

CTF 中那些脑洞大开的编码和加密

2016-07-14 23:14:53 27 4ido10n 3417 量子猫的安全站 出处: Tools



0x00 前言

正文开始之前先闲扯几句吧，玩 CTF 的小伙伴也许会遇到类似这样的问题:表哥，你知道这是什么加密吗？其实 CTF 中脑洞密码题(非现代加密方式)一般都是各种古典密码的变形，一般出题者会对密文进行一些处理，但是会给留一些线索，所以写此文的目的是想给小伙伴做题时给一些参考，当然常在 CTF 里出现的编码也可以了解一下。本来是想尽快写出参考的文章，无奈期间被各种事情耽搁导致文章断断续续写了 2 个月，文章肯定有许多没有提及到，欢迎小伙伴补充，总之，希望对小伙伴们有帮助吧！最后欢迎小伙伴来[博客](<https://www.hackfun.org/>)玩耍:P(ps:由于写文章是用 markdown，而论坛编辑器不支持 markdown 语法，虽然我已经尽力去调整对其字符，可是效果还是不尽人意，如果影响阅读理解可以去博客阅读:P)

0x01 目录

常见编码:

- 1.ASCII 编码
- 2.Base64/32/16 编码
- 3.shellcode 编码
- 4.Quoted-printable 编码
- 5.XXencode 编码
- 6.UUencode 编码
- 7.URL 编码
- 8.Unicode 编码
- 9.Escape/Unescape 编码
- 10.HTML 实体编码
- 11.敲击码(Tap code)
- 12.莫尔斯电码(Morse Code)
- 13.编码的故事

各种文本加密

换位加密:

- 1.栅栏密码(Rail-fence Cipher)

2. 曲路密码(*Curve Cipher*)
3. 列移位密码(*Columnar Transposition Cipher*)

替换加密:

1. 埃特巴什码(*Atbash Cipher*)
2. 凯撒密码(*Caesar Cipher*)
3. ROT5/13/18/47
4. 简单换位密码(*Simple Substitution Cipher*)
5. 希尔密码(*Hill Cipher*)
6. 猪圈密码(*Pigpen Cipher*)
7. 波利比奥斯方阵密码 (*Polybius Square Cipher*)
8. 夏多密码(曲折加密)
9. 普莱菲尔密码(*Playfair Cipher*)
10. 维吉尼亚密码(*Vigenère Cipher*)
11. 自动密钥密码(*Autokey Cipher*)
12. 博福特密码(*Beaufort Cipher*)
13. 滚动密钥密码(*Running Key Cipher*)
14. Porta 密码(*Porta Cipher*)
15. 同音替换密码(*Homophonic Substitution Cipher*)
16. 仿射密码(*Affine Cipher*)
17. 培根密码(*Baconian Cipher*)
18. ADFGX 和 ADFGVX 密码(*ADFG/VX Cipher*)
19. 双密码(*Bifid Cipher*)
20. 三分密码(*Trifid Cipher*)
21. 四方密码(*Four-Square Cipher*)
22. 棋盘密码 (*Checkerboard Cipher*)
23. 跨棋盘密码(*Straddle Checkerboard Cipher*)
24. 分组摩尔斯替换密码(*Fractionated Morse Cipher*)
25. Bazerics 密码(*Bazerics Cipher*)
26. Digrafid 密码(*Digrafid Cipher*)
27. 格朗普雷密码(*Grandpré Cipher*)
28. 比尔密码(*Beale ciphers*)
29. 键盘密码(*Keyboard Cipher*)

其他有趣的机械密码:

1. 恩尼格玛密码

代码混淆加密:

- 1.asp 混淆加密
- 2.php 混淆加密
- 3.css/js 混淆加密
- 4.VBScript.Encode 混淆加密
- 5.ppencode
- 6.rrencode
- 7.jjencode/aaencode
- 8.JSfuck
- 9.jother
- 10.brainfuck 编程语言

相关工具

参考网站

彩蛋

0x02 正文

常见编码

1.ASCII 编码

ASCII 编码大致可以分作三部分组成:

第一部分是: ASCII 非打印控制字符 (参详 ASCII 码表中 0-31);

第二部分是: ASCII 打印字符, 也就是 CTF 中常用到的转换;

十进制	二进制	符号	十进制	二进制	符号	十进制	二进制	符号	十进制	二进制
0	0000 0000	NUL	32	0010 0000	[空格]	64	0100 0000	@	96	0110 0000
1	0000 0001	SOH	33	0010 0001	!	65	0100 0001	A	97	0110 0001
2	0000 0010	STX	34	0010 0010	"	66	0100 0010	B	98	0110 0010
3	0000 0011	ETX	35	0010 0011	#	67	0100 0011	C	99	0110 0011
4	0000 0100	EOT	36	0010 0100	\$	68	0100 0100	D	100	0110 0100
5	0000 0101	ENQ	37	0010 0101	%	69	0100 0101	E	101	0110 0101
6	0000 0110	ACK	38	0010 0110	&	70	0100 0110	F	102	0110 0110
7	0000 0111	BEL	39	0010 0111	'	71	0100 0111	G	103	0110 0111
8	0000 1000	BS	40	0010 1000	(72	0100 1000	H	104	0110 1000
9	0000 1001	HT	41	0010 1001)	73	0100 1001	I	105	0110 1001
10	0000 1010	LF	42	0010 1010	*	74	0100 1010	J	106	0110 1010
11	0000 1011	VT	43	0010 1011	+	75	0100 1011	K	107	0110 1011
12	0000 1100	FF	44	0010 1100	,	76	0100 1100	L	108	0110 1100
13	0000 1101	CR	45	0010 1101	-	77	0100 1101	M	109	0110 1101
14	0000 1110	SO	46	0010 1110	.	78	0100 1110	N	110	0110 1110
15	0000 1111	SI	47	0010 1111	/	79	0100 1111	O	111	0110 1111
16	0001 0000	DLE	48	0011 0000	0	80	0101 0000	P	112	0111 0000
17	0001 0001	DC1	49	0011 0001	1	81	0101 0001	Q	113	0111 0001
18	0001 0010	DC2	50	0011 0010	2	82	0101 0010	R	114	0111 0010
19	0001 0011	DC3	51	0011 0011	3	83	0101 0011	S	115	0111 0011
20	0001 0100	DC4	52	0011 0100	4	84	0101 0100	T	116	0111 0100
21	0001 0101	NAK	53	0011 0101	5	85	0101 0101	U	117	0111 0101
22	0001 0110	SYN	54	0011 0110	6	86	0101 0110	V	118	0111 0110
23	0001 0111	ETB	55	0011 0111	7	87	0101 0111	W	119	0111 0111
24	0001 1000	CAN	56	0011 1000	8	88	0101 1000	X	120	0111 1000
25	0001 1001	EM	57	0011 1001	9	89	0101 1001	Y	121	0111 1001
26	0001 1010	SUB	58	0011 1010	:	90	0101 1010	Z	122	0111 1010
27	0001 1011	ESC	59	0011 1011	;	91	0101 1011	[123	0111 1011
28	0001 1100	FS	60	0011 1100	<	92	0101 1100	\	124	0111 1100
29	0001 1101	GS	61	0011 1101	=	93	0101 1101]	125	0111 1101
30	0001 1110	RS	62	0011 1110	>	94	0101 1110	^	126	0111 1110
31	0001 1111	US	63	0011 1111	?	95	0101 1111	_	127	0111 1111

第三部分是：扩展 ASCII 打印字符(第一第三部分详见[ASCII 码表](<http://www.asciima.com/>)解释)。

编码转换示例

源文本：`The quick brown fox jumps over the lazy dog`

要转的:

The quick brown fox jumps over the lazy dog

给我转!

URL格式

%54%68%65%20%71%75%69%63%68%20%62%72%6F%77%6E%20%66%6F%78%20%6A%75%6D%70%73%20%6F%76%65%72%20%74%68%65%20%6C%61%7A%79%20%64%6F%67

还原

SQL_En:

0x5400680065002000710075006900630068002000620072006F0077006E00200066006F00780020006A0075006D007000730020006F00760065007200200074006800650020006C0061007A007900200064006F006700

还原

Hex:

0x5468652071756963682062726F776E20666F78206A756D7073206F76657220746865206C617A7920646F67

还原

Asc:

84 104 101 32 113 117 105 99 107 32 98 114 111 119 110 32 102 111 120 32 106 117 109 112 1

单个还原

MD5_32:

9E107D9D372BB6826BD81D3542A419D6

MD5_16:

372BB6826BD81D35

Base64:

VGhlIHFlawNrlGJyb3dulGZveCBqdW1wcyBvdmVylHRaZSBsYXp5IGRvZw==

解密Base64:

ASCII 编码对应十进制:

预览源代码打印关于

```

1  84 104 101 32 113 117 105 99 107 32 98 114 111 119 110 32
   102 111 120 32 106 117 109 112 115 32 111 118 101 114 32
   116 104 101 32 108 97 122 121 32 100 111 103

```

对应可以转换成二进制，八进制，十六进制等。

2.Base64/32/16 编码

base64、base32、base16 可以分别编码转化 8 位字节为 6 位、5 位、4 位。16,32,64 分别表示用多少个字符来编码，这里我注重介绍 base64。Base64 常用于在通常处理文本数据的场合，表示、传输、存储一些二进制数据。包括 MIME 的 email，email via MIME,在 XML 中存储复杂数据。

编码原理: *Base64* 编码要求把 3 个 8 位字节转化为 4 个 6 位的字节, 之后在 6 位的前面补两个 0, 形成 8 位一个字节的形势, 6 位 2 进制能表示的最大数是 2 的 6 次方是 64, 这也是为什么是 64 个字符(A-Z,a-z, 0-9, +, /这 64 个编码字符, =号不属于编码字符, 而是填充字符)的原因, 这样就需要一张映射表, 如下:

The Base64 Alphabet

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		



举个例子(*base64*):

> 源文本: T h e
>
> 对应 *ascii* 码:84 104 101
>
> 8 位 *binary*: 01010100 01101000 01100101
>
> 6 位 *binary*: 010101 000110 100001 100101

>
> 高位补 0: 000010101 00000110 00100001 00100101
>
> 对应 *ascii* 码: 21 6 33 37
>
> 查表: V G h l

利用 *Python base64* 模块, 我们分别可以这样加密解密 *base64 32 16*:

```
C:\WINDOWS\system32\cmd.exe - python
C:\Users\L>python
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec 5 2015, 20:40:30) [MSC v.1500 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>>> import base64
>>> mingwen="The quick brown fox jumps over the lazy dog"
>>> miwen=base64.b64encode(mingwen)
>>> print miwen
VGh1IHFlYWNRIGJyb3duIGZveCBqdW1wcyBvdnVyIHRoZSBsYXp5IGRvZw==
>>> miwen1=base64.b32encode(mingwen)
>>> print miwen1
KRUGKIDROVUWG2ZAMJZG653OEBTG66BANJ2W24DTEBXXMZLSEB2GQZJANRQXU6JAMRXWO===
>>> miwen2=base64.b16encode(mingwen)
>>> print miwen2
54686520717569636B2062726F776E20666F78206A756D7073206F76657220746865206C617A7920646F67
>>> base64decode=base64.b64decode(miwen)
>>> print base64decode
The quick brown fox jumps over the lazy dog
>>> base32decode=base64.b32decode(miwen1)
>>> print base32decode
The quick brown fox jumps over the lazy dog
>>> base16decode=base64.b16decode(miwen2)
>>> print base16decode
The quick brown fox jumps over the lazy dog
>>> _
```

加密部分

解密部分

3.shellcode 编码

源文本: `The quick brown fox jumps over the lazy dog`

编码后:

预览源代码打印关于

```
1 \x54\x68\x65\x7f\x71\x75\x69\x63\x6b\x7f\x62\x72\x6f\x77\x6e\x7f\x66\x6f\x78\x
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Sunnyelf\Desktop>python shellcodeencdec.py
shellcode hex encode decoder
programmer : gunslinger_ <yudha.gunslinger[at]gmail.com>
what do you want to do ? encode / decode
=> decode
Please input data : \x54\x68\x65\x7f\x71\x75\x69\x63\x6b\x7f\x62\x72\x6f\x77\x6e\x7f\x
\x6a\x75\x6d\x70\x73\x7f\x6f\x76\x65\x72\x7f\x74\x68\x65\x7f\x6c\x61\x7a\x79\x7f\x64\x
hex      => 5468657f717569636b7f62726f776e7f666f787f6a756d70737f6f7665727f7468657f6c
plaintext => The quick brown fox jumps over the lazy dog
C:\Users\Sunnyelf\Desktop>
```

4.Quoted-printable 编码

它是多用途互联网邮件扩展 (MIME) 一种实现方式。有时候我们可以邮件头里面能够看到这样的编码, 编码原理[参考](<http://blog.chacuo.net/494.html>)。

```
--_000_OC84B99647CCED41BD7BA204FDF3A9719DF17E53EX2010C500wanco_
Content-Type: text/html; charset="gb2312"
Content-Transfer-Encoding: quoted-printable

p class=3DMsoNormal><span style=3D' font-family:=CB=CE=CC=E5'>=C4=BF=C7=B0=
=CE=DE=CF=DF=D3=D0=D2=BB=B8=F6=CE=DE=CF=DF=B2=CA=C6=B1=BF=CD=BB=A7=B6=CB=BD=
=D3=C8=EB=D6=A7=B8=B6=B1=A6=C7=AE=BO=FC=CF=EE=C4=BF=A3=AC=B8=C3=CF=EE=C4=BF=
=D0=E8=C7=F3=D6=F7=D5=BE</span><span lang=3DEN-US>service</span><span style=3D' font-family:=CB=CE=CC=E5'>=BD=D3=BF=DA=CC=E1=B9=A9=D6=A7=B3=D6=
=DF=CC=E5=DD=E8=C7=F3=BC=FB=BA=BD=BC=FE=A1=A3</span><span lang=3DEN-US>h=
```



T 零零CS.Net

源文本：`敏捷的棕色狐狸跳过了懒惰的狗`

编码后：
预览源代码打印关于

1	=E6=95=8F=E6=8D=B7=E7=9A=84=E6=A3=95=E8=89=B2=E7=8B=90=E7=8B=B8=E8=B7
2	=BF=87=E4=BA=86=E6=87=92=E6=83=BO=E7=9A=84=E7=8B=97

编码解码[链接](<http://www.mxcz.net/tools/QuotedPrintable.aspx>)

