Single table Substitution Cipher

EN | ZH

通用特点

在单表替换加密中，所有的加密方式几乎都有一个共性，那就是明密文一一对应。所以说，一般有以下两种方式来进行破解

- 在密钥空间较小的情况下，采用暴力破解方式
- 在密文长度足够长的时候，使用词频分析，http://quipqiup.com/

当密钥空间足够大，而密文长度足够短的情况下，破解较为困难。

凯撒密码

原理

凯撒密码（Caesar）加密时会将明文中的 **每个字母** 都按照其在字母表中的顺序向后（或向前）移动固定数目（**循环移动**）作为密文。例如，当偏移量是左移 3 的时候（解密时的密钥就是 3）：

```
明文字母表：ABCDEFGHIJKLMNOPQRSTUVWXYZ
密文字母表：DEFGHIJKLMNOPQRSTUVWXYZABC
```

使用时，加密者查找明文字母表中需要加密的消息中的每一个字母所在位置，并且写下密文字母表中对应的字母。需要解密的人则根据事先已知的密钥反过来操作，得到原来的明文。例如：

```
明文：THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
密文：WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ
```

根据偏移量的不同，还存在**若干特定的恺撒密码名称**：

- 偏移量为 10：Avocat （A→K）
- 偏移量为 13：ROT13
- 偏移量为 -5：Cassis （K 6）
- 偏移量为 -6：Cassette （K 7）

此外，还有还有一种基于密钥的凯撒密码 Keyed Caesar。其基本原理是 **利用一个密钥，将密钥的每一位转换为数字（一般转化为字母表对应顺序的数字），分别以这一数字为密钥加密明文的**

每一位字母。

这里以 **XMan 一期夏令营分享赛宫保鸡丁队 Crypto 100** 为例进行介绍。

```
密文：s0a6u3u1s0bv1a
密钥：guangtou
偏移：6,20,0,13,6,19,14,20
明文：y0u6u3h1y0uj1u
```

破解

对于不带密钥的凯撒密码来说，其基本的破解方法有两种方式

1. 遍历 26 个偏移量，适用于普遍情况
2. 利用词频分析，适用于密文较长的情况。

其中，第一种方式肯定可以得到明文，而第二种方式则不一定可以得到正确的明文。

而对于基于密钥的凯撒密码来说，一般来说必须知道对应的密钥。

工具

一般我们有如下的工具，其中 JPK 比较通用。

- JPK，可解带密钥与不带密钥
- http://planetcalc.com/1434/
- http://www.qqxiuzi.cn/bianma/ROT5-13-18-47.php

移位密码

与凯撒密码类似，区别在于移位密码不仅会处理字母，还会处理数字和特殊字符，常用 ASCII 码表进行移位。其破解方法也是遍历所有的可能性来得到可能的结果。

Atbash Cipher

原理

埃特巴什码（Atbash Cipher）其实可以视为下面要介绍的简单替换密码的特例，它使用字母表中的最后一个字母代表第一个字母，倒数第二个字母代表第二个字母。在罗马字母表中，它是这样出现的：

```
明文：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
密文：Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
```

下面给出一个例子：

```
明文：the quick brown fox jumps over the lazy dog
密文：gsv jfrxp yildm ulc qfnkh levi gsv ozab wlt
```

破解

可以看出其密钥空间足够短，同时当密文足够长时，仍然可以采用词频分析的方法解决。

工具

- http://www.practicalcryptography.com/ciphers/classical-era/atbash-cipher/

简单替换密码

原理

简单替换密码（Simple Substitution Cipher）加密时，将每个明文字母替换为与之唯一对应且不同的字母。它与恺撒密码之间的区别是其密码字母表的字母不是简单的移位，而是完全是混乱的，这也使得其破解难度要高于凯撒密码。 比如：

```
明文字母 : abcdefghijklmnopqrstuvwxyz
密钥字母 : phqgiumeaylnofdxjkrcvstzwb
```

a 对应 p，d 对应 h，以此类推。

```
明文：the quick brown fox jumps over the lazy dog
密文：cei jvaql hkdtf udz yvoxr dsik cei npbw gdm
```

而解密时，我们一般是知道了每一个字母的对应规则，才可以正常解密。

破解

由于这种加密方式导致其所有的密钥个数是$26!$，所以几乎上不可能使用暴力的解决方式。所以我们 一般采用词频分析。

工具

- http://quipqiup.com/

仿射密码

原理

仿射密码的加密函数是 $E(x) = (ax + b) \pmod{m}$，其中

- x 表示明文按照某种编码得到的数字
- a 和 m 互质
- m 是编码系统中字母的数目。

解密函数是 $D(x) = a^{-1}(x - b) \pmod{m}$，其中 a^{-1} 是 a 在 $\mathbb{Z}_m$ 群的乘法逆元。

下面我们以 $E(x) = (5x + 8) \bmod 26$ 函数为例子进行介绍，加密字符串为 `AFFINE CIPHER`，这里我们直接采用字母表 26 个字母作为编码系统

