CTF-密码学CRYPTO

讲师：时匣

目 录

CONTENTS

介绍

◆ 密码学（Cryptography）一般可分为古典密码学和现代密码学。
一般来说，密码设计者的根本目标是保障信息及信息系统的：

- 机密性（Confidentiality）
- 完整性（Integrity）
- 可用性（Availability）
- 认证性（Authentication）
- 不可否认性（Non-repudiation）

其中，前三者被称为信息安全的 CIA 三要素 。

对密码系统的攻击类型

而对于密码破解者来说，一般是要想办法识别出密码算法，然后进行暴力破解，或者利用密码体制的漏洞进行破解。当然，也有可能通过构造虚假的哈希值或者数字签名来绕过相应的检测。

➢ 一般来说，我们都会假设攻击者已知待破解的密码体制，而攻击类型通常分为以下四种：

攻击类型	攻击者掌握的内容
唯密文攻击	加密算法 截获的部分密文
已知铭文攻击	加密算法 截获的部分密文 一个或多个明文密文对
选择铭文攻击	加密算法 截获部分密文 自己选择的明文消息，以及由密钥刹那生的相应密文
选择密文文攻击	加密算法 截获的部分密文 自己选择的密文消息，以及相应的被解密的明文

目 录

CONTENTS

ASCII编码

ASCII（Ame 准代码）是基于拉丁字母的 通用的单字节编码系统，并

ASCII表

（ American Standard Code for Information Interchange 美国标准信息交换代码 ）

高四位 / 低四位		ASCII控制字符											ASCII打印字符																
		0000						0001					0010		0011		0100		0101		0100		0111							
		0						1					2		3		4		5		6		7							
		十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl
0000	0	0		^@	NUL	\0	空字符	16	▶	^P	DLE		数据链路转义	32		48	0	64	@	80	P	96	`	112	p					
0001	1	1	☺	^A	SOH		标题开始	17	◀	^Q	DC1		设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q					
0010	2	2	☻	^B	STX		正文开始	18	↕	^R	DC2		设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r					
0011	3	3	♥	^C	ETX		正文结束	19	‼	^S	DC3		设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s					
0100	4	4	♦	^D	EOT		传输结束	20	¶	^T	DC4		设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t					
0101	5	5	♣	^E	ENQ		查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u					
0110	6	6	♠	^F	ACK		肯定应答	22	▬	^V	SYN		同步空闲	38	&	54	6	70	F	86	V	102	f	118	v					
0111	7	7	•	^G	BEL	\a	响铃	23	↨	^W	ETB		传输块结束	39	'	55	7	71	G	87	W	103	g	119	w					
1000	8	8	◘	^H	BS	\b	退格	24	↑	^X	CAN		取消	40	(	56	8	72	H	88	X	104	h	120	x					
1001	9	9	○	^I	HT	\t	横向制表	25	↓	^Y	EM		介质结束	41	)	57	9	73	I	89	Y	105	i	121	y					
1010	A	10	◙	^J	LF	\n	换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z					
1011	B	11	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{					
1100	C	12	♀	^L	FF	\f	换页	28	∟	^\	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|					
1101	D	13	♪	^M	CR	\r	回车	29	↔	^]	GS		组分隔符	45	-	61	=	77	M	93	]	109	m	125	}					
1110	E	14	♫	^N	SO		移出	30	▲	^^	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~					
1111	F	15	☼	^O	SI		移入	31	▼	^-	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	⌂				^Backspace 代码：DEL	

GB2312编码

GB2312是由中国国家标准总[illegible]国家标准，GB2312编码适用于汉字处理、汉字通信等系统[illegible]也采用此编码。中国大陆几乎所有的中文系统和国际[illegible]

GB2312简体中文编码表

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
A1A0		、	。	·	ˉ	ˇ	¨	〃	々	—	～	‖	…	‘	’	
A1B0	“	”	（	）	〈	〉	《	》	「	」	『	』	〖	〗	【	】
A1C0	±	×	÷	：	∧	∨	∑	∏	∪	∩	∈	∷	√	⊥	//	∠
A1D0	⌒	⊙	∫	∮	≡	≌	≈	∽	∝	≠	≮	≯	≤	≥	∞	∵
A1E0	∴	♂	♀	°	′	″	℃	$	¤	￠	￡	‰	§	№	☆	★
A1F0	○	●	◎	◇	◆	□	■	△	▲	※	→	←	↑	↓	=	

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
A2A0		i	ii	iii	iv	v	vi	vii	viii	ix	x					
A2B0		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
A2C0	16.	17.	18.	19.	20.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
A2D0	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	①	②	③	④	⑤	⑥	⑦
A2E0	⑧	⑨	⑩	∈		㈠	㈡	㈢	㈣	㈤	㈥	㈦	㈧	㈨	㈩	
A2F0		I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII			

code	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
A3A0		!	"	#	¥	%	&	'	(	)	*	+	,	-	.	/
A3B0	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
A3C0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A3D0	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
A3E0	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
A3F0	p	q	r	s	t	u	v	w	x	y	z					

Unicode编码

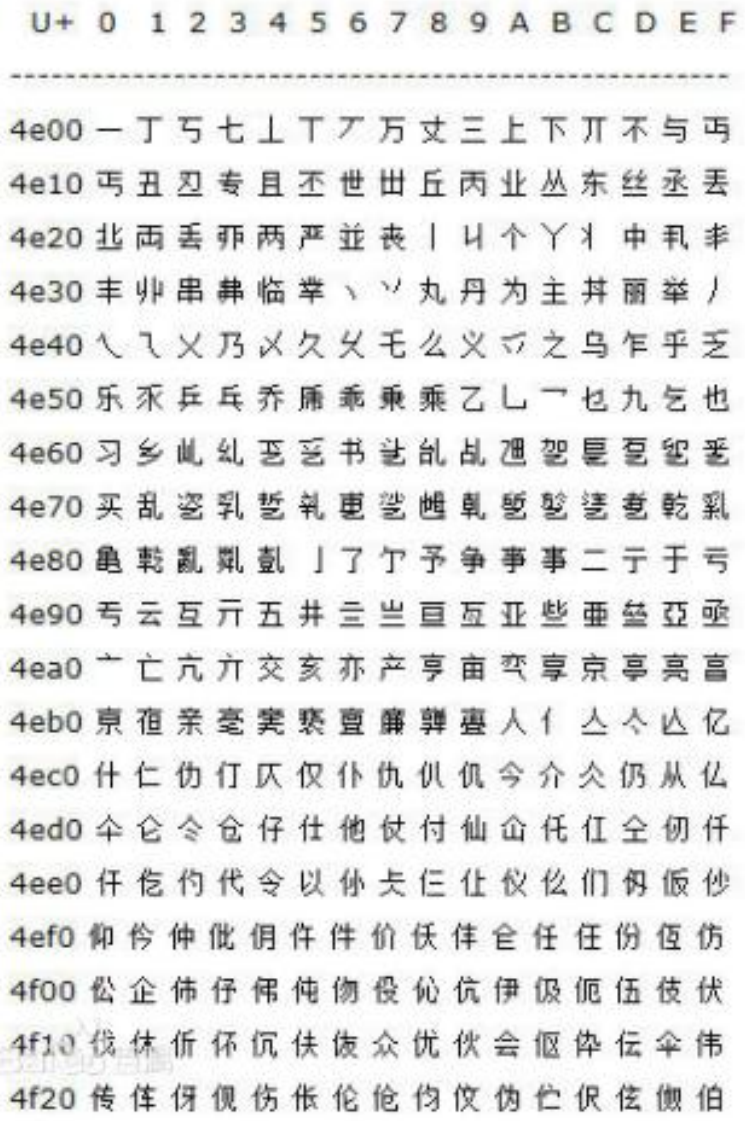
Unicode是CTF中常出现的简单编码类题目，编码形式：

\u0043\u0054\u0046\u6559\u7a0b

例如：原文本：You had me at hello

编码后

\u0059\u006f\u0075\u0020\u0068\u0061\u0064\u0020\u006d\u0065\u0020\u0061\u0074\u0020\u0068\u0065\u006c\u006c\