5.XXencode 编码

XXencode 将输入文本以每三个字节为单位进行编码。如果最后剩下的资料少于三个字节，不够的部份用零补齐。这三个字节共有 24 个 Bit，以 6bit 为单位分为 4 个组，每个组以十进制来表示所出现的数值只会落在 0 到 63 之间。以所对应值的位置字符代替。它所选择的可打印字符是：+-

0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz，
一共 64 个字符。跟 base64 打印字符相比，就是 UUencode 多一个“-”字符，少一个”/”字符。

XXencode编码转换过程

原始字符	C							a							t				
原始ASCII码（十进制）	67							97							116				
ASCII码（二进制）	0	1	0	0	0	0	1	1	0	1	1	0	0	0	0	1	0	1	1
新的十进制数值	16							54							5			52	
编码后的XXencode字符	E							q							3			0	

字符串：'Cat ' 编码后是：Eq30

源文本：`The quick brown fox jumps over the lazy dog`

编码后：`hJ4VZ653pOKBf647mPrRi64NjSO-eRKpkQm-jRaJm65FcNG-gMLdt64FjNkc`

编码解码[链接](<http://web.chacuo.net/charsetxxencode>)

6.UUencode 编码

UUencode 是一种二进制到文字的编码，最早在 unix 邮件系统中使用，全称：Unix-to-Unix encoding，UUencode 将输入文本以每三个字节为单位进行编码，如果最后剩下的资料少于三个字节，不够的部份用零补齐。三个字节共有 24 个 Bit，以 6-bit 为单位分为 4 个组，每个组以十进制来表示所出现的字节的数值。这个数值只会落在 0 到 63 之间。然后将每个数加上 32，所产生的结果刚好落在 ASCII 字符集中可打印字符（32-空白...95-底线）的范围之中。

源文本：`The quick brown fox jumps over the lazy dog`

编码后：`M5&AE('%U:6-

K(&)R;W=N(&9O>"!J=6UP<R!O=F5R('1H92!L87IY(&109PH*`

编码解码[链接](<http://web.chacuo.net/charsetuuencode>)

7.URL 编码

url 编码又叫百分号编码，是统一资源定位(URL)编码方式。URL 地址（常说网址）规定了常用地数字，字母可以直接使用，另外一批作为特殊用户字符也可以直接用（/,;:@等），剩下的其它所有字符必须通过%xx 编码处理。现在已经成为一种规范了，基本所有程序语言都有这种编码，如 js：有 encodeURIComponent、encodeURIComponent，PHP 有 urlencode、urldecode 等。编码方法很简单，在该字节 ascii 码的 16 进制字符前面加%。如 空格字符，ascii 码是 32，对应 16 进制是'20'，那么 urlencode 编码结果是:%20。

源文本：`The quick brown fox jumps over the lazy dog`

编码后：

%54%68%65%20%71%75%69%63%6b%20%62%72%6f%77%6e%20%66%6f%78%20%6a%75%6d%70%73%20%6f%76%65%72%20%74%68%65%20%6c%61%7a%79%20%64%6f%67

编码解码[链接](<http://web.chacuo.net/charseturlencode>)

8.Unicode 编码

Unicode 编码有以下四种编码方式:

```
源文本: `The`
```

```
&x [Hex]: `&x0054;&x0068;&x0065;`
```

```
& [Decimal]: `&00084;&00104;&00101;`
```

```
\U [Hex]: `\U0054\U0068\U0065`
```

```
\U+ [Hex]: `\U+0054\U+0068\U+0065`
```

编码解码[链接](<http://www.mxcz.net/tools/Unicode.aspx>)

9.Escape/Unescape 编码

Escape/Unescape 加密解码/编码解码,又叫%u 编码,采用 UTF-16BE 模式, Escape 编码/加密,就是字符对应 UTF-16 16 进制表示方式前面加%u。Unescape 解码/解密,就是去掉"%u"后,将 16 进制字符还原后,由 utf-16 转码到自己目标字符。如: 字符“中”, UTF-16BE 是: “6d93”, 因此 Escape 是“%u6d93”。

源文本: `The`

编码后: `%u0054%u0068%u0065`

10.HTML 实体编码

结果	描述	实体名称	实体编号
"	quotation mark	"	"
'	apostrophe	'	'
&	ampersand	&	&
<	less-than	<	<
>	greater-than	>	>

完整编码手册[参考](http://www.w3school.com.cn/tags/html_ref_entities.html)

11. 敲击码

敲击码(*Tap code*)是一种以非常简单的方式对文本信息进行编码的方法。因该编码对信息通过使用一系列的点击声音来编码而命名，敲击码是基于 5×5 方格波利比奥斯方阵来实现的，不同点是用 *K* 字母被整合到 *C* 中。

敲击码表:

预览源代码打印关于

1	1 2 3 4 5
2	1 A B C/K D E
3	2 F G H I J
4	3 L M N O P
5	4 Q R S T U
6	5 V W X Y Z

源文本	F	O	X
位置	2,1	3,4	5,3
敲击码

12. 莫尔斯电码

摩尔斯电码(*Morse Code*)是由美国人萨缪尔·摩尔斯在 1836 年发明的一种时通时断的且通过

不同的排列顺序来表达不同英文字母、数字和标点符号的信号代码，摩尔斯电码主要由以下 5 种它的代码组成：

1. 点 (.)
2. 划 (-)
3. 每个字符间短的停顿（通常用空格表示停顿）
4. 每个词之间中等的停顿（通常用 ` / ` 划分）
5. 以及句子之间长的停顿

摩尔斯电码字母和数字对应表：

A	.-	N	-. .	.	.-.-.-	+	.-.-.	1	.-----
B	-...	O	---	,	--.-.-	_	..-.-	2	..----
C	-.-. .	P	.-.-.	:	---...	\$...-.-	3	...--
D	-..	Q	--.-	"	.-.-.	&	.-...	4-
E	.	R	.-.	'	.-----	/	-.-.	5
F	..-. .	S	...	!	.-.-.-			6	-....
G	--.	T	-	?	..--..			7	--...
H	U	..-	@	.-.-.-.			8	----..
I	..	V	...-	--			9	-----.
J	.----	W	.-.-	;	.-.-.-.			0	-----
K	-.-	X	-.-.	(.-.-.				
L	.-..	Y	-.--)	.-.-.-.				
M	--	Z	--..	=	-...-				

源文本：`THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

编码后：

.- / -.-.- -.-. -.-. / -... .-. --- .-- -. / ..-. --- -.-. / .---- ..- --- .-
-.- ... / --- ...- .-. / - / .-.. .- ---. -.-- / -.. --- --.

在线编码解码[传送门](<http://rumkin.com/tools/cipher/morse.php>)

摩尔斯电码除了能对字母数字编码以外还对一些标点符号，非英语字符进行了编码，而且还有一些特定意义的组合称为特殊符号，比如`·-·-·-·-·-·-`表达的意思是调用信号，表示“我有消息发送”。如果你感兴趣可以参考

[Wiki](<https://zh.wikipedia.org/wiki/%E6%91%A9%E5%B0%94%E6%96%AF%E7%94%B5%E7%A0%81>)。

13. 编码的故事

推荐大家去看[编码的故

事](http://wenku.baidu.com/link?url=kTrscV5j5AsZq5zvBpr2jdkEJW8LqgrkkKsdLwWA3YlXmgeqh_be95nMxqbFPOYoVBVy3A6lutlcXVDYLdZ-3iRawJpcOVZ71as07FnxtGS)一文。

各种文本加密

文本加密可以将正常文本内容打乱为不可连读的文字或符号(汉字 数字 字母 音乐符号 国际音标 盲文 韩文 日文 傣文 彝文 箭头符号 花朵符号 俄文)，换行等格式信息也会被清除，达到加密的作用。在进行文本加密时可以设定一个密码，这样只有知道密码的人才能解密文本。密码可以是数字、字母和下划线，最多九位。

加密示例：

源文本：`敏捷的棕色狐狸跳过了懒惰的狗`

文本加密为盲文

敏捷的棕色狐狸跳过了懒惰的狗

加密 解密 ☐ 使用密码

[illegible]

加密：文本框输入原始文本，使用密码则在密码框中设定一个密码，点击加密按钮，下方将显示加密后的文本。

解密: 文本框输入加密文本, 如果有密码则在密码框中输入加密密码, 点击解密按钮, 下方将显示解密后的文本

这个文本加密和解密工具可以将正常文本内容打乱为不可连读的文字或符号，换行等格式信息也会被清除，达到加密，这样只有知道密码的人才能解密文本。密码可以是数字、字母和下划线，最多九位。

将文本加密为以下字符（密文为不可连读的指定字符）：

汉字 数字 字母 音乐符号 国际音标 盲文 韩文 日文 傣文 彝文 箭头符号 花朵符号 俄文

编码解码[链接](<http://www.qqxiuzi.cn/bianma/wenbenjiamei.php>)

换位加密

1. 栅栏密码

**** (1) 介绍 ****

栅栏密码(*Rail-fence Cipher*)就是把要加密的明文分成 N 个一组，然后把每组的第 1 个字符组合，每组第 2 个字符组合...每组的第 N (最后一个分组可能不足 N 个)个字符组合，最后把他们全部连接起来就是密文，这里以 2 栏栅栏加密为例。

明文: `The quick brown fox jumps over the lazy dog`

去空格: `Thequickbrownfoxjumpsoverthelazydog`

分组: `Th eq ui ck br ow nf ox ju mp so ve rt he la zy do g`

第一组: `Teucbonojmsvrhlzdg`

第二组: `hqikrwxupoeteayo`

密文: `Teucbonojmsvrhlzdghqikrwxupoeteayo`

加解密[传送门](<http://www.practicalcryptography.com/ciphers/classical-era/rail-fence/>)

2. 曲路密码

曲路密码(*Curve Cipher*)是一种换位密码，需要事先双方约定密钥(也就是曲路路径)。

明文: `The quick brown fox jumps over the lazy dog`

填入 5 行 7 列表(事先约定填充的行列数)

T	h	e	q	u	i	c
k	b	r	o	w	n	f
o	x	j	u	m	p	s
o	v	e	r	t	h	e
l	a	z	y	d	o	g

加密的回路线(事先约定填充的行列数)

T	h		e	q	u	i		c
k	b		r	o	w	n		f
o	x		j	u	m	p		s
o	v		e	r	t	h		e
l	a		z	y	d	o		g

密文: `gesfc inpho dtmwu qoury zejre hbxva lookT`

3.列移位密码

** (1) 介绍**


列移位密码(Columnar Transposition Cipher)是一种比较简单, 易于实现的换位密码, 通过一个简单的规则将明文打乱混合成密文。下面我们以明文 *The quick brown fox jumps over the lazy dog*, 密钥 *how are u* 为例:

填入 5 行 7 列表(事先约定填充的行列数, 如果明文不能填充完表格可以约定使用某个字母进行填充)

T	h	e	q	u	i	c
k	b	r	o	w	n	f
o	x	j	u	m	p	s
o	v	e	r	t	h	e
l	a	z	y	d	o	g

密钥: `how are u`

按 *how are u* 在字母表中的出现的先后顺序进行编号, 我们就有 *a* 为 1, *e* 为 2, *h* 为 3, *o* 为 4, *r* 为 5, *u* 为 6, *w* 为 7, 所以先写出 *a* 列, 其次 *e* 列, 以此类推写出的结果便是密文:

	h 3	o 4	w 7	a 1	r 5	e 2	u 6
T	h	e	q	u	i		
k	b	r	o	w	n		
o	x	j	u	m	p		
o	v	e	r	t	h		
l	a	z	y	d	o		

密文: `qoury inpho Tkool hbxva uwmtd cfseg erjez`

这里提供一个行列数相等的填充规则列移位密码加解密[链接](<http://www.practicalcryptography.com/ciphers/classical-era/columnar-transposition/>)

另外由列移位密码变化来的密码也有其他的, 比如[Amsco 密码](<http://www.thonky.com/kryptos/amsc0-cipher>)(Amsco Cipher)和[Cadenus 密码](<http://www.thonky.com/kryptos/cadenus-cipher>)(Cadenus Cipher)。

替换加密

1.埃特巴什码

** (1) 介绍**

埃特巴什码(Atbash Cipher)是一种以字母倒序排列作为特殊密钥的替换加密, 也就是下面的对应关系:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
 ZYXWVUTSRQPONMLKJIHGFEDCBA

明文: `the quick brown fox jumps over the lazy dog`

密文: `gsv jfrxp yildm ulc qfnkh levi gsv ozab wlt`

加解密[传送门](<http://www.practicalcryptography.com/ciphers/classical-era/atbash-cipher/>)

2.凯撒密码

** (1) 介绍**

凯撒密码(*Caesar Cipher* 或称恺撒加密、恺撒变换、变换加密、位移加密)是一种替换加密，明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。例，当偏移量是 3 的时候，所有的字母 A 将被替换成 D，B 变成 E，以此类推，更多[参考](https://en.wikipedia.org/wiki/Caesar_cipher)。

加密实例：

明文：`The quick brown fox jumps over the lazy dog`

偏移量：1

密文：`Uif rvjdl cspxo gpy kvnqt pwfs uif mbaz eph`

Caesar encryption

Input text:

The quick brown fox jumps over the lazy dog



Transformation:

Transformation	Transformed text
ROT0	The quick brown fox jumps over the lazy dog
ROT1	Uif rvjdl cspxo gpy kvnqt pwfs uif mbaz eph
ROT2	Vjg swkem dtqyp hqz lworu qxgt vjg ncba fqi
ROT3	Wkh txlfn eurzq ira mxpsv ryhu wkh odc b grj
ROT4	Xli uymgo fvsar jsb nyqtw sziv xli pedc hsk
ROT5	Ymj vznhp gwtbs ktc ozrux tajw ymj qfed itl
ROT6	Znk waoiq hxuct lud pasvy ubkx znk rgfe jum
ROT7	Aol xbpjr iyvdu mve qbtwz vcly aol shgf kvn
ROT8	Bpm ycqks jzwev nwf rcuxa wdmz bpm tihg lwo

你也可以使用 *Python* 的 *pycipher* 模块来加解密，如果提示没有这个模块可以通过`pip install pycipher`或者其他方式来安装 *pycipher* 模块。

预览源代码打印关于

```
1 >>> from pycipher import Caesar
2 >>> Caesar(key=1).encipher('The quick brown fox jumps over the lazy dog')
3 'UIFRVJDLCSXPXOGPYKVNQTPWFSUIFMBAZEPIH'
4 >>>
```

	<code>Caesar(key=1).decipher('UIFRVJDLCSXPQGYKVNQTPWFSUIFMBAZEPH')</code>
5	<code>'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'</code>

参考表(这里是向后移位加密, 向前移位解密):

凯撒密码位移规律表

	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	26
1	Z	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	25
2	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	24
3	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	23
4	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	V	22
5	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	21
6	U	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	20
7	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	19
8	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	R	18
9	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	17
10	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	16
11	P	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	15
12	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	14
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	13
14	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	12
15	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	11
16	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	10
17	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	9
18	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	8
19	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	7
20	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	6
21	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	E	5
22	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	4
23	D	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	3
24	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	2
25	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	A	1
26	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

加密顺序:即左移.把每个字母都按字母表中的顺序依次前移n个字母即可.

