凯撒密码

The Caesar Cipher

刘卓

凯撒密码，或称凯撒加密、凯撒变换、变换加密，是一种最简单且最广为人知的加密技术。距今已有 2000 余年的历史。

凯撒密码属于密码学中的替换加密，即密文是由明文中的所有字母在字母表上向后（或向前）按照一个固定数目进行偏移而生成。

1 加密

首先需要设置偏移量：

例 1

当偏移量为 16 时，得到如下加密方法。所有的字母 **A** 向拉丁字母表右移动 16 位，将被替换成字母 **Q**，字母 **B** 向字母表右移动 16 位变成字母 **R**，以此列推。

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

如明文：Person。加密后可以得到：FUHIED。 □

由此我们可以知道，密钥空间 $|K|$ = 密钥数量 = 26。

或者 $|K| = 25$，当偏移量为 26 时，字母 A 替换成字母 A 这种情况不算。

例 2

使用 *Python* 程序加密 I am a person.

```
Plaintext = 'iamaperson'
ciphertext = ''

key = 4 %偏移量

for i in Plaintext:
    if (ord(i) + key) > 122:
            alpha = chr((ord(i) + key) - 26)
```

```
    else:
        alpha = chr(ord(i) + key)
    ciphertext += alpha
print(ciphertext)
```

输出：meqetivwsr □

作为明文时一般没有空格，大小写也不区分。

2 解密

由于**拉丁字母表**中的字母有且仅有 26 个，使得凯撒密码易受频率分析和暴力破解的攻击。最多 26 种可能性即可破解。

例 3

破解密文：exxegoexsrgi

使用 *Python* 程序破解：

```
cipher = 'exxegoexsrgi'

for j in range(26):
    Plaintext = ''
    for i in cipher:
        if (ord(i) - j) < 97:
            alpha = chr((ord(i) - j) + 26)
        else:
            alpha = chr(ord(i) - j)
        Plaintext += alpha
    print(Plaintext)
```

输出：

$Shift$	$Output$
A	$exxegoexsrgi$
B	$dwwdfndwrqfh$
C	$cvvcemcvqpeg$
D	$buubdlbupodf$
E	$attackatonce$
F	$zsszbjzsnmbd$
...	...
X	$haahjrhavujl$
Y	$gzzgiqgzutik$
Z	$fyyfhpfytshj$

由此可知，解密后明文为 attack at once □

3 凯撒密码的改进

是否觉得凯撒密码太过于简单被破解？我们有三种方法可以使得密文更难破解：

- 随机替换 (Randomize the Order of Substitution)
- 特殊符号替代 (Homophonic Substitution)
- 多字母替代 (Poly-alphabetic Substitution)

3.1 随机替换

即不按照字母表顺序进行替换，而是随机移位，使得每个字母一一对应。如：

a	b	c	d	e	f	g	h	i	j	...
c	t	h	a	q	z	x	v	n	p	...

计算密钥空间。我们知道‘A’的偏移选择一共有 26 种（包括自己），‘B’的偏移选择一共有 25 种，以此列推。密钥空间一共就有 26 种，介于 2^{88} 和 2^{89} 之间。相比起传统的凯撒密码，破解难度大大提升。

然而，该方法还是易受频率分析的影响而被破解。明文和密文仍然存在遵循字母的频率分布。即统计一串字符每个字母出现次数除以总字符数（仅字母）。

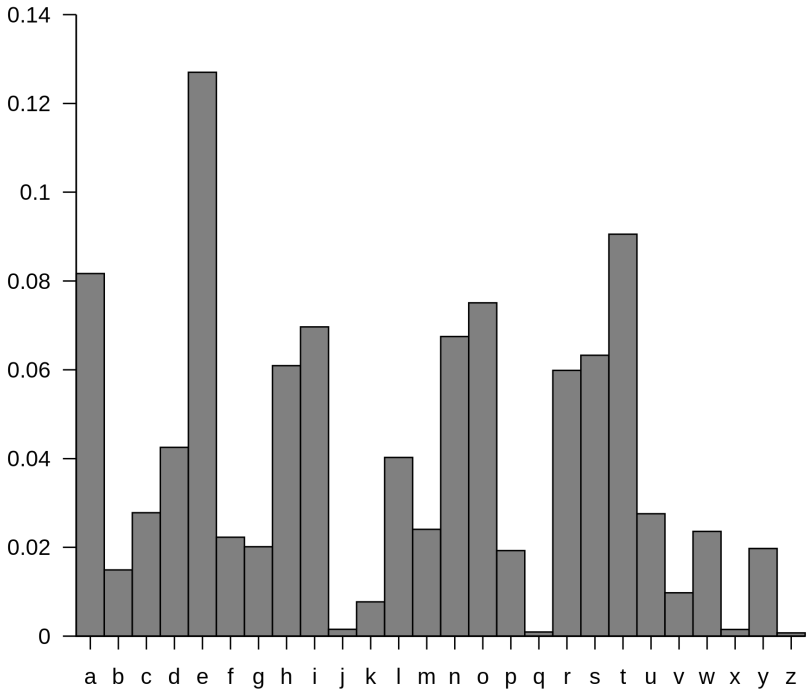

图 1: 英语语言材料中的字母频率

明文足够情况下，通过对密文的出现字母频率分析，依然可以推理出明文。

3.2 特殊符号替代

字母可以使用特殊符号替代，标点符号，数字，数学符号，希腊字母表，甚至是 emoji 表情都可以。并且每个字母可以分配多个密文符号。

a	b	c	d	e	f	g	h	i	j	k	l	m	n
!	4	#	$	1	%	&	*	(	)	3	2	=	+
†	∘	ξ	**N**	6	↗	♡	♯	♭			↘	◁	▷
↙				⇓		↻		∅					♠
Θ				⇑									

o	p	q	r	s	t	u	v	w	x	y	z
[	9	]	{	}	:	;	7	<	>	5	?
8	♣		Ω	⋆	÷	∪				⟅	
∞				♮	□	ℏ					
				∉	↖						

图 2: 特殊符号替代

例 4

密文 ◊□□Δ 既可以是 $FOOD$ 也可以是 $AMMO$。 □

例 5

明文 OK 既可以是"83" 也可以是"[3"。 □

相比随机替换，不容易被频率分析破解。

3.3 多字母替代

1467 年，莱昂 · 巴蒂斯塔 · 阿尔贝蒂（Leon Battista Alberti）发明了密码盘。允许发送者对明文的不同部分使用不同的字母。16 世纪，有人根据给定的密钥使用多个凯撒密码对明文进行加密。

例 6

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

其中，密钥 (Key) 为 REV

对明文 $Avenged\ Sevenfold$ 进行加密：

明文	A	V	E	N	G	E	D	S	E	V	E	N	F	O	L	D
$Key1$	R			E			U			M			W			U
$Key2$		Z			K			W			I			S		
$Key3$			Z			Z			Z			I			G	
密文	R	Z	Z	E	K	Z	U	W	Z	M	I	I	W	S	G	U

得到密文 $RZZEKZUWZMIIWSGU$ □

多字母替代的密钥空间等于 26 的密钥长度次方。如果 Key = WATER，则密钥空间为 26^5。

其安全性再被发明后的三个世纪内都没有被破解。直至 1863 年被查尔斯 · 巴贝奇（Charles Babbage）使用 Kasiski's 测试，利用推断 Key 的长度，破解多字母替代加密法。