UTF-8编码

UTF-8就是在互联网上使用最广的一种Unicode的实现方式。其他实现方式还包括UTF-16（字符用两个字节或四个字节表示）和UTF-32（字符用四个字节表示），不过在互联网上基本不用。重复一遍，这里的关系是，UTF-8是Unicode的实现方式之一。

Unicode/UCS-4	bit数	UTF-8	byte数	备注
0000 ~ 007F	0~7	0XXX XXXX	1	
0080 ~ 07FF	8~11	110X XXXX 10XX XXXX	2	
0800 ~ FFFF	12~16	1110XXXX 10XX XXXX 10XX XXXX	3	基本定义范围：0~FFFF
1 0000 ~ 1F FFFF	17~21	1111 0XXX 10XX XXXX 10XX XXXX 10XX XXXX	4	Unicode6.1定义范围：0~10 FFFF
20 0000 ~ 3FF FFFF	22~26	1111 10XX 10XX XXXX 10XX XXXX 10XX XXXX 10XX XXXX	5	说明：此非unicode编码范围，属于UCS-4 编码 早期的规范UTF-8可以到达6字节序列，可以覆盖到31位元（通用字符集原来的极限）。尽管如此，2003年11月UTF-8被 RFC 3629 重新规范，只能使用原来Unicode定义的区域， U+0000到U+10FFFF。根据规范，这些字节值将无法出现在合法 UTF-8序列中
400 0000 ~ 7FFF FFFF	27~31	1111 110X 10XX XXXX 10XX XXXX 10XX XXXX 10XX XXXX 10XX XXXX	6	

Base家族16/32/64

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。

索引	对应字符	索引	对应字符	索引	对应字符	索引	对应字符
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		
15	P	32	g	49	x		
16	Q	33	h	50	y		

URL编码

url编码又叫百分号编码，址（常说网址）规定了常用地数字，字母可以直接位直接用（/，:@等），剩下的其它所有字符必须通过%

backspace %08	I %49	v %76	ó %D3
tab %09	J %4A	w %77	Ô %D4
linefeed %0A	K %4B	x %78	Õ %D5
creturn %0D	L %4C	y %79	Ö %D6
space %20	M %4D	z %7A	Ø %D8
! %21	N %4E	{ %7B	ù %D9
" %22	O %4F	| %7C	ú %DA
# %23	P %50	} %7D	Û %DB
$ %24	Q %51	~ %7E	ü %DC
% %25	R %52	¢ %A2	Y %DD
& %26	S %53	£ %A3	T %DE
' %27	T %54	¥ %A5	ß %DF
(%28	U %55	| %A6	à %E0
) %29	V %56	§ %A7	á %E1
* %2A	W %57	« %AB	a %E2
+ %2B	X %58	¬ %AC	ã %E3
, %2C	Y %59	¯ %AD	ä %E4
- %2D	Z %5A	o %B0	å %E5
. %2E	[%5B	± %B1	æ %E6
/ %2F	\ %5C	a %B2	ç %E7
0 %30	] %5D	, %B4	è %E8
1 %31	^ %5E	µ %B5	é %E9
2 %32	_ %5F	» %BB	ê %EA
3 %33	` %60	¼ %BC	ë %EB
4 %34	a %61	½ %BD	ì %EC
5 %35	b %62	¿ %BF	í %ED
6 %36	c %63	à %C0	î %EE

Escape/Unescape编码

Escape/Unescape编码又叫%u编码，跟URL编码类似，只不过Escape/Unescape编码是采用UTF-16BE模式。从编码字符串出现有"u"，可以判断它是unicode编码，Escape编码实际上就是采用了UTF-16BE模式的unicode编码。这样一来问题非常简单了。 Escape编码就是字符对应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”，反之也一样。

UUencode编码

UUencode是一种二进制到文字的编码，是将二进制文件转换为文本文件的过程，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，如果最后剩下的字节少于三个字节，不够的部份用零补齐。

- UUencode编码原理：
- 明文：AnY
- 第一步：先转换为ASCII码为：65 110 89
- 第二步：转为二进制为：01000001 01101110 01011001
- 第三步：6个一组为：010000 010110 111001 011001
- 第四步：转为十进制：16 22 57 25
- 第五步：加32为：48 54 89 57
- 第六步：AnY的长度为3，加上32为35，25对应的ASCII码为#
- 第七步：则“AnY”的Uuencode编码为“#06Y9”。

Uuencode编码与Base64编码很像，区别是在第五步，Uuencode直接加上32，同时再标明明文长度，而Base64是对应Base64编码表得到。

原文：Day is a lovely day

编码：31&%Y(&ES(&$@;&]V96QY(&1A>0``

Jother编码

当我们看到一大堆[]()+!这些字符组成的编码，就是Jother编码。

Jother编码有两大特点

1、简单，不需要太多

2、是在对javascript

```
[+[]]+({}+[])[!![]+!![]+!![]+!![]+!![]]+(!![]+[])[+[]]+({}+[])[+!![]]+(!![]+[])[+!![]]((!![]+[])[+!![]]+(!![]+[])[!![]
+!![]+!![]]+(!![]+[])[+[]]+({}[[]]+[])[+[]]+(!![]+[])[+!![]]+({}[[]]+[])[+!![]]+({}+[])[!![]+!![]+!![]+!![]+!![]+!![]+!
![]]+(![]+[])[!![]+!![]]+({}+[])[+!![]]+({}+[])[!![]+!![]+!![]+!![]+!![]]+(![]+[])[+!![]]+(!![]+[])[+[]]+({}[[]]+[])[!!
+!![]+!![]+!![]+!![]]+({}+[])[+!![]]+({}[[]]+[])[+!![]])()[!![]+!![]+!![]]+(!![]+[])[!![]+!![]+!![]])()([][(![]+[])[!![]
+!![]+!![]]+({}+[])[+!![]]+(!![]+[])[+!![]]+(!![]+[])[+[]]][({}+[])[!![]+!![]+!![]+!![]+!![]]+({}+[])[+!![]]+({}[[]]+[]
)[+!![]]+(!![]+[])[!![]+!![]+!![]]+(!![]+[])[+[]]+(!![]+[])[+!![]]+({}[[]]+[])[+[]]+({}+[])[!![]+!![]+!![]+!![]+!![]]+(!!
[]+[])[+[]]+({}+[])[+!![]]+(!![]+[])[+!![]]]((!![]+[])[+!![]]+(!![]+[])[!![]+!![]+!![]]+(!![]+[])[+[]]+({}[[]]+[])[+[]]
+(!![]+[])[+!![]]+({}[[]]+[])[+!![]]+({}+[])[!![]+!![]+!![]+!![]+!![]+!![]+!![]]+(!![]+[])[!![]+!![]+!![]]+(![]+[])[!![]
+!![]+!![]]+({}+[])[!![]+!![]+!![]+!![]+!![]]+(![]+[])[+!![]]+([]+[][(![]+[])[!![]+!![]+!![]]+({}+[])[+!![]]+(!![]+[])[
+!![]]+(!![]+[])[+[]]][({}+[])[!![]+!![]+!![]+!![]+!![]]+({}+[])[+!![]]+({}[[]]+[])[+!![]]+(!![]+[])[+!![]]+(!![]+[])[!!
+[]][+[]]+(!![]+[])[+!![]]+({}[[]]+[])[+[]]+({}+[])[!![]+!![]+!![]+!![]+!![]+!![]+!![]+!![]]+(!![]+[])[+!![]]+(!![]+[])
[+!![]]]((!![]+[])[+!![]]+(!![]+[])[!![]+!![]+!![]]+(!![]+[])[+[]]+({}[[]]+[])[+[]]+(!![]+[])[+!![]]+({}[[]]+[])[+!![]]+
({}+[])[!![]+!![]+!![]+!![]+!![]+!![]+!![]]+(!![]+[])[+!![]]+({}+[])[+!![]]+({}+[])[!![]+!![]+!![]+!![]+!![]]+(![]+[]
)[+!![]]+(!![]+[])[+[]]+({}[[]]+[])[!![]+!![]+!![]+!![]+!![]]+({}+[])[+!![]]+({}[[]]+[])[+!![]])()[!![]+!![]+!![]]+(!![]
```