例:(移1位) A=Z, B=A, C=B.

解密顺序:即右移.把每个字母都按字母表中的顺序依次后移n个字母即可.

例:(移1位) A=B, B=C, C=D.

英文字母的移位以移25位为一个循环,移26位等于没有移位.

丁零零 CS.Net

加密解密[链接](<http://planetcalc.com/1434/>)(这个网站可以将 26 种情况一次性列举出来, 比较方便)

3.ROT5/13/18/47

** (1) 介绍**

ROT5/13/18/47 是一种简单的码元位置顺序替换暗码。此类编码具有可逆性, 可以自我解密, 主要用于应对快速浏览, 或者是机器的读取。

> ROT5 是 rotate by 5 places 的简写, 意思是旋转 5 个位置, 其它皆同。下面分别说说它

们的编码方式:

>

> **ROT5**: 只对数字进行编码, 用当前数字往前数的第 5 个数字替换当前数字, 例如当前为 0, 编码后变成 5, 当前为 1, 编码后变成 6, 以此类推顺序循环。

>

> **ROT13**: 只对字母进行编码, 用当前字母往前数的第 13 个字母替换当前字母, 例如当前为 A, 编码后变成 N, 当前为 B, 编码后变成 O, 以此类推顺序循环。

>

> **ROT18**: 这是一个异类, 本来没有, 它是将 ROT5 和 ROT13 组合在一起, 为了好称呼, 将其命名为 ROT18。

>

> **ROT47**: 对数字、字母、常用符号进行编码, 按照它们的 ASCII 值进行位置替换, 用当前字符 ASCII 值往前数的第 47 位对应字符替换当前字符, 例如当前为小写字母 z, 编码后变成大写字母 K, 当前为数字 0, 编码后变成符号_。用于 ROT47 编码的字符其 ASCII 值范围是 33-126, 具体可参考 ASCII 编码, 下面以 rot13 为例。

明文: `the quick brown fox jumps over the lazy dog`

密文: `gur dhvpx oebja sbk whzcf bire gur ynml qbt`

[传送门](<http://www.qqxiuzi.cn/bianma/ROT5-13-18-47.php>)

4.简单替换密码

**** (1) 介绍****

简单换位密码(Simple Substitution Cipher)加密方式是以每个明文字母被与之唯一对应且不同的字母替换的方式实现的, 它不同于恺撒密码, 因为密码字母表的字母不是简单的移位, 而是完全是混乱的。 比如:

明文字母: *abcdefghijklmnopqrstuvwxyz*

明文字母: *phqgiumeaylnofdxjkrvcstzwb*

明文: `the quick brown fox jumps over the lazy dog`

密文: `cei jvaql hkdtf udz yvoxr dsik cei npbw gdm`

**** (2) 破解****

当密文数据足够多时这种密码我们可以通过字频分析方法破解或其他方法破解，比较好的在线词频分析网站<http://quipqiup.com/index.php>(翻=墙)，这里推荐一篇通过“爬山算法”来破解简单替换密码[文章](<http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-simple-substitution-cipher/>)，基于文中的算法实现的工具来破解示例。

密文：

pmpafxaikkkitprdsikcplifhwceigixkirradfeirdgkipgigudkcekiigpwrpucikceiginasikw
duearrxiiepcceindgmieinpwdfprduppcedoikiqiasafmfddfipfgmdafmfdeiki

解密：

```
python break_simplesub.py

C:\python_cryptanalysis>python break_simplesub.py
Substitution Cipher solver, you may have to wait several iterations
for the correct result. Press ctrl+c to exit program.

best score so far: -773.282341503 on iteration 1
  best key: EUAWRHZSNYLKQMFJJPIDGOTBCV
  plaintext: RNRCPCSLLSWRETHSLYRKSOFDYASUSPLSEECTOASETULSRUSUBTLYALSSURDERBYSLYASUS
RETBRRYATVSLSMSCHCONOTTOSROUTCONOTWASLS

best score so far: -769.470111922 on iteration 2
  best key: FNMTPWHQGYORSCKXZDEIUJVLAB
  plaintext: ECEYAPYTOOTDELRMTONEXTAGFNSTITPOTLLYRASTLRIOTEITIURONSOTTIEFLEUNTONSTIT
ELRUEENSRKTOTHTYMYACARRATEAICRYACARDSTOT

best score so far: -761.922058483 on iteration 3
  best key: FYUTROLAGBHDWMKXJPEICQNZSV
  plaintext: RNRHAPHTOOTDRELYTOURGTAKMUSTITPOTEEHLASTELIOTRITICLOUSOTTIRMERCUTOUSTIT
RELCCRUSLFTOTVTHYHANALLATRAINLHANALDSTOT

best score so far: -590.058647248 on iteration 4
  best key: PHQGIUMEAVLNOFDXBKRCZSTJWY
  plaintext: AGAINPIERREWASOVERTAKENBYTHEDEPRESSIONHESODREADEDFORTHREEDAYSAFTERTHEDE
ASOFAATHOMERECEIVINGNOONEANDGOINGNOWHERE

微软拼音 半：
```

(ps:score 值越小越准确)

密钥: `PHQGIUMEAVLNOFDXBKRCZSTJWY`

明文:

AGAINPIERREWASOVERTAKENBYTHEDEPRESSIONHESODREADEDFORTHRE
EDAYSATERTHEDELIVERYOFHISSPEECHATTHELODGEHELAYONASOFAATHO
MERECEIVINGNOONEANDGOINGNOWHERE

将明文转换成可读句子:

again pierre was over taken by the depression he so dreaded for three day safter
the delivery of his speech at the lodge he lay on a sofa at home receiving no one
and going no where

5.希尔密码

** (1) 介绍**

希尔密码(Hill Cipher)是基于线性代数多重代换密码, 由 *Lester S. Hill* 在 1929 年发明。每个字母转换成 26 进制数字: $A=0, B=1, C=2...Z=25$ 一串字母当成 n 维向量, 跟一个 $n \times n$ 的矩阵相乘, 再将得出的结果 $MOD 26$ 。更多[参考](https://en.wikipedia.org/wiki/Hill_cipher)

** (2) 加密**

明文: `ACT`

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

明文对应矩阵:

$$\begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix}$$

加密密钥: `GYBNQKURP`

加密矩阵:

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}$$

计算过程:

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix} = \begin{pmatrix} 67 \\ 222 \\ 319 \end{pmatrix} \equiv \begin{pmatrix} 15 \\ 14 \\ 7 \end{pmatrix} \pmod{26}$$

密文: `FIN`

**** (3) 解密 ****

密文: `FIN`

计算加密矩阵的逆矩阵:

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}^{-1} \equiv \begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix} \pmod{26}$$

解密计算:

$$\begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix} \begin{pmatrix} 15 \\ 14 \\ 7 \end{pmatrix} \equiv \begin{pmatrix} 260 \\ 574 \\ 539 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 2 \\ 19 \end{pmatrix} \pmod{26}$$

明文: `ACT`

至于证明和求逆可以参考线性代数知识。

** (4) 破解**

密码分析一门破解编码和密码的艺术。当我们尝试去攻破希尔密码你会发现频率分析实际上没有什么用处，特别在密钥长度增多的情况下。对于较长的二元矩阵（ 2×2 的希尔密码）频率分析可能可能会有帮助，但是对于较短的密文分析是没有实际作用的。

这里推荐一篇关于用[已知明文样本攻击的方式破解希尔密

码]([http://www.practicalcryptography.com/cryptanalysis/stochastic-](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-hill-cipher/)

searching/cryptanalysis-hill-cipher/)的文章，基础的希尔密码用[已知明文攻

击](https://en.wikipedia.org/wiki/Known-plaintext_attack)的方式是可攻破的，由于加密完全是线性的，所以攻击者在截取到部分明文/密文字符对可以轻松建立一个线性系统，轻松搞定希尔密码，如果不能完全确定线性系统，那么只需要添加部分明文/密文对即可。已知明文攻击时最好的方式去破解写入密码，如果明文一无所知，那就进行推测猜出部分明文。基于已知明文样本攻击的方式破解希尔密码的算法的实现工具破解示例：

密文：

XUKEXWSLZJUAXUNKIGWFSOZRAWURORKXAOSLHROBxBTKCMUWDVPTF
BLMKEFVWMUXTVTWUIDDJVZKBRMCWOIWYDXMLUFPVSHAGSVWUFWORCWU
IDUJCNVTTBERTUNOJUZHVTWKORSVRZSVVFSQXOCMUWPYTRLGBMCYPOJCL
RIYTVFCCMUWUFPOXCNMCIWMSKPxEDLYIQKDJIWCJUMVRCJUMVRKXWUR
KPSEIwZVXULEIOETOOFWKBIUXPXUGOWLFPWUSCH

解密：

解密[脚本实例](<http://bobao.360.cn/ctf/learning/136.html>)

在线加解密[传送门](<http://www.practicalcryptography.com/ciphers/hill-cipher/>)

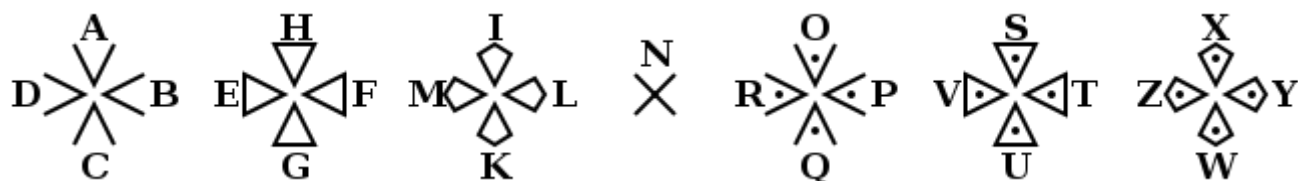
6. 猪圈密码

** (1) 介绍**

** (2) 变种 **

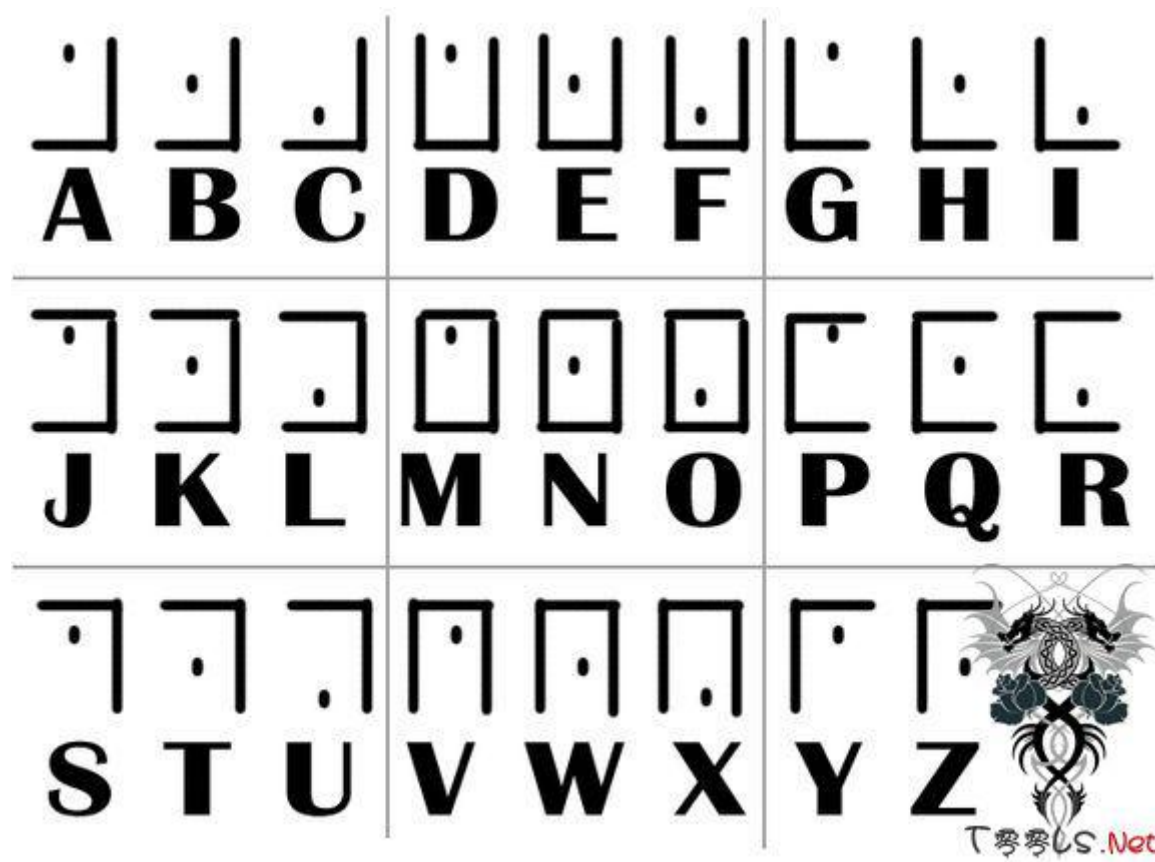
圣堂武士密码(Templar Cipher)是共济会的“猪圈密码”的一个变种，一直被共济会圣殿骑士用。