明文	A	F	F	I	N	E	C	I	P	H	
x	0	5	5	8	13	4	2	8	15	7	
$y = 5x + 8$	8	33	33	48	73	28	18	48	83	43	
$y \mod 26$	8	7	7	22	21	2	18	22	5	17	
密文	I	H	H	W	V	C	S	W	F	R	

其对应的加密结果是 `IHHWVCSWFRCP`。

对于解密过程，正常解密者具有 a 与 b，可以计算得到 a^{-1} 为 21，所以其解密函数是 $D(x) = 21(x - 8) \pmod{26}$，解密如下

密文	I	H	H	W	V	C	S	W	F
y	8	7	7	22	21	2	18	22	5
$x = 21(y - 8)$	0	-21	-21	294	273	-126	210	294	-63
$x \mod 26$	0	5	5	8	13	4	2	8	15
明文	A	F	F	I	N	E	C	I	P

可以看出其特点在于只有 26 个英文字母。

破解

首先，我们可以看到的是，仿射密码对于任意两个不同的字母，其最后得到的密文必然不一样，所以其也具有最通用的特点。当密文长度足够长时，我们可以使用频率分析的方法来解决。

其次，我们可以考虑如何攻击该密码。可以看出当$a = 1$时，仿射加密是凯撒加密。而一般来说，我们利用仿射密码时，其字符集都用的是字母表，一般只有 26 个字母，而不大于 26 的与 26 互素的个数一共有

$$\phi(26) = \phi(2) \times \phi(13) = 12$$

算上 b 的偏移可能，一共有可能的密钥空间大小也就是

$$12 \times 26 = 312$$

一般来说，对于该种密码，我们至少得是在已知部分明文的情况下才可以攻击。下面进行简单的分析。

这种密码由两种参数来控制，如果我们知道其中任意一个参数，那我们便可以很容易地快速枚举另外一个参数得到答案。

但是，假设我们已经知道采用的字母集，这里假设为 26 个字母，我们还有另外一种解密方式，我们只需要知道两个加密后的字母 y_1, y_2 即可进行解密。那么我们还可以知道

$$y_1 = (ax_1 + b) \pmod{26}$$
$$y_2 = (ax_2 + b) \pmod{26}$$

两式相减，可得

$$y_1 - y_2 = a(x_1 - x_2) \pmod{26}$$

这里 y_1, y_2 已知，如果我们知道密文对应的两个不一样的字符 x_1 与 x_2，那么我们就可以很容易得到 a，进而就可以得到 b 了。

例子

这里我们以 TWCTF 2016 的 super_express 为例进行介绍。简单看一下给的源码

```
import sys
key = '****CENSORED***************'
flag = 'TWCTF{*******CENSORED********}'

if len(key) % 2 == 1:
    print("Key Length Error")
    sys.exit(1)

n = len(key) / 2
encrypted = ''
for c in flag:
    c = ord(c)
    for a, b in zip(key[0:n], key[n:2*n]):
        c = (ord(a) * c + ord(b)) % 251
    encrypted += '%02x' % c
```

```
print encrypted
```

可以发现，虽然对于 flag 中的每个字母都加密了 n 次，如果我们仔细分析的话，我们可以发现

$$
\begin{aligned}
c_1 &= a_1 c + b_1 \\
c_2 &= a_2 c_1 + b_2 \\
&= a_1 a_2 c + a_2 b_1 + b_2 \\
&= kc + d
\end{aligned}
$$

根据第二行的推导，我们可以得到其实 c_n 也是这样的形式，可以看成 $c_n = xc + y$，并且，我们可以知道的是，key 是始终不变化的，所以说，其实这个就是仿射密码。

此外，题目中还给出了密文以及部分部分密文对应的明文，那么我们就很容易利用已知明文攻击的方法来攻击了，利用代码如下

```
import gmpy

key = '****CENSORED****************'
flag = 'TWCTF{*******CENSORED********}'

f = open('encrypted', 'r')
data = f.read().strip('\n')
encrypted = [int(data[i:i + 2], 16) for i in range(0, len(data), 2)]
plaindelta = ord(flag[1]) - ord(flag[0])
cipherdalte = encrypted[1] - encrypted[0]
a = gmpy.invert(plaindelta, 251) * cipherdalte % 251
b = (encrypted[0] - a * ord(flag[0])) % 251
a_inv = gmpy.invert(a, 251)
result = ""
for c in encrypted:
    result += chr((c - b) * a_inv % 251)
print result
```

结果如下

```
➜  TWCTF2016-super_express git:(master) X python exploit.py
TWCTF{Faster_Than_Shinkansen!}
```

评论