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K字母被整合到C中。

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

摩尔斯电码

摩尔斯电码（又译为摩斯密码，Morse code）是一种时通时断的信号代码，通过不同的排列顺序来表达不同的英文字母、数字和标点符号。它发明于1837年，发明者有争议，是美国人塞缪尔·莫尔斯或者艾尔菲德·维尔。 摩尔斯电码是一种早期的数字化通信形式，但是它不同于现代只使用零和一两种状态的二进制代码，它的代码包括五种： 点、划、点和划之间的停顿、每个字符间短的停顿（在点和划之间）、每个词之间中等的停顿以及句子之间长的停顿。

字符	电码符号	字符	电码符号	字符	电码符号	字符	电码符号
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	−−··				

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	1	0	1	0	0
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

目 录

CONTENTS

古典密码简介

在古典密码学中，我们主要介绍单表替代密码，多表替代密码，以及一些其它比较有意思的密码。

值得一提的是，在古典密码学中，设计者主要考虑消息的保密性，使得只有相关密钥的人才可以解密密文获得消息的内容，对于消息的完整性和不可否认性则并没有进行太多的考虑。

古典密码学：单表代换加密、多表代换加密

单表代换加密

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

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埃特巴什码

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/

多表代换加密

对于多表替换加密来说，加密后的字母几乎不再保持原来的频率，所以我们一般只能通过寻找算法实现对应的弱点进行破解。

Playfair普来费尔

原理¶

Playfair 密码（Playfair cipher or Playfair square）是一种替换密码，1854 年由英国人查尔斯·惠斯通（Charles Wheatstone）发明，基本算法如下：

1、选取一串英文字母，除去重复出现的字母，将剩下的字母逐个逐个加入 5 × 5 的矩阵内，剩下的空间由未加入的英文字母依 a-z 的顺序加入。注意，将 q 去除，或将 i 和 j 视作同一字。

2、将要加密的明文分成两个一组。若组内的字母相同，将 X（或 Q）加到该组的第一个字母后，重新分组。若剩下一个字，也加入 X 。

3、在每组中，找出两个字母在矩阵中的地方。

- 若两个字母不同行也不同列，在矩阵中找出另外两个字母（第一个字母对应行优先），使这四个字母成为一个长方形的四个角。
- 若两个字母同行，取这两个字母右方的字母（若字母在最右方则取最左方的字母）。
- 若两个字母同列，取这两个字母下方的字母（若字母在最下方则取最上方的字母）。

Playfair普来费尔

- 新找到的两个字母就是原本的两个字母加密的结果。

  以 playfair example 为密匙，得

  P L A Y F

  I R E X M

  B C D G H

  K N O Q S

  T U V W Z

  要加密的讯息为 Hide the gold in the tree stump

  HI DE TH EG OL DI NT HE TR EX ES TU MP

  就会得到

  BM OD ZB XD NA BE KU DM UI XM MO UV IF

➢ 工具¶

  CAP4

Polybius棋盘密码

原理¶

Polybius密码又称为棋盘密码，其一般是将给定的明文加密为两两组合的数字，其常用密码表，举个例子，明文 HELLO，加密后就是 23 15 31 31 34。

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

Polybius棋盘密码

另一种密码表

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

注意，这里字母的顺序被打乱了。

A D F G X 的由来：

1918 年，第一次世界大战将要结束时，法军截获了一份德军电报，电文中的所有单词都由 A、D、F、G、X 五个字母拼成，因此被称为 ADFGX 密码。ADFGX 密码是 1918 年 3 月由德军上校 Fritz Nebel 发明的，是结合了 Polybius 密码和置换密码的双重加密方案。

举个例子，HELLO，使用这个表格加密，就是 DD XF AG AG DF。

工具¶

- CrypTool

Vigenere 维吉尼亚密码

原理¶

维吉尼亚密码（Vigenere）是使用一系列凯撒密码组成密码字母表的加密算法，属于多表密码的一种简单形式。

	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

Vigenere 维吉尼亚密码

下面给出一个例子

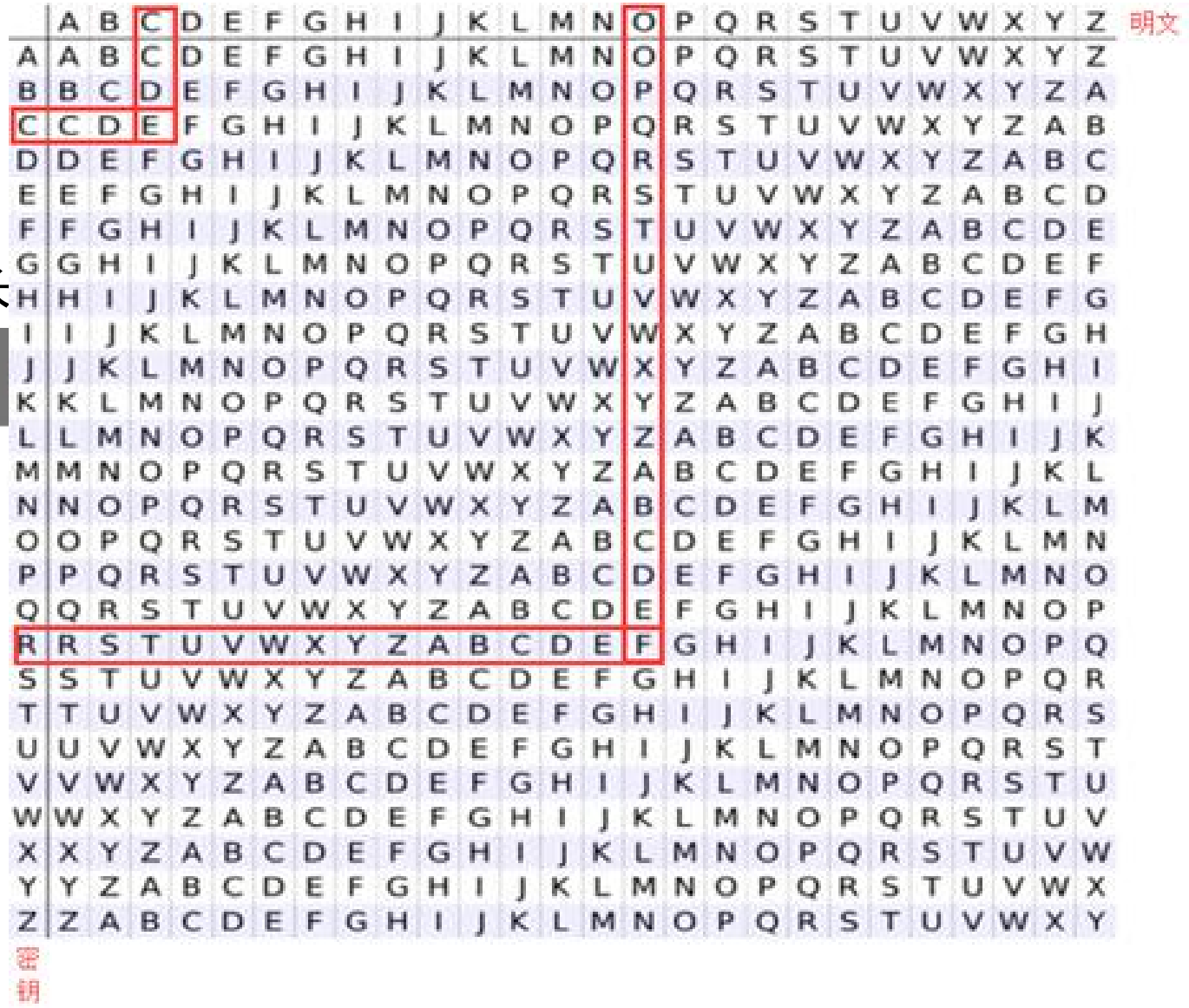
明文：come greatwall

密钥：crypto

首先，对密钥进行填充使其长

明文	c	o	m	e	g
密钥	c	r	y	p	t

其次，查表得密文



明文：come greatwall
密钥：crypto
密文：efkt zferrltzn

Nihilist关键字密码

原理¶

- Nihilist密码又称关键字密码：明文 + 关键字 = 密文。以关键字 helloworld 为例。
- 首先利用密钥构造棋盘矩阵（类似 Polybius 密码） - 新建一个 5 × 5 矩阵 - 将字符不重复地依次填入矩阵 - 剩下部分按字母顺序填入 - 字母 i 和 j 等价