明文字母和对应密文：

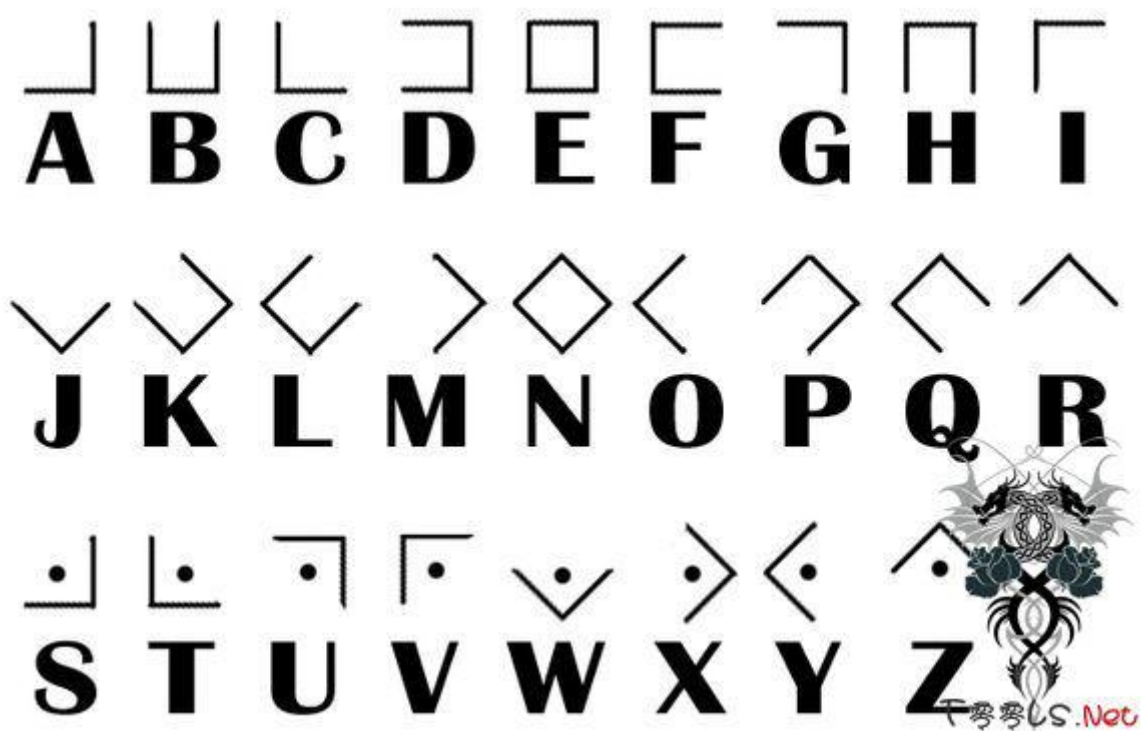


** (3) 其他变种 **

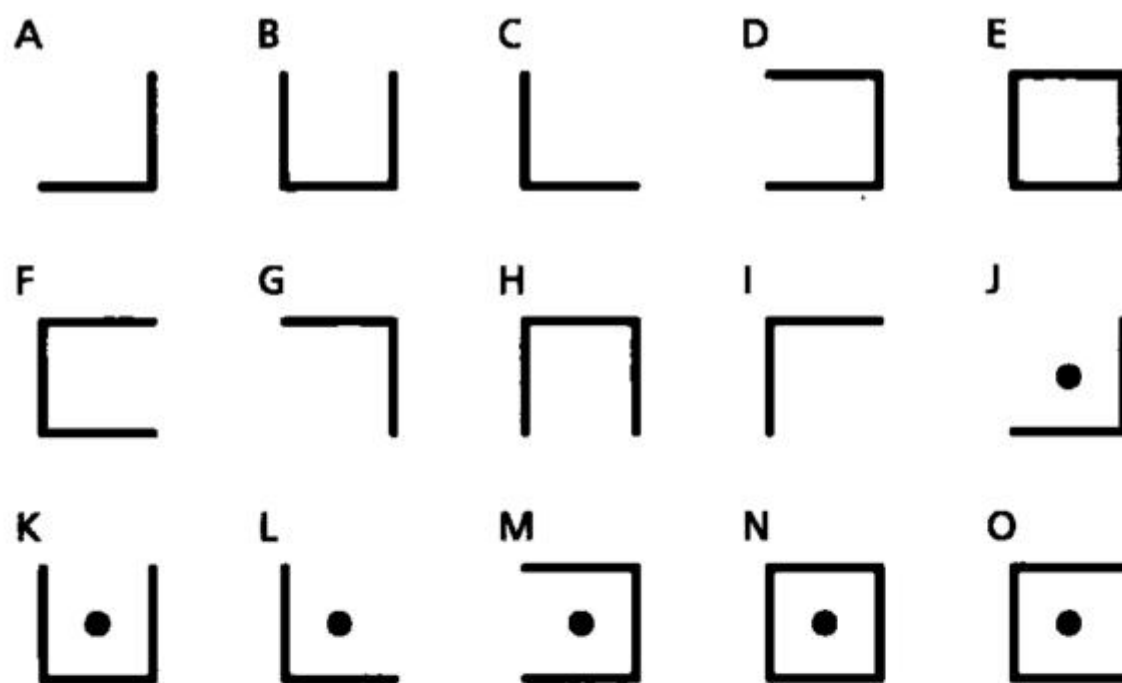
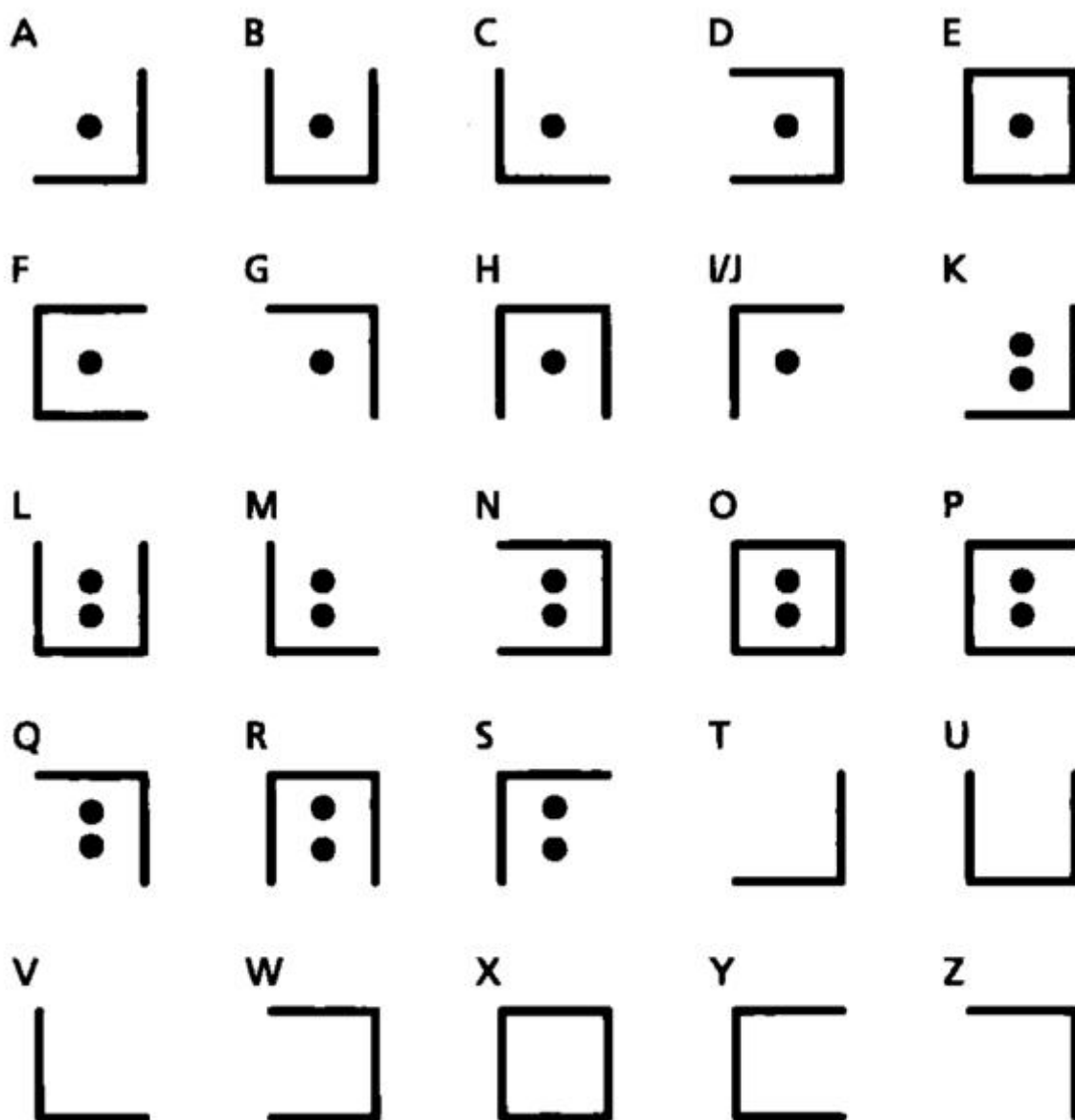
明文字母和对应密文：



明文字母和对应密文：



明文字母和对应密文：



7.波利比奥斯方阵密码

** (1) 介绍**

波利比奥斯方阵密码 (Polybius Square Cipher 或称波利比奥斯棋盘) 是棋盘密码的一种, 是利用波利比奥斯方阵进行加密的密码方式, 简单的来说就是把字母排列好, 用坐标(行列)的形式表现出来。字母是密文, 明文便是字母的坐标。更多[参考](https://en.wikipedia.org/wiki/Polybius_square)

常见的排布方式:

	1	2	3	4	5
1	A	B	C	D	E
2	F	G			
3	L	M			
4	Q	R			
5	V	W			
		X	Y	Z	

加密实例:

明文: `THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

密文: `442315 4145241325 1242345233 213453 2445323543 442315 31115554 143422`

8.夏多密码(曲折加密)

** (1) 介绍**

夏多密码是作者麦克斯韦·格兰特在中篇小说《死亡之链》塑造夏多这一英雄人物中所自创的密码, 如下图所示:

A



B



C



D



E



F



G



H



I



J



K



L



M



N



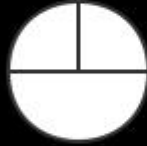
O



P



Q



R



S



T



U



V



W



X



Y



Z



1



2



3



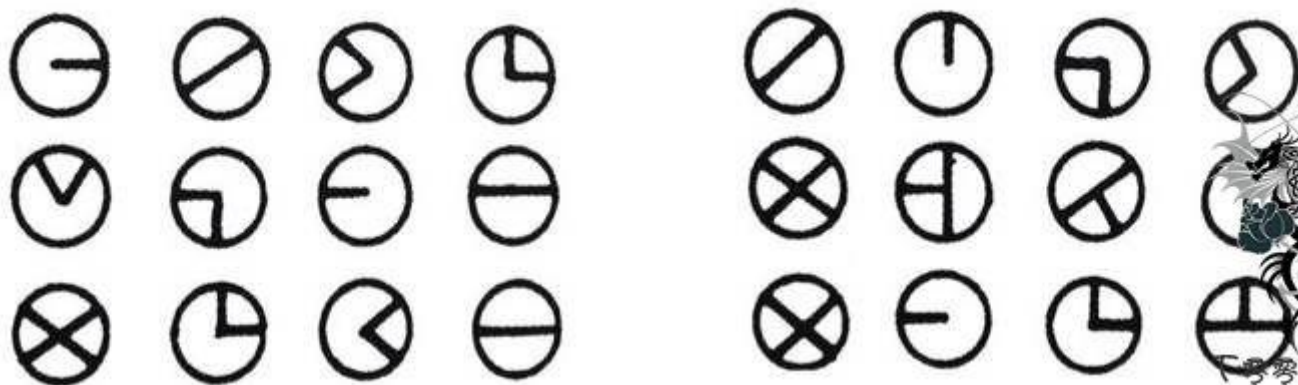
Make by 4ido1

トモトモS.Net

- > 注意，在以上所示的字母表密钥的底部，列有四个附加符号 1, 2, 3, 4. 他们可以放在密文中的任何地方。每个附加符号指示，如何转动写有密文的纸张，再进行后续的加密或解密操作，直到出现另一个附加符号。可以把每个附加符号中的那根线看作是指针，它指示了纸张的上端朝上，朝右，朝下，朝左。比如说：如果出现符号 3，那么纸张就应该转动 180 度，使其上端朝下；
- > 符号 2 表示纸张上端朝右，依次类推。

源文本：`I AM IN DANGER SEND HELP(我有危险，速来增援)`

密文：



9. 普莱菲尔密码

普莱菲尔密码(Playfair Cipher)是第一种用于实际的双字替换密码，用双字加密取代了简单代换密码的单字加密，很明显这样使得密文更难破译，因为使用简单替换密码的频率分析基本没有什么作用，虽然频率分析，通常仍然可以进行，但是有 $25 \times 25 = 625$ 种可能而不是 25 种可能，可以分为三个步骤，即编制密码表、整理明文、编写译文，下面我们以明文：


`THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`和密钥`CULTURE`为例来讲解。普莱菲尔密码又称为单方密码(Single Cipher)之后又出现它的升级版 Double Playfair，也就是[二方密码](https://en.wikipedia.org/wiki/Two-square_cipher)(Two-square Cipher),在之后又有四方密码(Four-square Cipher)

****(1)编制密码表****

1.整理密钥字母`C U L T U R E`，去掉后面重复的字母得到：`C U L T R E`

2.用上一步得到的字母自上而下来填补 5 乘 5 方表的纵列（也可横排），之后的空白按照相同的顺序用字母表中剩余的字母依次填补完整，得到如下的方格：

	1	2	3	4	5
1	C	E	G	N	
2	U	A	H	O	
3	L	B	I/J	P	
4	T	D	K	Q	
5	R	F	M	S	



> 这一步需要注意的要点：整理密钥字母时，如果出现“Z”，则需要去除，因为在英文里“Z”的使用频率最低，相应的如果是德文，则需将“l”与“J”当作一个字母来看待，而法语则去掉“W”或“K”。

** (2) 整理明文 **

我们要遵循的原则是“两个一组”，得到是若干个两两成对的字母段，用到的是明文`THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`与字母“X”：

1.将明文两两一组按顺序排开，得到：`TH EQ UI CK BR OW NF OX JU MP SO VE RT HE LA ZY DO G`

2.对于末尾的单个字母要加上一个“X”使之成对：`TH EQ UI CK BR OW NF OX JU MP SO VE RT HE LA ZY DO GX`

> 这一步需要注意的要点：对于相连字母相同者，每个后面都需要加“X”，例如`TOMORROW`，需要写成：`TO MO RX RX OW`。

** (3) 编写密文 **

我们要得到的密文，当然，对于每个字母对，要严格遵循如下的原则：

> 1.如果两个字母在同一行则要用它右邻的字母替换，如果已在最右边，则用该行最左边的替换，如明文为“CE”，依据上表，应替换为“EQ”；

> 2.如果两个字母在同一列则要用它下边的字母替换，如果已在最下边，则用该行最上边的替换，如明文为“OQ”，依据上表，应替换为“PS”；

> 3.如果两个字母在不同的行或列，则应在密码表中找两个字母使四个字母组成一个矩形，明文占据两个顶点，需用另外两个顶点的字母替换，如明文为“HX”，可以替换为“WI/J”或“I/JW”（下面的例子将按照横向替换原则即同行优先）。

