其乘法逆元。

## 脚本

```
#encoding=utf-8

def xgcd(a,b,n):
    if b==0:
        return 1,0,a
    x,y,g=xgcd(b,a%b,n)
    t=x
    x=y
    y=t-(a//b)*x
    return x%n,y%n,g

def gcd(a,b):
    if b==0:
        return a
    return gcd(b,a%b)

def get_a():
    a buf=[]
```

```

    for i in range(1,26):
        if gcd(i,26)==1:
            x,y,g=xgcd(i,26,26)
            a_buf.append(x)
    return a_buf

def FS_Encrypt(plaintext,a,b,m):
    ret=''
    for c in plaintext.upper():
        ret+=chr((((ord(c)-65)*a+b)%26)+65)
    if m==0:
        return ret
    else:
        return ret.lower()

def FS_Decrypt(cipher,a,b,m):
    ret=[]
    tmp=''
    if a==0:
        a_buf=get_a()
        for ta in a_buf:
            tmp=''
            for c in cipher.upper():
                tmp+=chr((ta*((ord(c)-65)-b)%26)+65)
            if m==0:
                ret.append(tmp)
            else:
                ret.append(tmp.lower())
        return ret
    else:
        x,y,g=xgcd(a,26,26)
        for c in cipher.upper():
            tmp+=chr((x*((ord(c)-65)-b)%26)+65)
        if m==0:
            ret.append(tmp)
        else:
            ret.append(tmp.lower())
        return ret

print '[*] 1.加密, 2.解密:'
f=int(raw_input())
if f==1:
    print '[*] 请输入明文:'
    plaintext=raw_input()
    print '[*] 请输入a:'
    a=int(raw_input())
    print '[*] 请输入b:'
    b=int(raw_input())
    print '[*] 0.输出为大写, 1.输出为小写:'
    m=int(raw_input())
    print FS_Encrypt(plaintext,a,b,m)
else:
    if f==2:
        print '[*] 请输入密文:'
        cipher=raw_input()
        print '[*] 请输入a:'
        a=int(raw_input())
        print '[*] 请输入b:'
        b=int(raw_input())

```

```
print '[*] 0.输出为大写, 1.输出为小写:'  
m=int(raw_input())  
print FS_Decrypt(cipher,a,b,m)
```

## 测试

```
[*] 1.加密, 2.解密:  
1  
[*] 请输入明文:  
hack  
[*] 请输入a:  
5  
[*] 请输入b:  
18  
[*] 0.输出为大写, 1.输出为小写:  
1  
bscq
```

```
[*] 1.加密, 2.解密:  
2  
[*] 请输入密文:  
bscq  
[*] 请输入a:  
0  
[*] 请输入b:  
18  
[*] 0.输出为大写, 1.输出为小写:  
1  
['jaky', 'dami', 'hack', 'fauw', 'baeu', 'paio', 'lasm', 'zawg', 'vage', 'tayq',  
'xaos', 'raqc']  
https://blog.csdn.net/qq\_41252520
```