	1	2	3	4	5
1	h	e	l	o	w
2	r	d	a	b	c
3	f	g	i/j	k	m
4	n	p	q	s	t
5	u	v	x	y	z

- 对于加密过程参照矩阵 M 进行加密：

  a -> M[2,3] -> 23

  t -> M[4,5] -> 45

- 对于解密过程

  参照矩阵 M 进行解密：

  23 -> M[2,3] -> a

  45 -> M[4,5] -> t

可以看出，密文的特征有如下几点

- 纯数字
- 只包含 1 到 5
- 密文长度偶数。

AutokeyCipher自动密钥密码¶

原理¶

自动密钥密码（Autokey Cipher）也是多表替换密码，与维吉尼亚密码密码类似，但使用不同的方法生成密钥。通常来说它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，从相应的表格可得：

密文

VBP JOZGD IVEQV HYY AIICX CSNL FWW ZVDP WVK

目 录

CONTENTS

栅栏密码

栏栅栏密码原理：

明文：Day is a lovely day

去掉空格后变为：Dayisalovelyday

分组（2个一组）：Da yi sa lo ve ly da y

第一组（取第一个字符）：Dyslvldy

第二组（取第二个字符）：aiaoeya

连在一起即为密文：Dyslvldyaiaoeya

7栏栅栏密码原理就是分为7个一组，然后组合。

破解栅栏密码，首先需要先明确是几栏栅栏密码加密，然后进行破解。

曲路密码

曲路密码（Curve Cipher）是一种换位密码，需要双方事先约定密钥（也就是曲路路径）。

曲路密码原理：

明文：Your happiness often in the eyes of others

Y	o		Y	o			u	r			h
a	p		a	p			p	i			n
e	s		e	s			s	o			f
t	e		t	e			n	i			n
t	h		t	h			e	e			y
e	s		e	s			o	f			o
t	h		t	h			e	r			s

密文：soynfnh rioiefr eoenspu opsehsh tetteaY

破解曲路密码，首先需要先明确表填充的几行几列曲路密码加密，然后进行破解。

埃特巴什码

埃特巴什码（Atbash Cipher）是一种以字母倒序排列作为密钥的替换加密。

埃特巴什码原理：

加密的对应关系：

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ZYXWVUTSRQPONMLKJIHGFEDCBA

明文：Your happiness often in the eyes of others

密文：Blfi szkkrmvhh lugvm rm gsv vbvh lu lgsvih

ROT5/13/18/47

ROT5/13/18/47加密是一种替换加密，它具有可逆性。ROT5/13/18/47是rotate by 5/13/18/47places的简写，意思是旋转几个位置。

ROT5：只对数字进行编码，用当前数字往前数的第五个数字替换当前数字。比如当前数字为0，替换后就变成5；当前数字为1，替换后就变成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十三个字母替换当前字母，比如当前字母为A，替换后变成N；当前字母为B，替换后就变成O；以此类推顺序循环。

ROT18：它的编码规则是将ROT5和ROT13的编码规则组合在一起，比如当前明文是A0，替换后就变成N5；当前明文是B1，替换后就变成O6；以此类推顺序循环。

ROT47：对数字、字母、常用符号进行编码，按照它们的ASCII码值进行位置替换，用当前字符ASCII值往前数的第47位对应字符替换当前字符，比如当前为小写字母z，

简单换位密码

简单换位密码（Simple Substitution Cipher），它的加密方法是每个明文字母都被唯一且不同的字母替换。它的密码字母表的字母不是简单的移位，而是完全是混乱的。

明文：Your happiness often in the eyes of others

字母对应关系：

明文字母：abcdefghijklmnopqrstuvwxyz

密文字母：phqgiumeaylnofdxjkrcvstzwb

密文：

破解简单换位密码，当密文数据足够多时可以通过字频分析方法破解或其他方法破解。

希尔密码

希尔密码（Hill Cipher）是基于线性代数多重代换密码。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*n的矩阵相乘，再将得出的结果MOD26。