按照上述原则，将明文`TH EQ UI CK BR OW NF OX JU MP SO VE RT HE LA ZY DO GX`加以转换得到`KU ND LH GT LF WU ES PW LH SI/J NP CQ CR AG BU VZ QA I/JV`（/表示或者，不过一般用 I 不用 J，所以分析密文时你看 25 个字母都有而只差一个字母没有用到可以考虑一下这种加密方式）将得到的字母改为大写并五个一组列好，得到密文`KUNDL HGTLF WUESP WLHSI NPCQC RAGBU VZQAI V`。

加密解密[传送门](<http://www.practicalcryptography.com/ciphers/classical-era/playfair/>)(ps: 这里加解密是横向编制密码表)

加密解密实例(ps: 这里加解密也是横向编制密码表):

预览源代码打印关于

1	>>>from pycipher import Playfair
2	>>>Playfair('CULTREABDFGHIKMNOPQSVWXYZ').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')
3	'UKDNLHTGFLWUSEPWHLISNPCQCRGAUBVZAQIV'
4	>>>Playfair('CULTREABDFGHIKMNOPQSVWXYZ').decipher('UKDNLHTGFLWUSEPWHLISNPCQCRGAUBVZAQIV')
5	'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOGX'

10.维吉尼亚密码

** (1) 介绍**

维吉尼亚密码(Vigenère Cipher)是在单一恺撒密码的基础上扩展出多表代换密码，根据密钥(当密钥长度小于明文长度时可以循环使用)来决定用哪一行的密表来进行替换，以此来对抗字频统计，更多[参考](https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher)。

密表:

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

明文: `THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

密钥(循环使用，密钥越长相对破解难度越大): `CULTURE`

加密过程：如果第一行为明文字母，第一列为密钥字母，那么明文字母'T'列和密钥字母'C'行的交点就是密文字母'V'，以此类推。

密文：`VBP JOZGM VCHQE JQR UNGGW QPPK NYI NUKR XFK`

****（2）已知密钥加解密****

预览源代码打印关于

1	>>>from pycipher import Vigenere
2	>>>Vigenere('CULTURE').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')
3	'VBPJOZGMVCHQEJQRUNGQWQPPKNYINUKRXFK'
4	>>>Vigenere('CULTURE').decipher('VBPJOZGMVCHQEJQRUNGQWQPPKNYINUKRXFK')
5	'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'

在线加密解密[传送门](<http://planetcalc.com/2468/>)

****（3）未知密钥破解****

可以参考[维吉尼亚密码分

析](<http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher/>)这篇文章，破解维吉尼亚密码第一步是确定密钥长度，维吉尼亚密码分析这篇文章里介绍了使用[重合指数](https://en.wikipedia.org/wiki/Index_of_coincidence)算法来确定密钥长度，在确定密钥长度后就可以尝试确定密钥，通常我们可以使用[卡方检验](https://en.wikipedia.org/wiki/Chi-squared_test)来找到每个字母的偏移量，基于维吉尼亚密码分析一文中的算法实现的工具破解示例。

密文：`kiqpbkxspshwehospzqhoinlgapp`

解密：

C:\WINDOWS\system32\cmd.exe

```
C:\python_cryptanalysis>python break_vigenere.py
-142.899449737 Vigenere, klen 3 : "PHL", VBFAUZILEDALPADDIOBADTGARTEA
-143.174707481 Vigenere, klen 4 : "BGJH", JCHIAEOLOMYPDBFLOTHANCEEFUGI
-96.1817289368 Vigenere, klen 5 : "HELLO", DEFENDTHEEASTWALLOFTHECASTLE
-122.036509523 Vigenere, klen 6 : "MHWEVW", YBULGOLLTOMASASOUDEASESPUTTL
-128.135339377 Vigenere, klen 7 : "OADLWWL", WINEFOMEPPWAIWASMOULDUNIVETE
-116.700505383 Vigenere, klen 8 : "CGJDXHVE", ICHMEDCONMYTHATONTHERBSHEUGM
-113.686233035 Vigenere, klen 9 : "ZAORBXUWL", LICYANDWETHINGRYTORHARMOMEEQ
-96.1817289368 Vigenere, klen 10 : "HELLOHELLO", DEFENDTHEEASTWALLOFTHECASTLE
-103.213291701 Vigenere, klen 11 : "WROHOEVEDOP", ORCINGCOMESANTHELEMEATRUSTBL
-105.954980587 Vigenere, klen 12 : "CVKBWRQOXEUE", INGOFTHESONSCMERTIATRETHEFFO
-95.0914490799 Vigenere, klen 13 : "WOZLIDTBPGHDA", OURETHERAMATELATERNONTHEDATB
-96.7165056538 Vigenere, klen 14 : "GFJHJDXAASZJLA", EDHISHASPAINTHINGSHERINTHREP
-90.717335872 Vigenere, klen 15 : "ZIVLTXELLPHDLAK", LAVEINTHEDATTHETHEFORECARTME
-92.6918459999 Vigenere, klen 16 : "PIXHNFSCPHJBOHO", VATIONSANDANDTHEARTABLITELIG
-91.1050618778 Vigenere, klen 17 : "RDZBPLRLBHJBOHOK", TFROMEMBERANDTHEFININTHAPPOI
-85.7948458835 Vigenere, klen 18 : "XAKIIDTAWLDELAKQBM", NIGHTESTHESTHECONTHEAFINATE
-87.0421774808 Vigenere, klen 19 : "ZBQUXTELPZZIROHOYVN", LHAVERTHATIONTHEREDINSTONWEA

C:\python_cryptanalysis>
```

(ps:结合左边的值，密钥以及解出明文可以确定 *kien 5* 或者 *klen 10* 为准确的结果)

明文: `DEFEND THE EAST WALL OF THE CASTLE`

** (4) 变种**

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

有几种密码和维吉尼亚密码相似，格罗斯费尔德密码(*Gronsfeld cipher*)实际上和维吉尼亚密码相同，除了使用了数字来代替字母以外没有什么区别。数字可以选择一种数列，如斐波那契数列，或者一些其他的伪随机序列。格罗斯费尔德密码密码分析过程和维吉尼亚密码大同小异，不过，自动密钥密码不能使用[卡西斯基算

法](<http://www.zybang.com/question/a0a1108423f63d10dbbf0c3e1bfdf3b3.ht>

ml)(*kasiski*)来破译，因为自动密钥密码的密钥不重复循环使用，破译自动密钥密码最好的方法的就是从密文不断尝试和猜测其中明文或密钥的一部分。

预览源代码打印关于

1	>>>from pycipher import Gronsfeld
2	>>>Gronsfeld([2,20,11,45,20,43,4]).encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')
3	VBPJOZGQMVCHQEJQRUNGQWQPPKNYINUKRXFK'
4	>>>Gronsfeld([2,20,11,45,20,43,4]).decipher('VBPJOZGQMVCHQEJQRUNGQWQPPKNYINUKR')
5	THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'

在线加解密[传送门](<http://rumkin.com/tools/cipher/gronsfeld.php>)

11. 自动密钥密码

** (1) 介绍**

自动密钥密码(*Autokey Cipher*)是多表替换密码，与维吉尼亚密码密切相关，但使用不同的方法生成密钥，通常来说要比维吉尼亚密码更安全。自动密钥密码主要有两种，关键词自动密钥密码和原文自动密钥密码。下面我们以关键词自动密钥为例：

明文：`THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

关键词：`CULTURE`

自动生成密钥：`CULTURE THE QUICK BROWN FOX JUMPS OVER THE`

接下来的加密过程和维吉尼亚密码类似，从密表可得：

密文：`VBP JOZGD IVEQV HYY AIICX CSNL FWV ZVDP WVK`

** (2) 已知关键词加解密**

预览源代码打印关于

1	>>>from pycipher import Autokey
2	>>>Autokey('CULTURE').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')
3	'VBPJOZGDIVEQVHYYAIICXCSNLFWVZVDPWVK'
4	>>>Autokey('CULTURE').decipher('VBPJOZGDIVEQVHYYAIICXCSNLFWVZVDPWVK')

5 'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'

在线加解密[传送门](<http://www.practicalcryptography.com/ciphers/classical-era/autokey/>)

**** (3) 未知关键词破解 ****

推荐去看这篇[自动密钥密码分析文

章](<http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-autokey-cipher/>), 基于文中的算法实现的工具来破解示例。

密文:

isjiqymdebvuzrvwhmvysibugzhyinmiyeiklcvioimbninyksmmnjmgalvimlhspjxmgfi
raqlhjcpvolqmnyynhpdetoxemgnoxl

解密

```
C:\WINDOWS\system32\cmd.exe

C:\python_cryptanalysis>python break_autokey.py
-691.381143727 autokey, klen 3 : "CIH", GKCCGWKXIRYMITJODHKPBRFFICTALTINLAXACYIMKEPDEY
-676.13815219 autokey, klen 4 : "KZFW", YTEMSFIRMWNDNVITURNFYROPIITJAFTZYZPLNDGXBFGEMDH
-668.340118608 autokey, klen 5 : "XYREZ", LUSERNSLAKICORLOFYENEDDQTVEVSUREDmothzjuvbnst
-650.874870922 autokey, klen 6 : "VTXEVQ", NZMEVIZESXAMANDZHAVLPJUULOSPOTBUGPURKIPTURCT
-622.932210312 autokey, klen 7 : "WSKXQKG", MAZLAOGRECKULLESFCBNHEJPEYUREEXEAKRGHFRIERG
-614.8274635 autokey, klen 8 : "ZPSOILKP", JDRUINCOVYEARETIMORYBEIMULQAHJEWETSKETRMKPUR
-614.82969963 autokey, klen 9 : "BEPVSFSSX", HOUNYTULHUHAMTCCWFBRWISEDcxrvqagafiolasoc
-609.364325115 autokey, klen 10 : "TGFVYGSRPJ", PMENSSUMPSGIVEDENAGGMAGQDVUYCHAIsofpret
-572.296404606 autokey, klen 11 : "ANJVNJIAQKR", IFANDPEDOREMURITSISKBEAPROGAVCHUPIVUO
-578.496461761 autokey, klen 12 : "PJSODFSZFTGG", TJRUNTUEZIPOGIECUTBUTAMGARDWOULOFEWEL
-431.935686608 autokey, klen 13 : "FORTIFICATION", DESPITEBEINGMORESECURETHANTHEVIGENER
-569.30241445 autokey, klen 14 : "VOQQOYIHDPRVSO", NETSCAEWBMEZHDISOUTYOMAICAavavyofguy
-552.454159723 autokey, klen 15 : "KZCYMZIVRVIVURV", YTHKEZEINGNZFAAYOFUTETHAMITINOUTT
-546.828526815 autokey, klen 16 : "IKDIPQYRMTSSZPKY", AIGABIOMSIDCACLYHEPYRANIOREWILBKR
-521.482309988 autokey, klen 17 : "AZEUZJGCRUKQHFRMU", ITFORPGBNHLESMEKNECTERMofmaneVAE
-526.172419614 autokey, klen 18 : "UAFEMGURQKPBFARETU", OSEEESSMORGTURESOSHGOEXCONTHCUS
-529.449159901 autokey, klen 19 : "CEWLIGAPXXAMRENJXTB", GONXISMOHEVIININKTUSEVEMONTRES

C:\python_cryptanalysis>
```

(ps:从 klen 13 可以看出使用的关键词为'FORTIFICATION')

明文:

DESPITEBEINGMORESECURETHANTHEVIGENERECIPHERTHEAUTOKEYCIPHERISSTILLVERYEASYTOBREAKUSINGAUTOMATEDMETHODS

将明文转换成可读句子：

despite being more secure than the vigenere cipher the autokey cipher is still very easy to break using automated methods

12.博福特密码

（1）介绍

博福特密码(Beaufort Cipher)，是一种类似于维吉尼亚密码的代换密码，由弗朗西斯·蒲福(Francis Beaufort)发明。它最知名的应用是 Hagelin M-209 密码机。博福特密码属于对等加密，即加密演算法与解密演算法相同。

明文：`THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

密钥(循环使用，密钥越长相对破解难度越大)：`CULTURE`

加密过程：如果第一行为明文字母，第一列为密文字母，那么沿明文字母`T`列出现密钥字母`C`的行号就是密文字母`J`，以此类推。

密文：`JNH DAJCS TUFYE ZOX CZICM OZHC BKA RUMV RDY`

（2）已知密钥加解密

预览源代码打印关于

1	
2	>>>from pycipher import Beaufort
3	>>>Beaufort('CULTURE').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')
4	'JNHDAJCS TUFYEZOX CZICM OZHC BKA RUMV RDY'
5	>>>Beaufort('CULTURE').decipher('JNHDAJCS TUFYEZOX CZICM OZHC BKA RUMV RDY')

6	'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'
---	---------------------------------------

在线加解密[传送门](<http://www.practicalcryptography.com/ciphers/classical-era/beaufort/>)

13.滚动密钥密码

**** (1) 介绍****

滚动密钥密码(*Running Key Cipher*)和维吉尼亚密码有着相同的加密机制，区别是密钥的选取，维吉尼亚使用的密钥简短，而且重复循环使用，与之相反，滚动密钥密码使用很长的密钥，比如引用一本书作为密钥。这样做的目的是不重复循环使用密钥，使密文更难破译，尽管如此，滚动密钥密码还是可以被攻破，因为有关于密钥和明文的统计分析模式可供利用，如果滚动密钥密码使用统计上的随机密钥来源，那么理论上是不可破译的，因为任何可能都可以成为密钥，并且所有的可能性都是相等的。

明文: `THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

密钥: 选取 *C 语言编程*(1978 版)第 63 页第 1 行"*errors can occur in several places. A label has...*", 去掉非字母部分作为密钥(实际选取的密钥很长，长度至少不小于明文长度)。