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

希尔密码

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

加密矩阵：

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

计算过程：

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

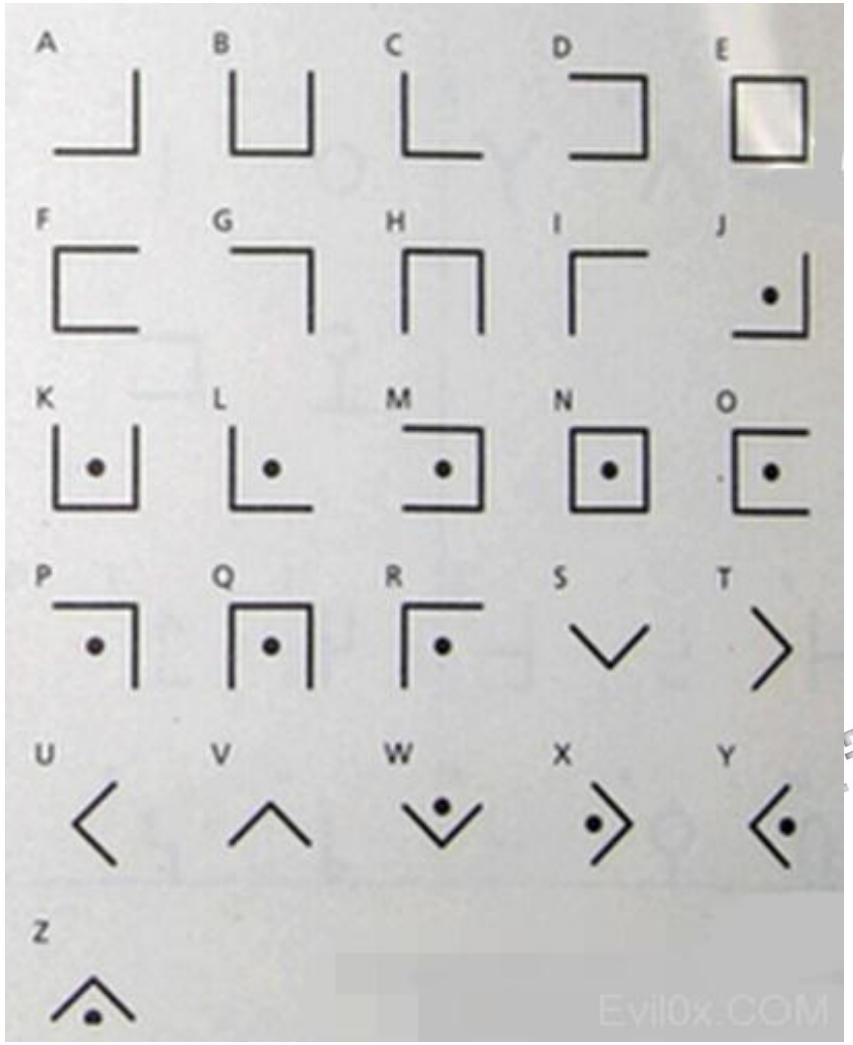
密文：POH

破解希尔密码，只需得到加密矩阵的逆矩阵，再与密文对应的矩阵相乘。

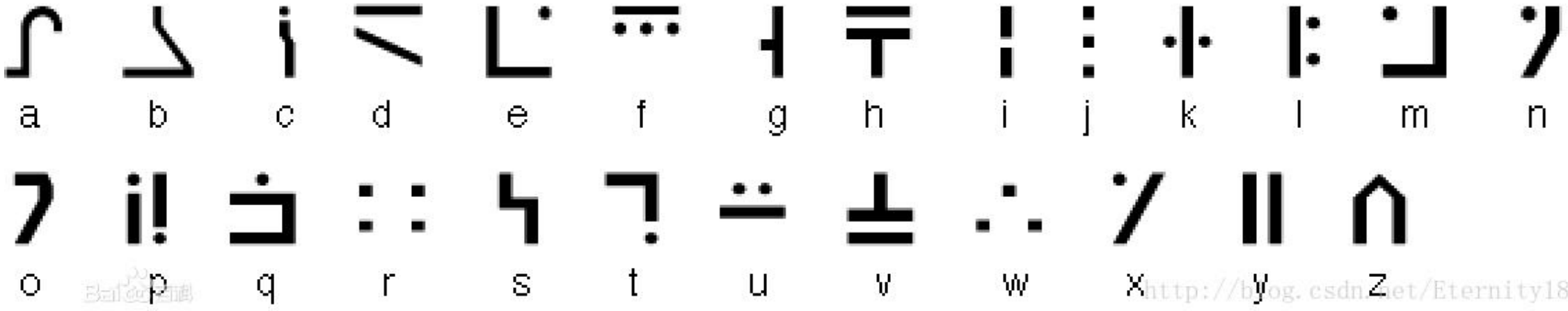
猪圈密码

猪圈密码（Pigpen Cipher或称九宫格密码、朱高密码、共济会密码、共济会员密码），是一种以格子为基础的简单替代式密码。

明文字母和密文对应关系：



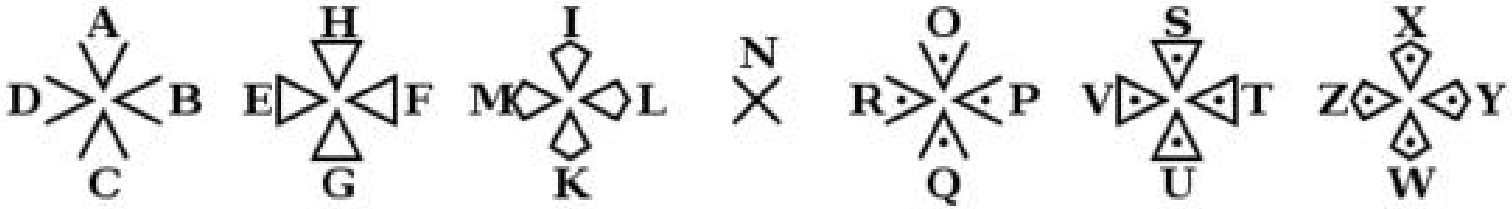
标准银河字母



圣堂武士密码(Templar Cipher)

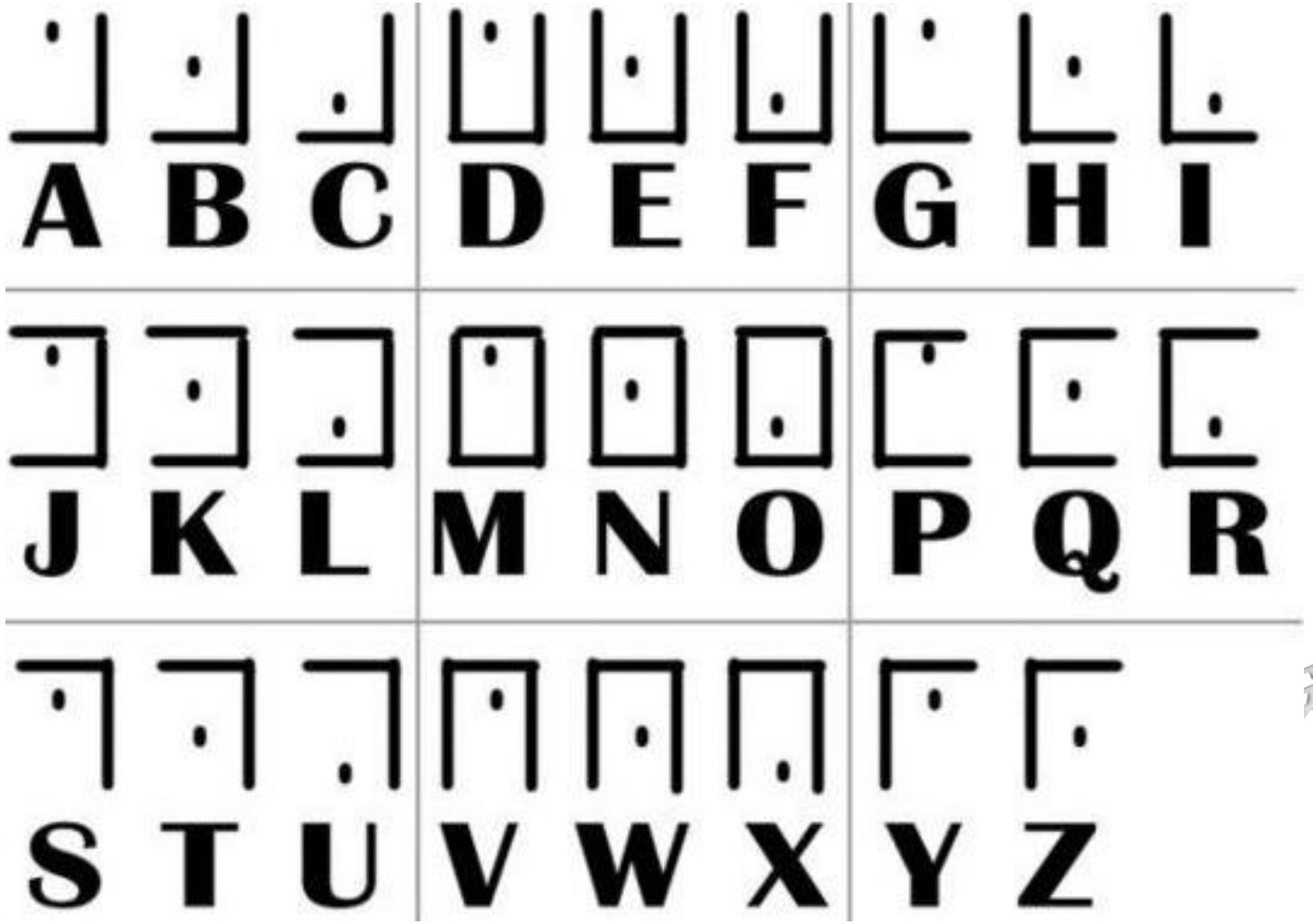
圣堂武士密码(Templar Cipher)是共济会的“猪圈密码”的一个变种，一直被共济会圣殿骑士用。

明文字母和对应密文：



变种密码

明文字母和对应密文：



波利比奥斯方阵密码

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

常见的排布方式：

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

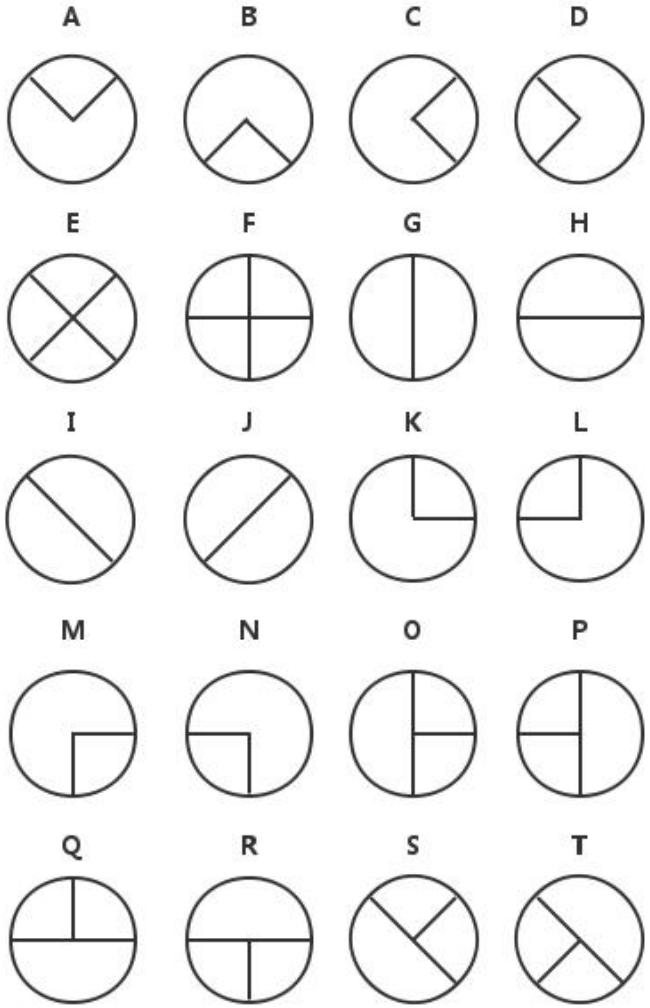
加密实例：

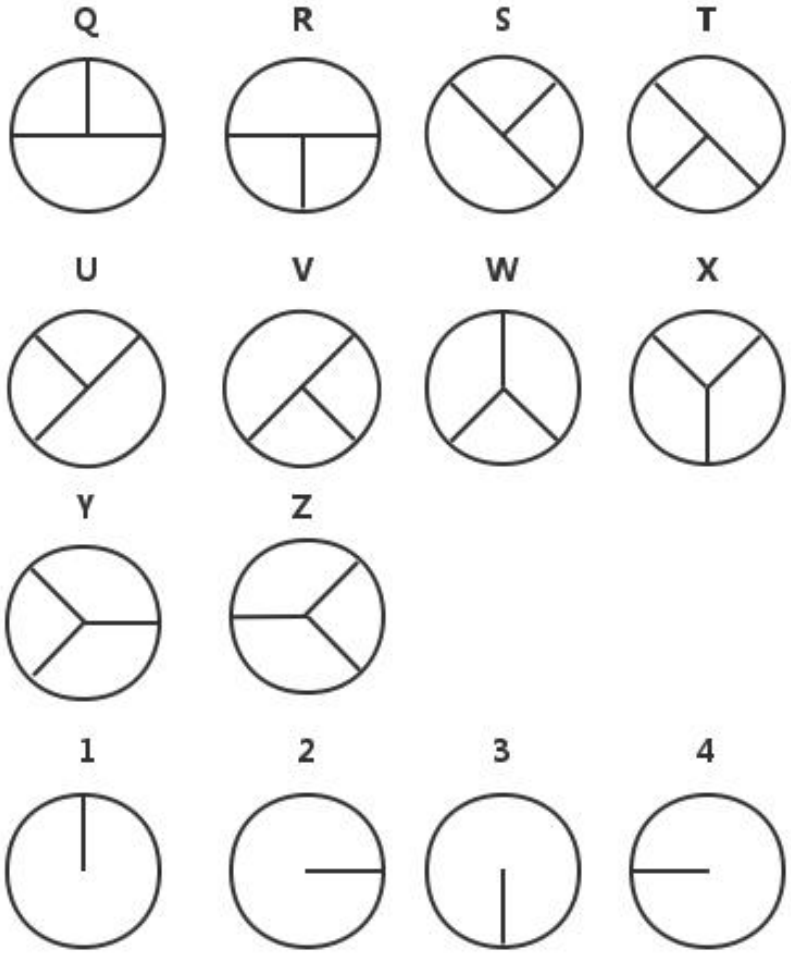
明文： THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

密文： 442315 4145241325 1242345233 213453 2445323543 442315 31115554 143422

夏多密码(曲折加密)

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：



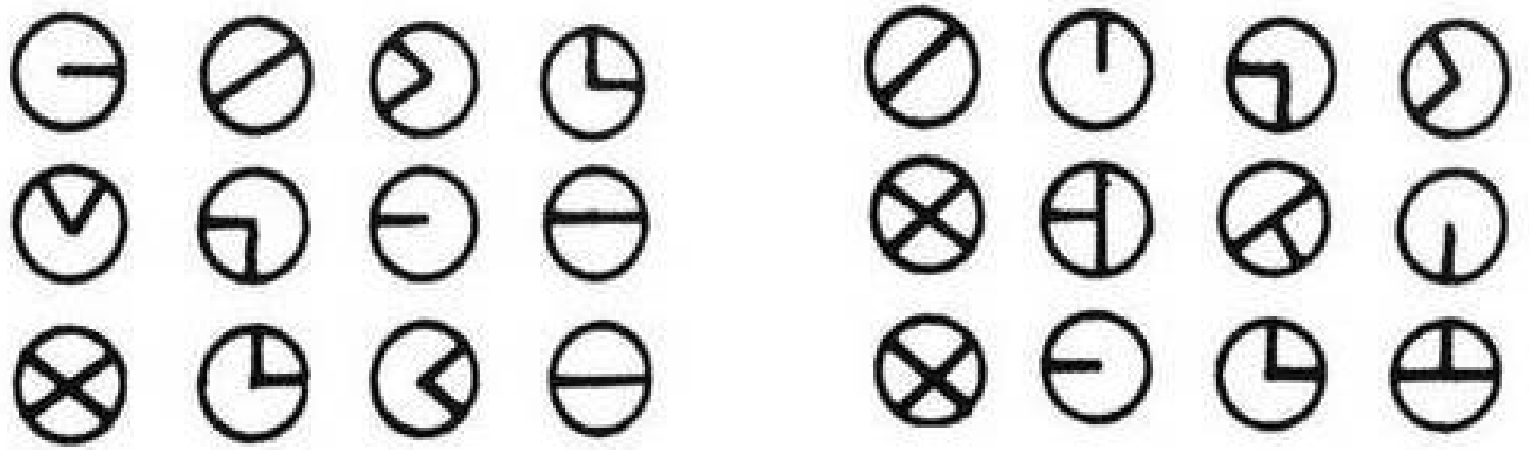


夏多密码(曲折加密)

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示针，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：



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25×25=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 (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	V
2	U	A	H	O	W
3	L	B	I/J	P	X
4	T	D	K	Q	Y
5	R	F	M	S	Z

自动密钥密码

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 FWW ZVDP WVK

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

博福特密码

（2）已知密钥加解密

```
#!python
>>>from pycipher  import Beaufort
>>>Beaufort('CULTURE').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')
'JNHDAJCSTUFYEZOXCZICMOZHCBKARUMVRDY'
>>>Beaufort('CULTURE').decipher('JNHDAJCSTUFYEZOXCZICMOZHCBKARUMVRDY')
'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'
```