加密过程: 加密过程和维吉尼亚密码加密过程相同

密文: `XYV ELAEK OFQYH WWK BYHTJ OQTC TJI DAK YESR`

已知密钥在线加解密[传送

门](<http://www.practicalcryptography.com/ciphers/classical-era/running-key/>)

14.Porta 密码

**** (1) 介绍****

Porta 密码(*Porta Cipher*)是一个由意大利那不勒斯的医生 *Giovanni Battista della Porta* 发明的多表代换密码，Porta 密码具有加密解密过程的是相同的特点。

密表:

KEYS| 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

----- -----	

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

明文: `THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

密钥(循环使用, 密钥越长相对破解难度越大): `CULTURE`

加密过程: 明文字母`T`列与密钥字母`C`行交点就是密文字母`F`,以此类推。

密文: `FRW HKQRY YMFMF UAA OLWHD ALWI JPT ZXHC NGV`

已知密钥在线加解密[传送

门](<http://www.practicalcryptography.com/ciphers/classical-era/porta/>)

**** (2) 破解****

Porta 密码可以被以[维吉尼亚密

码]([http://www.practicalcryptography.com/cryptanalysis/stochastic-](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher-part-2/)

[searching/cryptanalysis-vigenere-cipher-part-2/](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher-part-2/))破解相类似方式进行自动攻破, 破解

Porta 密码第一步是先确定密钥长度, 这里推荐一篇关于使用[重合指数算

法](https://en.wikipedia.org/wiki/Index_of_coincidence)确定为维吉尼亚密钥长度[文

章]([http://www.practicalcryptography.com/cryptanalysis/stochastic-](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher/)

[searching/cryptanalysis-vigenere-cipher/](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher/))。

15.同音替换密码

** (1) 介绍**

同音替换密码(*Homophonic Substitution Cipher*)是单字母可以被其他几种密文字母同时替换的密码,通常要比标准替换密码破解更加困难,破解标准替换密码最简单的方法就是分析字母出现频率,通常在英语中字母'E'(或'T')出现的频率是最高的,如果我们允许字母'E'可以同时被 3 种不同字符代替,那么就不能还是以普通字母的频率来分析破解,如果允许可代替字符越多,那么密文就会更难破译。

常见代换规则表:

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
D	X	S	F	Z	E	H	C	V	I	T	P	G	A	Q	L	K	J	R	U	O	W	M	Y	B	N
9				7				3				5	0				4	6							
				2																					
				1																					

明文: `THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

密文(其中一种): `6CZ KOVST XJOMA EQY IOQL4 OW1J UC7 P9NB FOH`

** (2) 破解**

如果同音替换密码的同音词个数很多,那么破解它难度很大,通常的方法采取类似破解替换密码的"爬山算法",除了找到一个明文字母映射几个字符之外,我们还需要确定映射了那些字符,可以尝试[2 层嵌套"爬山算法

"](<http://www.cs.sjsu.edu/faculty/stamp/RUA/homophonic.pdf>)来破解,外层确定映射的数量,内层确定映射字符。

16.仿射密码

** (1) 介绍**

仿射密码(*Affine Cipher*)是一种单表代换密码,字母表中的每个字母相应的值使用一个简单的数学函数映射到对应的数值,再把对应数值转换成字母。这个公式意味着每个字母加密都会返回一个相同的字母,意味着这种加密方式本质上是一种标准替代密码。因此,它具有所有替代

密码的弱点。每一个字母都是通过函数 $(ax + b) \bmod m$ 加密，其中 B 是位移量，为了保证仿射密码的可逆性， a 和 m 需要满足 $\gcd(a, m)=1$ ，一般 m 为设置为 26。更多[参考](https://en.wikipedia.org/wiki/Affine_cipher)

常见的字母对应关系：

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

下面我们以 $E(x) = (5x + 8) \bmod 26$ 函数为例子

明文	T	H	E	Q	U	I	C	K	B	R	O	W	N	F	
x	19	7	4	16	20	8	2	10	1	17	14	22	13	5	
(5x + 8)	103	43	28	98	108	48	18	85	13	93	78	118	73	33	
(5x + 8)mod26	25	17	2	10	4	22	18	6	13	15	0	14	21	7	
密文	Z	R	C	K	E	W	S	G	N	P	A	O	V	H	

至于解密我们知道

$$E(x) = (ax + b) \bmod m,$$

$$1 = aa^{-1} \bmod m.$$

即可得出解密结果

$$D(x) = a^{-1}(x - b) \bmod m,$$

以 $E(x) = (5x + 8) \bmod 26$ 加密，通过计算可得 $D(x) = 21(x - 8) \bmod 26$ ，这样便可以得到明文。

可参考的 *Python* 脚本

```
# -*- coding: utf-8 -*-
#打印一个仿射密码的换位表
#a必须和m=26互素
def affine(a, b):
    for i in range(26):
        print chr(i+65) + ": " + chr(((a*i+b)%26)+65)

#一个调用例子
affine(5, 8)
```



加解密[传送门](<http://www.practicalcryptography.com/ciphers/classical-era/affine/>)

17.培根密码

** (1) 介绍 **

培根密码(Baconian Cipher)是一种替换密码，每个明文字母被一个由 5 字符组成的序列替换，最初的加密方式就是由'A'和'B'组成序列替换明文(所以你当然也可以用别的字母)，比如字母'D'替换成"aaabb"，以下是全部的对应关系(另一种对于关系是每个字母都有唯一对应序列，I和J与U/V各自都有不同对应序列)：

A = aaaaa I/J = abaaa R = baaaa

B = aaaab K = abaab S = baaab

C = aaaba L = ababa T = baaba

D = aaabb M = ababb U/V = baabb

E = aabaa N = abbaa W = babaa

F = aabab O = abbab X = babab

G = aabba P = abbba Y = babba

H = aabbb Q = abbbb Z = babbb

明文: ` T H E F O X`

密文: `baaba aabbb aabaa aabab abbab babab`

加解密[传送门](<http://rumkin.com/tools/cipher/baconian.php>)

18.ADFGX 和 ADFGVX 密码

** (1) ADFGX 密码**

ADFGX 密码(ADFGX Cipher)是结合了改良过的 *Polybius* 方格替代密码与单行换位密码的矩阵加密密码, 使用了 5 个合理的密文字母: A, D, F, G, X, 这些字母之所以这样选择是因为当转译成摩尔斯电码(ADFGX 密码是德国军队在一战发明使用的密码)不易混淆, 目的是尽可能减少转译过程的操作错误。

加密矩阵示例:

	A	D	F	G	X
A	p	h	q	g	m
D	e	a	y	n	o
F	f	d	x	k	r
G	c	v	s	z	w
X	b	u	t	i/j	l

明文: `THE QUICK BROWN FOX`

结果矩阵加密:

XF AD DA AF XD XG GA FG XA FX DX GX DG FA DX FF

列移位密钥: `how are u`

	h	o	w	a	r	e	u
	3	4	7	1	5	2	6
X	F	A	D	D	A	A	
F	X	D	X	G	G	A	
F	G	X	A	F	X		
X	G	X	D	G	F		
D	X	F	F				

T 考考 CS .Net

密文: `DXADF AGXF XFFXD FXGGX DGFG AADA ADXXF`

已知密钥加解密:

预览源代码打印关于

1	>>>from pycipher import ADFGX
2	>>>a = ADFGX('phqgmeaynofdxkrcvszwbutil','HOWAREU')
3	>>>a.encipher('THE QUICK BROWN FOX')
4	'DXADFAGXFXFFXDFXGGXDGFGAADAADXXF'
5	>>>a.decipher('DXADFAGXFXFFXDFXGGXDGFGAADAADXXF')
6	'THEQUICKBROWNFOX'

在线加解密[传送门](<http://www.practicalcryptography.com/ciphers/adfgx-cipher/>)

**** (2) ADFGVX 密码****

ADFGVX 密码实际上就是 ADFGX 密码的扩充升级版, 一样具有 ADFGX 密码相同的特点, 加密过程也类似, 不同的是密文字母增加了 V, 使得可以再使用 10 数字来替换明文。

A D F G V X


```

-----
A | p h o q g 6
D | 4 m e a 1 y
F | 1 2 n o f d
G | x k r 3 c v
V | s 5 z w 7 b
X | j 9 u t i 8

```

由于两种加密过程完全类似这里就不再重复给出加密过程。

预览源代码打印关于

1	>>>from pycipher import ADFGVX
2	>>>a = ADFGVX('phOqg64mea1y12nofdxkr3cvs5zw7bj9uti8','HOWAREU')
3	>>>a.encipher('THE QUICK BROWN FOX')
4	'DXXFAFGFFXGGGFGXDVGDVGFVAVFVGG'
5	>>>a.decipher('DXXFAFGFFXGGGFGXDVGDVGFVAVFVGG')
6	'THEQUICKBROWNFOX'

19.双密码

**** (1) 双密码 ****

双密码(*Bifid Cipher*)结合了波利比奥斯方阵换位密码，并采用分级实现扩散，这里的“双”是指用 2 个密钥进行加密。双密码是由法国 *Felix Delastelle* 发明，除此之外 *Felix Delastelle* 还发明了三分密码(*Trifid Cipher*)，四方密码(*Four-Square Cipher*)。还有一个[两方密码](https://en.wikipedia.org/wiki/Two-square_cipher)(*Two-Square*)与四方密码类似，[共轭矩阵双密码](<http://www.thonky.com/kryptos/cm-bifid-cipher>)(*Conjugated Matrix Bifid Cipher*)也是双密码的变种。

示例密阵：

```

1 2 3 4 5
-----
1 | p h q g m
2 | e a y l n

```

3		o	f	d	x	k
4		r	c	v	s	z
5		w	b	u	t	i/j

明文: `THE QUICK BROWN FOX`

经过密阵转换:

行: `512 15543 54352 333`

列: `421 33525 21115 214`

分组:

`51215 54354 35233 3`

`42133 52521 11521 4`

合并:

5121542133 5435452521 3523311521 34

在经过密阵转换后密文: `WETED TKZNE KYOME X`

****（2）已知密阵加解密****

预览源代码打印关于

1	
2	>>>from pycipher import
3	>>>Bifid('phqgmeaylnofdxkrcvswbuti',5).encipher('THE QUICK BROWN FOX')
4	'WETEDTKZNEKYOMEX'
5	>>>Bifid('phqgmeaylnofdxkrcvswbuti',5).decipher('WETEDTKZNEKYOMEX')
6	'THEQUICKBROWNFOX'

在线加解密[传送门](<http://www.practicalcryptography.com/ciphers/classical-era/bifid/>)

** (3) 未知密阵破解 **

手工分析破解双密码是有一定难度的，每个字母都是同过 3 个数字进行非线性代替转换，而且之后还会对字母顺序进行打乱，这样使双密码比一些替换密码和换位密码更难破解。然而，现在是在计算机时代，这种加密方式没有安全性可言，通过[模拟退

火](http://baike.baidu.com/link?url=mkceUrOW4L7B7UVQxc-dUkXKPJbj9v4YyBh_hrskt5iXk99UdnjW6mZ_YxoJO1PkT1zdjEZD2hd7TCMiSxpOma)算法就能快速找到双密码的密阵。

这里推荐一篇详细的[双密码破解分

析](<http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-bifid-cipher/>)的文章，基于模拟退火算法实现的工具破解示例：

密文：

KWTAZQLAWWZCPONIVBTTBVQUZUGRNHAYIYGIAAYURCUQLDFTYVHTNQE
ENUPAIFCUNQTNQITEFUSHFDWHRIFSVTBISYDHASQSROMUEVPQHHCCRBYT
QBHWYRRHTEPEKHOBFSZUQBTSYRSQUDCSAOVUUGXOAUYWHPGAYHDNKEZ
PFKKWRIEHDWPEIOTBKESYETPBPOGTHQSPUMDOVUEQAUPCPFCQHRPHSOPQ
RSSLPEVWNIQDIOTSQESDHURIEREN

解密：

```
root@kali: ~/break_bifid
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali:~/break_bifid# ./bifidcrack
Running bifidcrack, this could take a few minutes...
best score so far: -1091.442261, on iteration 1
Key: 'GBCM KDHUETLVYWIXOZSPNFARQ'
plaintext: 'CRYPTANALYSIS OF BIFID BY HAND IS ACTUALLY FAIRLY DIFFICULT THE FRACTIONATING NATURE OF THE CIPHER IE EACH LETTER IS SUBSTITUTED BY CHARACTERS THEN THESE CHARACTERS ARE IUM BLED WHICH WILL PULL THEM APART MAKES THE CIPHER MUCH STRONGER THAN SUBSTITUTION CIPHERS OR TRANSPOSITION CIPHERS ON THEIR OWN'
```

得到加密矩阵:

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

明文:

CRYPTANALYSIS OF BIFID BY HAND IS ACTUALLY FAIRLY DIFFICULT
THE FRACTIONATING NATURE OF THE CIPHER IE EACH LETTER IS
SUBSTITUTED BY CHARACTERS THEN THESE CHARACTERS ARE IUM BLED
WHICH WILL PULL THEM APART MAKES THE CIPHER MUCH STRONGER
THAN SUBSTITUTION CIPHERS OR TRANSPOSITION CIPHERS ON THEIR OWN

20.三分密码

三分密码(*Trifid Cipher*)结合换位和替换，三分密码与双密码非常相似，差别之处就是用除了 $3 \times 3 \times 3$ 的密阵代替 5×5 密阵。

示例密阵：

密阵顺序 = EPSDUCVWYM.ZLKXNBTFGORIJHAQ

方阵 1	方阵 2	方阵 3
1 2 3	1 2 3	1 2 3
1 E P S	1 M . Z	1 F G O
2 D U C	2 L K X	2 R I J
3 V W Y	3 N B T	3 H A Q

明文: `THE QUICK BROWN FOX.`

经过密阵转换：

THEQUICKBROWNFOX.
2 3 1 3 1 3 1 2 2 3 3 1 2 3 3 2 2
3 3 1 3 2 2 2 2 3 2 1 3 3 1 1 2 1
3 1 1 3 2 2 3 2 2 1 3 2 1 1 3 3 2

T(233)表示 T 在第一个方阵第三行第三列的位置

分组(分组密钥以 5 为例)：

THEQUICKBROWNFOX.
23131 31223 31233 22
33132 22232 13311 21
31132 23221 32113 32

合并：

23131 33132 31132 31223 22232 23221 31233 13311 32113 22

在经过密阵转换后密文:

23131331323113231223223223221312331331132113222132
N O O N W G B X X L G H H W S K W

想要深入了解三分密码并破解三分密码的小伙伴推荐去看 LANIKI 教授的一篇密码课程章节的[讲

义](<http://www.und.nodak.edu/org/crypto/crypto/lanaki.crypt.class/lessons/lesson17.zip>)。

21.四方密码

** (1) 介绍**

四方密码(Four-Square Cipher)是类似普莱菲尔密码双字母加密密码, 这样使加密效果强于其他替换密码, 因为频率分析变得更加困难了。

四方密码使用 4 个预先设置的 5×5 字母矩阵, 每个矩阵包括 25 个字母, 通常字母 'j' 被融入到 'i' 中(维基百科上说 'q' 被忽略, 不过这不重要, 因为 'q' 和 'j' 都是很少出现的字母), 通常左上和右下矩阵是标准字母排序明文矩阵, 右上和左下矩阵是打乱顺序的密钥矩阵。

示例矩阵:

a b c d e	Z G P T F
f g h i k	O I H M U
l m n o p	W D R C N
q r s t u	Y K E Q A
v w x y z	X V S B L

M F N B D	a
C R H S A	f
X Y O G V	l
I T U E W	q r (s) u
L Q Z K P	v w x y z

T 考考 CS.Net

明文: `THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

整理明文(分组不够时用'X'填充): `TH EQ UI CK BR OW NF OX JU MP SO VE RT HE
LA ZY DO GX`

加密过程: 分别在明文矩阵中找到'TH', 分别找到他们在右上矩阵有左下矩阵的交点字母'ES'
就是密文, 以此类推。

密文: `ESZWQAFHGTDKWHRKUENYQOLMQTUNWMBPTGHQ`

**** (2) 已知密钥矩阵加解密 ****

预览源代码打印关于

1	
2	>>>from pycipher import Foursquare
3	>>>fs = Foursquare('zgptfoihmuwdrcnykeqaxvsbl','mfnbdcrhsaxyogvituewlqzkp')
4	>>>fs.encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')
5	'ESZWQAFHGTDKWHRKUENYQOLMQTUNWMBPTGHQ'
6	>>>fs.decipher('ESZWQAFHGTDKWHRKUENYQOLMQTUNWMBPTGHQ')
7	'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'

在线加解密[传送门](<http://www.practicalcryptography.com/ciphers/classical-era/four-square/>)

**** (3) 未知密钥矩阵破解 ****

推荐一篇关于采用[模拟退火算

法](<http://blog.csdn.net/xianlingmao/article/details/7798647>)的[四方密码分

析]([http://www.practicalcryptography.com/cryptanalysis/stochastic-](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-foursquare-cipher/)

[searching/cryptanalysis-foursquare-cipher/](http://www.practicalcryptography.com/cryptanalysis-stochastic-searching/cryptanalysis-foursquare-cipher/))文章, 如果有足够多的密文那么四方密码可以
以轻易被破解, 如果知道了明文和密文推出密钥是很容易的, 猜测部分明文是一个有效的方法
去破解四方密码, 如果一部分明文已知或者可以被猜测出

那么我们首先要确定尽可能多可利用的密钥, 然后才可以进行更多的推测或者用其他的方法破
译。基于四方密码分析一文实现的[C 代

码]([http://www.practicalcryptography.com/cryptanalysis/stochastic-](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-foursquare-cipher/)

searching/cryptanalysis-foursquare-cipher/)破解示例:

密文(密文最好在 200 个字符以上):

```
HMMKEQESDTMDHLAWFWMNKSOSFOMRFNLWLKHNSQGGEKXEOLLVDXNRSQ
QGARTFKSAVNUDLFDNDHESPZGQ
TWESAGPGSQSQSTPKUSBBQLQHESAGPGSQSQGXLNAVHTPMHMKKNYGSUGD
MTPDGFNKYAVHXLWGKRIEESLZ
ZOFNAVIHRHRKAGHSMYUGEGNSRGAVMVOQPRLNKRXLMYLQPXILESQYBNRHR
KAGKYQXDIHMPGPYOERZOLBEZ
LURFWLWUOLDDPNSQYAGMUQPQWESBEZVEQESDTMDBQLWDIUSHB
```

用法:

预览源代码打印关于

1	<code>gcc -O3 -lm foursquarecrack2.c scoreText_2.c -o fsc</code>
2	<code>./fsc</code>

输出结果:

```
Running foursquarecrack, this could take a few minutes...
best score so far: -1239.505249, on iteration 1
Key:
'KFMLUGWSQEPOZTNRBHDAVXCIY','UGSVKFIZMOYXPQRWTHLNCABED'
plaintext:
'THECIPHERTEXTSQUARESCANBEGENERATEDUSINGAKEYWORDDROPPINGDU
PLICAT
ELETTERSTHENFILLTHEREMAININGSPACESWITHTHEREMAININGL
ETTERSOFTHEA
LPHABETINORDERALTERNATIVELYTHECIPHERTEXTSQUARESCAN
BEGENERATEDCO
MPLETELYRANDOMLYTHEFOURSQUAREALGORITHMALLOWSFORT
WOSEPARATEKEYSO
NEFOREACHOFTHETWOCIPHERTEXTMATRICESX'
```

22. 棋盘密码

棋盘密码 (Checkerboard Cipher) 是使用一个波利比奥斯方阵和两个密钥作为密阵的替换密

码，通常在波利比奥斯方阵中 J 字母往往被包含在 I 字母中。

示例密阵：

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

经过密阵替换：

明文: T H E Q U I C K B R O W N F O X
密文: RK BK RU OC OC BI NK BQ WK RI OQ WI BU NU OQ WU

23. 跨棋盘密码

跨棋盘密码(*Straddle Checkerboard Cipher*)是一种替换密码，当这种密码在结合其他加密方式，加密效果会更好。

棋盘示例(选择 3 和 7 作为变换)：

```
0 1 2 3 4 5 6 7 8 9
f k m c p d y e
3: h b i g q r o s a z
7: l u t j n w v x
```

明文: `THEQUICKBROWNFOX`

经过加密棋盘替换得到密文: `7230934713241313536757403677`

当然我们还可以继续用其他的加密方式在对跨棋盘密码加密出的结果再进行加密：

示例变换密钥: 83729

8372983729837298372983729837

+7230934713241313536757403677

5502817432078501808630122404

在经过棋盘转换后:

5502817432078501808630122404
ppfmyk n if pfkyfyd hkmmcf

最终得到密文: ppfmyk n if pfkyfyd hkmmcf

在线加解密[传送门](<http://www.practicalcryptography.com/ciphers/classical-era/straddle-checkerboard/>)

24. 分组摩尔斯替换密码

分组摩尔斯替换密码(Fractionated Morse Cipher)首先把明文转换为莫尔斯电码, 不过每个字母之间用`x`分开, 每个单词用`xx`分开。然后使用密钥生成一个替换密表, 这个密表包含所有`.`-`x`组合的情况(因为不会出现`xxx`的情况, 所以一共 26 种组合)。

密钥: `MORSECODE`

密表:

MORSECDABFGHIJKNPQTUVWXYZ
.....-----XXXXXXXXX
...---XXX...---XXX...---XX
.-X.-X.-X.-X.-X.-X.-X.-X.-

说明: 密表下半部分是固定的, 密表的安全性以及加密效果主要取决于使用的密钥。

明文: `THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

(类似)摩尔斯电码:

.-X....X.XX-.-X..-X..X-.-X-.-XX-...X.-X---X.-X-..XX..-X---X-..-XX.---X.- --
X.-.-X...XX---X...-X.X.-XX-X....X.XX.-..X.-X---X-.-XX-..X---X---.

说明:明文在转换为(类似)摩尔斯电码后进行每 3 个字符分组, 再进行密表的查表。

密文(经过密表替换):`LMUWC OQVHG ZMTAK EVYSW NOYJQ NLIQB JQCDH
XMDYF TWRGP FWNH`

已知密钥在线加解密[传送门](http://ruffnekk.stormloader.com/fractmorse_tool.html)

2.5.Bazeries 密码

Bazeries 密码(*Bazeries Cipher*)是换位密码和替换密码的组合, 使用两个波利比奥斯方阵, 一个明文字母方阵, 使用一个随机的数字(一般小于 1000000)的生成一个密钥矩阵同时作为第一轮明文划分分组, 比如 2333 这个数字翻译为英文便是 TWO THOUSAND THREE HUNDRED THIRTY THREE,从第一个字母 T 开始选取不重复的字母, 之后再从字母表中按序选取没有出现的字母组成密钥矩阵。

明文:`THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG`

随机数字:`2333`

明文矩阵:

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

示例密钥矩阵:

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

明文分组:

2 3 3 3 2 3 3 3 2 3 3 3
TH EQU ICK BRO WN FOX JUM PSO VE RTH ELA ZYD OG

分组明文反序:

HT UQE KCI ORB WN XOF MUJ OSP EV EHT ALE DYZ GO

使用密钥矩阵替换:

IL XHP QEG KDS YR CKW NXG KBV PU ILD TOP FMZ AK

(比如'H'在明文矩阵对应到密钥矩阵的位置就是'I')

已知密钥在线加解密[传送门](http://ruffnekk.stormloader.com/bazeries_tool.html)

26.Digrafid 密码

Digrafid 密码(Digrafid Cipher)使用两个密钥生成分别生成类似波利比奥斯方阵的 **3x9** 方格的密表。，主要有 **3 分组**和 **4 分组**两类。

第一个方阵密钥: `digrafid`

第二个方阵密钥: `cipher`

密表:

1 2 3 4 5 6 7 8 9
DIGRAFD B C 1 2 3
E H J L M N O P Q 4 5 6
S T U V W X Y Z 7 8 9

cfs1
igt2
pju3
hkv4
elw5

r m x 6
 a n y 7
 b o z 8
 d q # 9

明文: `THE QUICK BROWN FOX`

密表转换(以 4 分组为例):

Th	Eq	Ui	Ck	Br	Ow	Nf	Ox
2	1	3	9	8	7	6	7
7	5	7	2	1	6	5	6
4	9	2	4	6	5	1	6

说明: T 在第一矩阵第 2 列, h 在第二矩阵第 4 行, T 所在的行与 h 所在的列相交的位置数字为 7, 所以 Th 表示为 274。

转换密文:

213 975 724 924 876 716 566 516
 lp #e Dk Ck Zr Dr Mx Ar

27. 格朗普雷密码

格朗普雷密码(*Grandpré Cipher*)是替换密码的一种, 一般使用 8 个 8 字母的单词横向填充 8x8 方阵, 且第一列为一个单词, 并且在方阵中 26 个字母都必须出现一次以上。

示例密阵:

	1	2	3	4	5	6	7	8
1	L	A	D	Y	B	U	G	S
2	A	Z	I	M	U	T	H	S
3	C	A	L	F	S	K	I	N
4	Q	U	A	C	K	I	S	H
5	U	N	J	O	V	I	S	H
6	E	V	U	L	S	I	S	H
7	R	O	W	D	Y	I	S	H
8	S	E	X	T	U	P	I	Y

(零零CS.Net

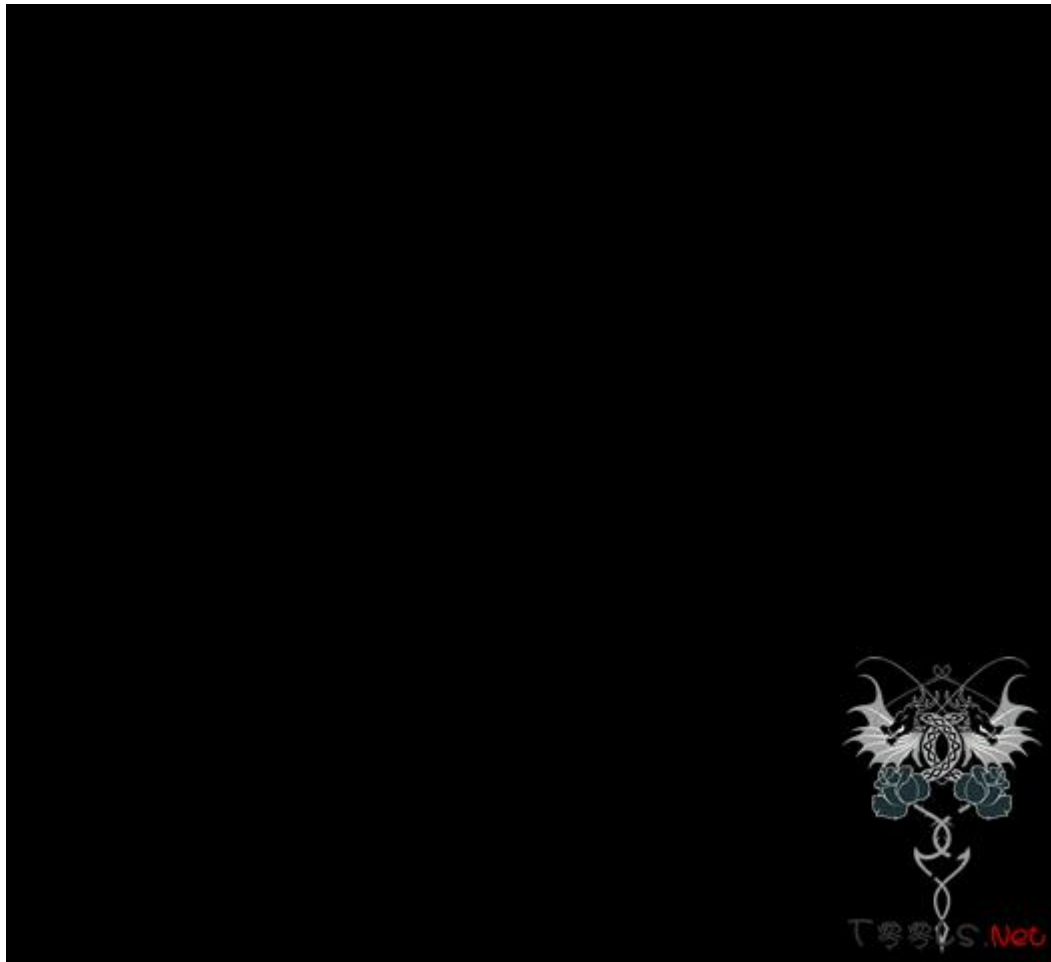
明文: T H E Q U I C K B R O W N F O

密文: 84 27 82 41 51 66 31 36 15 71 67 73 52 34 67

说明: 明文中的字母在密阵位置可能不止一个, 所以加密结果可能有多种, 但是不影响解密。
密阵还有 6x6, 7x7, 9x9, 10x10 几种。显然密阵越大每个字母被替换的情况就可能越多, 那么加密效果就更好。

28. 比尔密码

比尔密码(Beale ciphers)有三份密码, 当然这里说的是已被破解第二份, 是一种类似书密码的替换密码。



以第二密码为例，每一个数字代表美国《独立宣言》的文本中的第几个词的首字母，如 1 代表第 1 个词的首字母“w”，2 代表第 2 个词首字母“i”。解密后的文字如下：

I have deposited in the county of Bedford...