滚动密钥密码

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 OGTC TJI DAK YESR

Porta密码

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：

```
#!shell
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

同音替换密码

- 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：

明文: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

密文(其中一种)： 6CZ KOVST XJOMA EQY IOGL4 OW1J UC7 P9NB FOH

- 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" 来破解，外层确定映射的数量，内层确定映射字符。

仿射密码

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，意义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）mod 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。

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) mod 26函数为例子

明文	T	H	E	Q	U	I	C	K	B	R	O	W	N	F	O	X
x	19	7	4	16	20	8	2	10	1	17	14	22	13	5	14	23
(5x + 8)	103	43	28	98	108	48	18	85	13	93	78	118	73	33	78	123
(5x + 8)mod26	25	17	2	10	4	22	18	6	13	15	0	14	21	7	0	19
密文	Z	R	C	K	E	W	S	G	N	P	A	O	V	H	A	T

仿射密码

至于解密我们知道

$$\mathrm{E}(x) = (ax + b) \mod m,$$

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

即可得出解密结果

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

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

以E(x) = (5x + 8) mod 26加密，通过计算可得D(x)=21(x - 8) mod 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)
```

培根密码

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：ＴＨＥＦＯＸ

密文： baaba aabbb aabaa aabab abbab babab

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 (Two-Square)与四方密码类似， 共轭矩阵双密码 (Conjugated Matrix Bifid Cipher)也是双密码的变种。

➢ 示例密阵:

#!shell

	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:

```
#!shell
5121542133 5435452521 3523311521 34
```

在经过密阵转换后密文: WETED TKZNE KYOME X

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3×3×3的密阵代替5×5密阵。

示例密阵:

#!shell

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.

三分密码

经过密阵转换：

T H E Q U I C K B R O W N F O X .

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)：

THEQU ICKBR OWNFO X.

23131 31223 31233 22

33132 22232 13311 21

31132 23221 32113 32

合并：

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

在经过密阵转换后密文:

2313133132311323122322322322132131233113211322132

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

四方密码

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式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
f	g	h	i	k
l	m	n	o	p
q	r	s	t	u
v	w	x	y	z

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

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

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

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

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

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: T H E Q U I C K B R O W N F O X

经过加密棋盘替换得到密文: 72 30 9 34 71 32 4 1 31 35 36 75 74 0 36 77

跨棋盘密码

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

```
  837298372983729837298372983729837
+72309347132413135367574036 77
--------------------------------
  55028174320785018086301224 04
```

在经过棋盘转换后:

55028174320785018086301224 04

ppfmyk n if pfkyfyd hkmmcfc

最终得到密文: ppfmyk n if pfkyfyd hkmmcfc

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:

```
MORSECDABFGHIJKLNPQTUVWXYZ
.........---------XXXXXXXX
...---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。

分组摩尔斯替换密码

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

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:

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

示例密钥矩阵:

#!shell				
T	W	O	H	U
S	A	N	D	R
E	I/J	Y	B	C
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')

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

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

Digrafid密码

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
Ip	#e	Dk	Ck	Zr	Dr	Mx	Ar

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	A	L
6	E	V	U	L	S	I	O	N
7	R	O	W	D	Y	I	S	M
8	S	E	X	T	U	P	L	Y

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。

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。

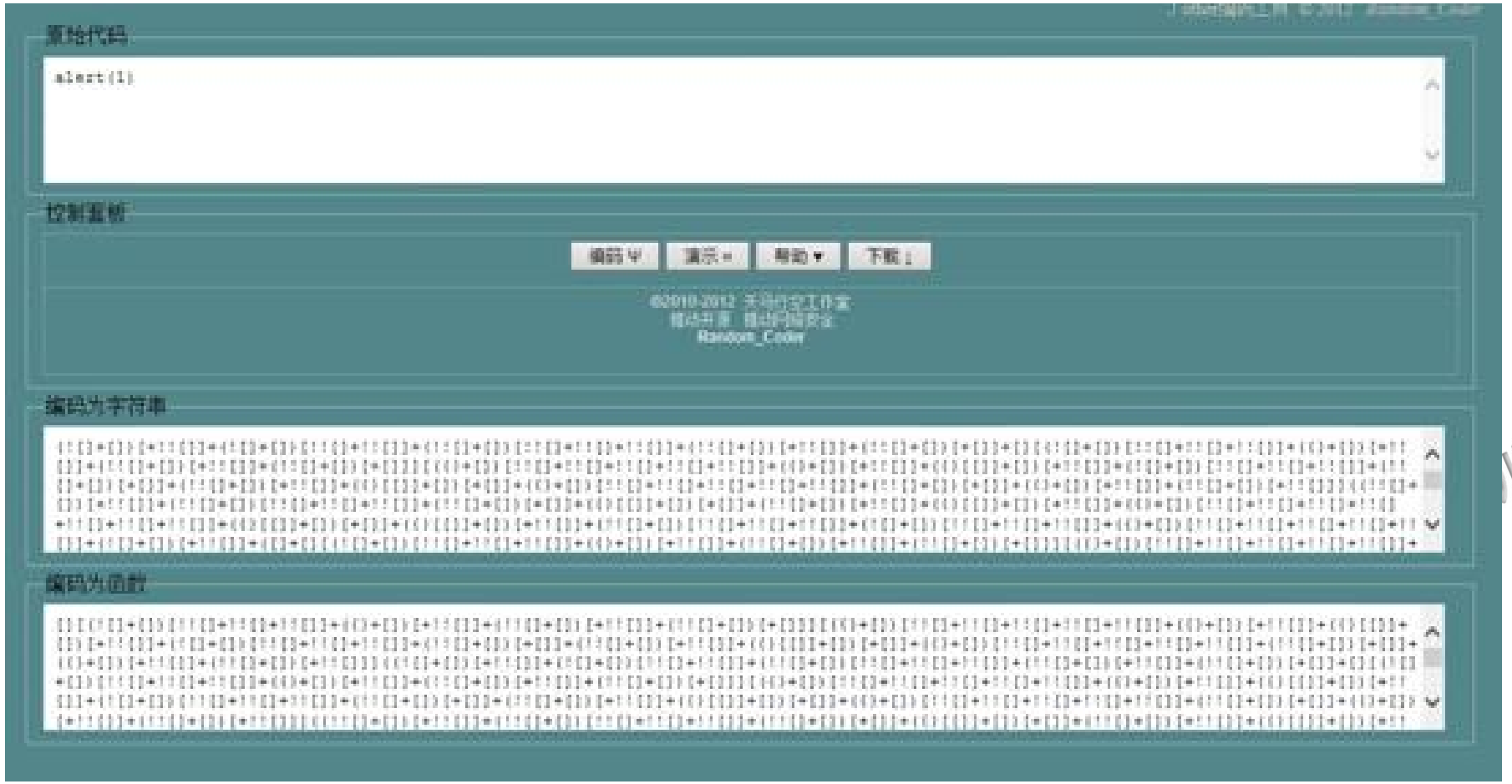
明文：hello!

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

jother

http://tmxk.org/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。



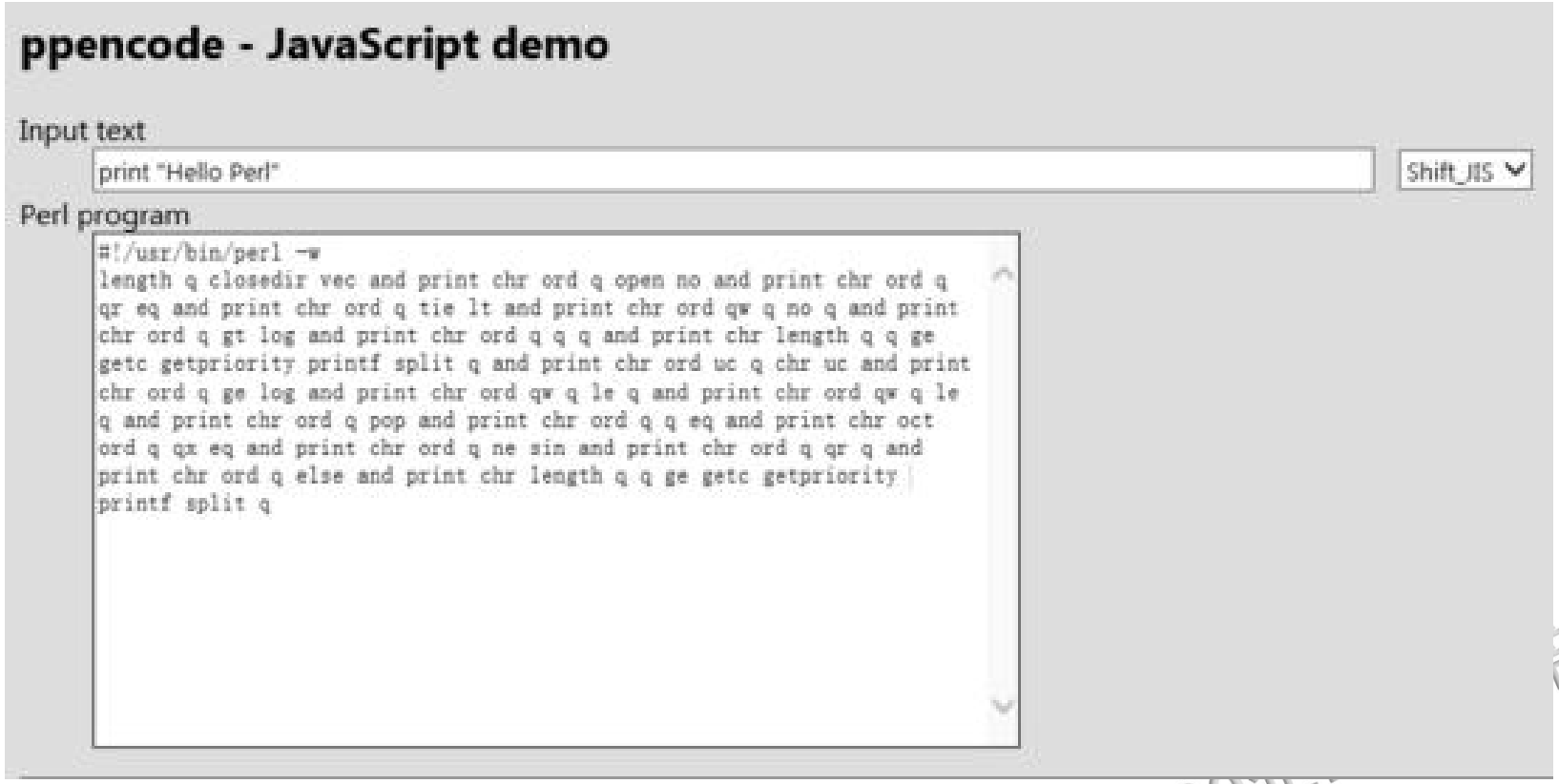
JSfuck

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

http://www.jsfuck.com/

ppencode

ppencode-Perl把Perl代码转换成只有英文字母的字符串。

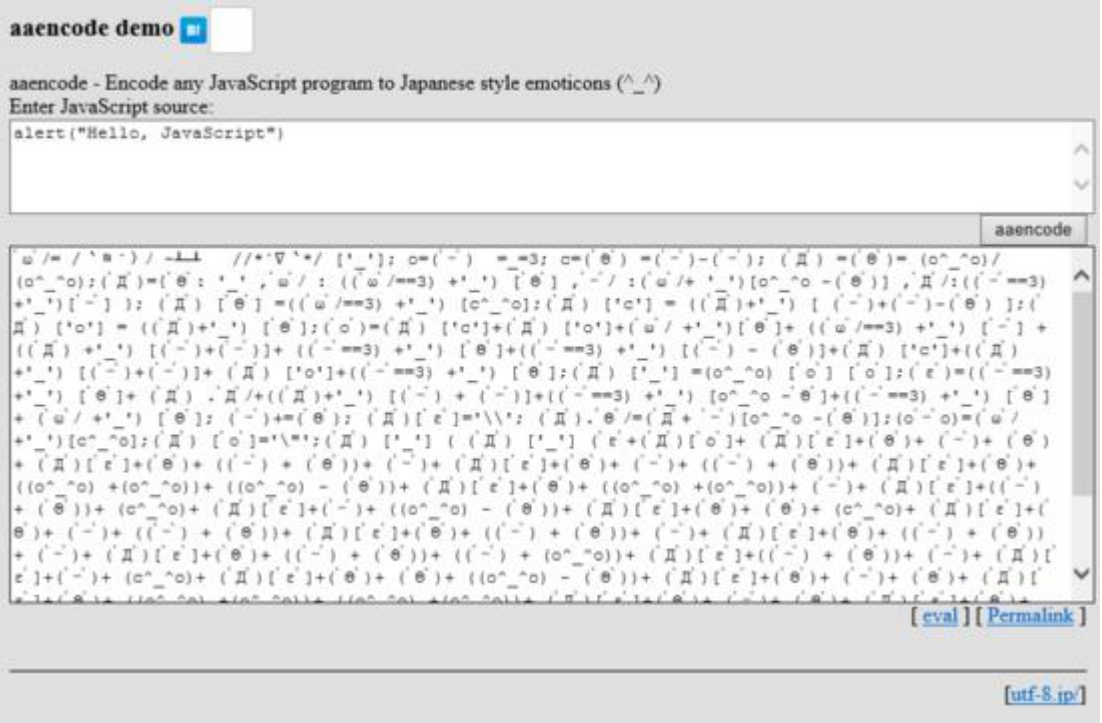


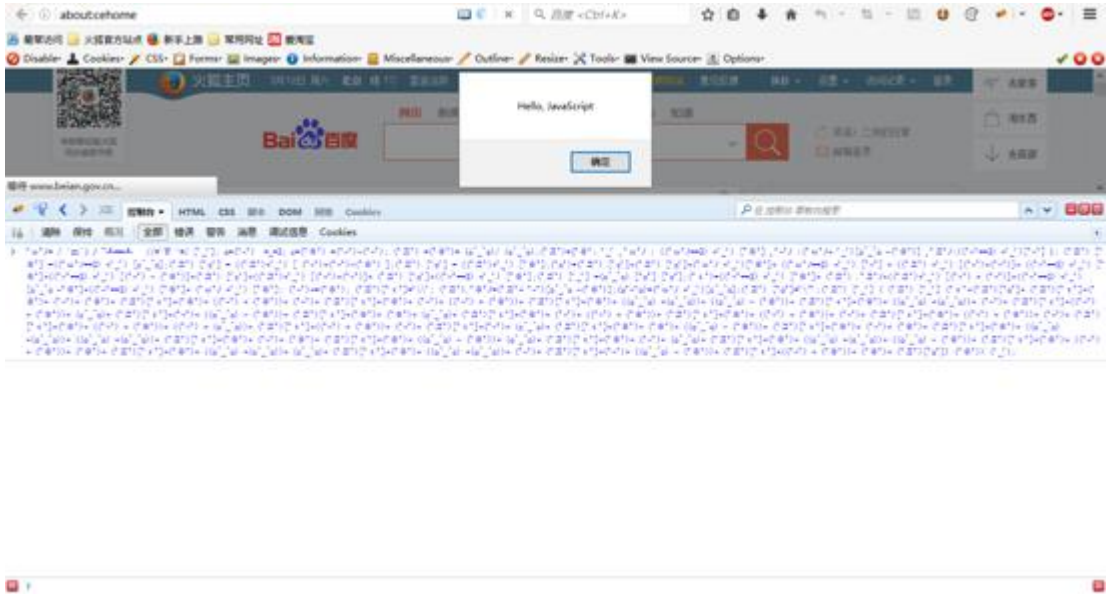
http://namazu.org/~takesako/ppencode/demo.html

jjencode/aaencode

http://utf-8.jp/public/aaencode.html

- jjencode将JS代码转换成只有符号的字符串，类似于rrencode，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。
- 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 ，你也可以在github上 下载 实现jjdecoder的源码进行分析。





潜心读书，专心学习，成就未来