比尔密码还有一段有趣的故事，感兴趣可以看一下比尔密码的[详细介绍](<https://zh.wikipedia.org/wiki/%E6%AF%94%E5%B0%94%E5%AF%86%E7%AO%81>)。

29. 键盘密码

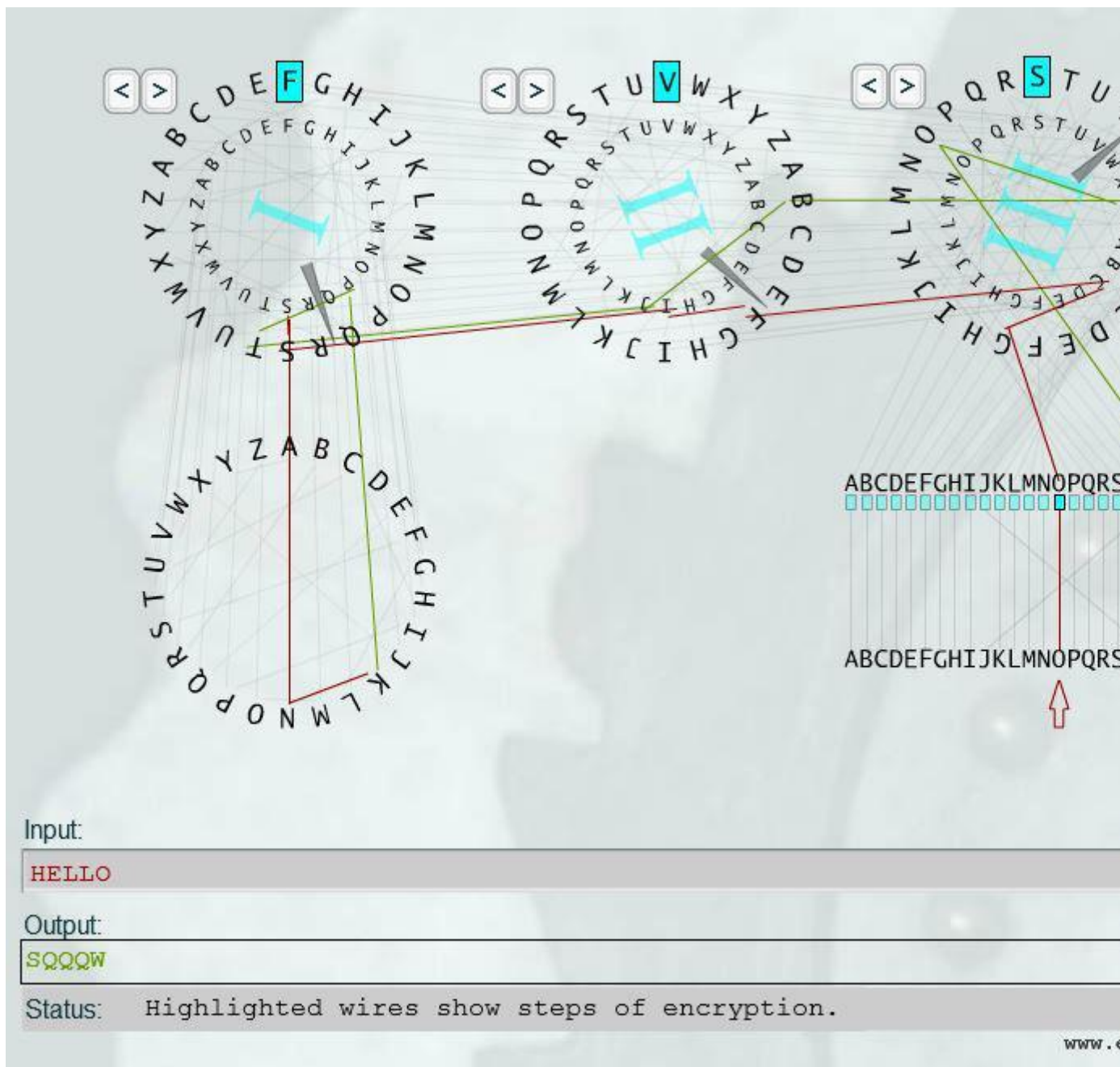
一般用到的键盘密码就是手机键盘和电脑键盘两种，2014 Octf 比赛里 Crypto 类型中 Classic 一题就是电脑键盘密码，详细可以[参考](<http://www.programlife.net/Oops-ctf-writeup.html>)，另外给出另外一些[参考](<http://www.secbox.cn/hacker/ctf/8078.html>)情况。

其他有趣的机械密码

1. 恩尼格玛密码

恩尼格玛密码机（德语：*Enigma*，又译哑谜机，或“谜”式密码机）是一种用于加密与解密文件的密码机。确切地说，恩尼格玛是对二战时期纳粹德国使用的一系列相似的转子机械加解密机器的统称，它包括了许多不同的型号，为密码学对称加密算法的流加密。详细工作原理参考[维基百

科](<https://zh.wikipedia.org/wiki/%E6%81%A9%E5%B0%BC%E6%AO%BC%E7%8E%9B%E5%AF%86%E7%AO%81%E6%9C%BA>)。



在线模拟[传送门](<http://enigmaco.de/enigma/enigma.html>)

感兴趣可以观看[播单:计算机历史文化

课](<http://list.youku.com/albumlist/show?id=23400097&ascending=1&page=1>)

代码混淆加密

1.asp 混淆加密

[asp 混淆加密](<http://www.zhaoyuanma.com/aspfix.html>)

2.php 混淆加密

[php 混淆加密](http://www.zhaoyuanma.com/phpjmvip.html)

3.css/js 混淆加密

[css/js 混淆加密](http://tool.css-js.com/)

4.VBScript.Encode 混淆加密

[VBScript.Encode 混淆加密](http://www.zhaoyuanma.com/aspfix.html)

5.pencode

pencode-Perl 把 Perl 代码转换成只有英文字母的字符串。

ppencode - JavaScript demo

Input text

```
print "Hello Perl"
```

Perl program

```
#!/usr/bin/perl -w
length q closedir vec and print chr ord q open no and print chr ord q
qr eq and print chr ord q tie lt and print chr ord qw q no q and print
chr ord q gt log and print chr ord q q q and print chr length q q ge
getc getpriority printf split q and print chr ord uc q chr uc and print
chr ord q ge log and print chr ord qw q le q and print chr ord qw q le
q and print chr ord q pop and print chr ord q q eq and print chr oct
ord q qx eq and print chr ord q ne sin and print chr ord q qr q and
print chr ord q else and print chr length q q ge getc getpriority
printf split q
```

pencode[传送门](http://namazu.org/~takesako/ppencode/demo.html)

6.rrencode

[illegible]

7.jjencode/aaencode

`jjencode` 将 JS 代码转换成只有符号的字符串，类似于 `rrencode`，介绍的 [PPT](<http://utf-8.jp/public/20090710/jjencode.pps>)，`aaencode` 可以将 JS 代码转换成常用的网络表情，也就是我们说的颜文字 js 加密。

aaencode demo



aaencode - Encode any JavaScript program to Japanese style emoticons (^_^)

Enter JavaScript source:

```
alert("Hello, JavaScript")
```

```
ω/= /`m`)/~11 // *`∇`*/ ['_']; o=(-) =_3; c=(Θ) =(-)-(-); (Д) =(Θ)=
(o^_o); (Д)={Θ: '_', ω/: ((ω/=3) +'_') [Θ], -/: (ω/+ '_')[o^_o-(Θ)]
+'_'[-] }; (Д) [Θ] =((ω/=3) +'_') [c^_o]; (Д) ['c'] = ((Д)+'_') [(-)+(-)
Д) ['o'] = ((Д)+'_') [Θ]; (o)=(Д) ['c']+(Д) ['o']+(ω/+ '_')[Θ]+ ((ω/=3)
((Д) +'_') [(-)+(-)]+ ((-==3) +'_') [Θ]+((-==3) +'_') [(-)-(Θ)]+(Д) [
+'_' [(-)+(-)]+ (Д) ['o']+( (-==3) +'_') [Θ]; (Д) ['_'] =(o^_o) [o] [o];
+'_' [Θ]+ (Д) .Д /+((Д)+'_') [(-)+(-)]+((-==3) +'_') [o^_o-Θ]+((-==
+ (ω/+ '_') [Θ]; (-)+=(Θ); (Д) [ε]='\\'; (Д) .Θ /=(Д+ '-') [o^_o-(Θ)]; (
+'_' [c^_o]; (Д) [o]='\"'; (Д) ['_'] ( (Д) ['_'] (ε+(Д) [o]+ (Д) [ε]+(Θ)
+ (Д) [ε]+(Θ)+ ((-) + (Θ))+ (-)+ (Д) [ε]+(Θ)+ ((-) + (Θ))+ (Д)
((o^_o) +(o^_o))+ ((o^_o) -(Θ))+ (Д) [ε]+(Θ)+ ((o^_o) +(o^_o))+ (-)+ (Д
+ (Θ))+ (c^_o)+ (Д) [ε]+(-)+ ((o^_o) -(Θ))+ (Д) [ε]+(Θ)+ (Θ)+ (c^_o)
Θ)+ ((-) + ((-) + (Θ))+ (Д) [ε]+(Θ)+ ((-) + (Θ))+ (-)+ (Д) [ε]+(Θ)+
+ ((-) + (Д) [ε]+(Θ)+ ((-) + (Θ))+ ((-) + (o^_o))+ (Д) [ε]+((-) + (Θ))
ε]+((-) + (c^_o)+ (Д) [ε]+(Θ)+ (Θ)+ ((o^_o) -(Θ))+ (Д) [ε]+(Θ)+ (-)+
ε]+(Θ)+ ((o^_o) +(o^_o))+ ((o^_o) -(Θ))+ (Д) [ε]+(Θ)+ (-)+ (Θ)+ (π) [ε]+(Θ)+ (-)+ (Θ)+ (π)
```

aaencode[传送门](<http://utf-8.jp/public/aaencode.html>)

jjencode/aaencode 的解密直接在浏览器的控制台里输入密文即可执行解密，想要详细了解 jjencode 是如何进行请[参考](<http://pferrie2.tripod.com/papers/jjencode.pdf>)，你也可以在 github 上[下载](<https://github.com/jacobsoo/Decoder-JJEncode>)实现 jjdecoder 的源码进行分析。

JSFuck 可以让你只用 6 个字符 `[] () ! + `` 来编写 JavaScript 程序。



[[!] JSFuck

JSFuck is an esoteric and educational programming style based on the atomic parts of JavaScript. It uses only six different characters to write and execute code.

It does not depend on a browser, so you can even run it on Node.js.

Use the form below to convert your own script. Uncheck "eval source" to get back a plain string.

☒ Encode ☐ Eval Source

```
(![]+[]) [+!+[]]+(![]+[]) [!+[]+!+[]]+(!![]+[]) [!+[]+!+[]+!+[]]+(!![]+
[]) [+!+[]]+(!![]+[]) [+[]]+(![]+[]) [(![]+[]) [+[]]+(![]+[]) [![])] [+!+[]
+[]+[]]+(![]+[]) [!+[]+!+[]]+(!![]+[]) [+[]]+(!![]+[]) [!+[]+!+[]+!+[]]
+(!![]+[]) [+!+[])] [!+[]+!+[]+[]+[]]+[+!+[]]+(!![]+[]) [(![]+[]) [+[]]+
(![]+[]) [![])] [+!+[]+[]+[]]+(![]+[]) [!+[]+!+[]]+(!![]+[]) [+[]]+(!![]
+[]) [!+[]+!+[]+!+[]]+(!![]+[]) [+!+[])] [!+[]+!+[]+[]+[]]
```

396 chars

Run This

Links

JSfuck[传送门](<http://www.jsfuck.com/>)

9.jother

jother 是一种运用于 *javascript* 语言中利用少量字符构造精简的匿名函数方法对于字符串进行的编码方式。其中 8 个少量字符包括：! + () [] { } `。只用这些字符就能完成对任意字符串的编码。

[do9gy](http://drops.wooyun.org/author/do9gy)的[jother 编码之谜](http://drops.wooyun.org/web/4410)



10.brainfuck

Brainfuck 是一种极小化的计算机语言，按照“Turing complete（完整图灵机）”思想设计的语言，它的主要设计思路是：用最小的概念实现一种“简单”的语言，BrainF**k 语言只有八种符号，所有的操作都由这八种符号（< > + - . , [] \）的组合来完成。

明文: *hello!*

```
+++++ +++++ [->+ +++++ +++++] >++++ .---. +++++ ++..+ ++.<+ +++++ +++++  
[->+ +++++ +++++< ]>+++ +++++. <++++ +++++[- >----- ---<] >--.< +++++ ++[->  
----- ---<]> ----- -----.<
```

brainfuck[传送门](<http://www.splitbrain.org/services/ook>)

其他稀奇古怪的编程语言请[参考](<http://news.mydrivers.com/1/190/190926.htm>)

相关工具

[JPK 神器](http://www.wechall.net/applet/JPK_406.jar)

[密码破解脚本](https://github.com/jameslyons/python_cryptanalysis)

[shellcode 转换脚

本](<https://dl.packetstormsecurity.net/shellcode/shellcodeencdec.py.txt>)

参考网站

[Wikipedia Classical Cipher](https://en.wikipedia.org/wiki/Classical_cipher)

[Cryptogram Cipher Types](http://cryptogram.org/cipher_types.html)

[Practical Cryptography](<http://www.practicalcryptography.com/>)

[Rumkin Cipher Tools](<http://rumkin.com/tools/cipher/>)

[encode-decode](<https://encode-decode.appspot.com/>)

[4ido1On Blog](<https://www.hackfun.org/>)

[Anagram](<http://wordsmith.org/anagram/>)

[Thonky](<http://www.thonky.com/kryptos/>)

彩蛋是什么



