常见密码和编码总结 CTF中Crypto和Misc必备

原创

思源湖的鱼 于 2020-11-20 13:00:40 发布 7843 收藏 63

分类专栏： cyber security 文章标签： 密码学 编码学 ctf misc crypto

本文链接：https://blog.csdn.net/weixin_44604541/article/details/109817480

前言

对常见的编码和密码做个归纳

并记录一些可用的网站和工具

可以当做手册使用

一、常见编码

1、ASCII编码

现今最通用的单字节编码系统，并等同于国际标准ISO/IEC 646

可以分作三部分组成

第一部分是：ASCII非打印控制字符

第二部分是：ASCII打印字符；

第三部分是：扩展ASCII打印字符

ASCII控制字符

二进制	十进制	十六进制	缩写	可以显示的表示法	名称/意义
0000 0000	0	00	NUL	NUL	空字符（Null）
0000 0001	1	01	SOH	SOH	标题开始
0000 0010	2	02	STX	STX	本文开始
0000 0011	3	03	ETX	ETX	本文结束
0000 0100	4	04	EOT	EOT	传输结束
0000 0101	5	05	ENQ	ENQ	请求
0000 0110	6	06	ACK	ACK	确认回应
0000 0111	7	07	BEL	BEL	响铃
0000 1000	8	08	BS	BS	退格
0000 1001	9	09	HT	HT	水平定位符号
0000 1010	10	0A	LF	LF	换行键

0000 1011	11	0B	VT	VT	垂直定位符号
0000 1100	12	0C	FF	FF	换页键
0000 1101	13	0D	CR	CR	归位键
0000 1110	14	0E	SO	SO	取消变换（Shift out）
0000 1111	15	0F	SI	SI	启用变换（Shift in）
0001 0000	16	10	DLE	DLE	跳出数据通讯
0001 0001	17	11	DC1	DC1	设备控制一（XON 启用软件速度控制）
0001 0010	18	12	DC2	DC2	设备控制二
0001 0011	19	13	DC3	DC3	设备控制三（XOFF 停用软件速度控制）
0001 0100	20	14	DC4	DC4	设备控制四
0001 0101	21	15	NAK	NAK	确认失败回应
0001 0110	22	16	SYN	SYN	同步用暂停
0001 0111	23	17	ETB	ETB	区块传输结束
0001 1000	24	18	CAN	CAN	取消
0001 1001	25	19	EM	EM	连接介质中断
0001 1010	26	1A	SUB	SUB	替换
0001 1011	27	1B	ESC	ESC	跳出
0001 1100	28	1C	FS	FS	文件分割符
0001 1101	29	1D	GS	GS	组群分隔符
0001 1110	30	1E	RS	RS	记录分隔符
0001 1111	31	1F	US	US	单元分隔符
0111 1111	127	7F	DEL	DEL	删除

ASCII可显示字符

二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形
0010 0000	32	20	（空格）（SP）	0100 0000	64	40	@	0110 0000	96	60	`
0010 0001	33	21	!	0100 0001	65	41	A	0110 0001	97	61	a
0010 0010	34	22	"	0100 0010	66	42	B	0110 0010	98	62	b
0010 0011	35	23	#	0100 0011	67	43	C	0110 0011	99	63	c
0010 0100	36	24	$	0100 0100	68	44	D	0110 0100	100	64	d
0010 0101	37	25	%	0100 0101	69	45	E	0110 0101	101	65	e
0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f
0010 0111	39	27	'	0100 0111	71	47	G	0110 0111	103	67	g
0010 1000	40	28	(	0100 1000	72	48	H	0110 1000	104	68	h
0010 1001	41	29	)	0100 1001	73	49	I	0110 1001	105	69	i
0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j
0010 1011	43	2B	+	0100 1011	75	4B	K	0110 1011	107	6B	k
0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	l
0010 1101	45	2D	-	0100 1101	77	4D	M	0110 1101	109	6D	m
0010 1110	46	2E	.	0100 1110	78	4E	N	0110 1110	110	6E	n
0010 1111	47	2F	/	0100 1111	79	4F	O	0110 1111	111	6F	o
0011 0000	48	30	0	0101 0000	80	50	P	0111 0000	112	70	p
0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q
0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	114	72	r
0011 0011	51	33	3	0101 0011	83	53	S	0111 0011	115	73	s
0011 0100	52	34	4	0101 0100	84	54	T	0111 0100	116	74	t
0011 0101	53	35	5	0101 0101	85	55	U	0111 0101	117	75	u
0011 0110	54	36	6	0101 0110	86	56	V	0111 0110	118	76	v
0011 0111	55	37	7	0101 0111	87	57	W	0111 0111	119	77	w
0011 1000	56	38	8	0101 1000	88	58	X	0111 1000	120	78	x
0011 1001	57	39	9	0101 1001	89	59	Y	0111 1001	121	79	y
0011 1010	58	3A	:	0101 1010	90	5A	Z	0111 1010	122	7A	z

0011 1011	59	3B	;	0101 1011	91	5B	[	0111 1011	123	7B	{
0011 1100	60	3C	<	0101 1100	92	5C	\	0111 1100	124	7C	|
0011 1101	61	3D	=	0101 1101	93	5D	]	0111 1101	125	7D	}
0011 1110	62	3E	>	0101 1110	94	5E	^	0111 1110	126	7E	~
0011 1111	63	3F	?	0101 1111	95	5F	_				

高四位 / 低四位		扩充ASCII码字符集															
		1000		1001		1010		1011		1100		1101		1110		1111	
		8		9		A/10		B/16		C/32		D/48		E/64		F/80	
		+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符
0000	0	128	Ç	144	É	160	á	176	▒	192	└	208	╨	224	α	240	≡
0001	1	129	ü	145	æ	161	í	177	▒	193	┴	209	╤	225	ß	241	±
0010	2	130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
0011	3	131	â	147	ô	163	ú	179	|	195	├	211	╙	227	π	243	≤
0100	4	132	ä	148	ö	164	ñ	180	┤	196	—	212	Ô	228	Σ	244	⌠
0101	5	133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
0110	6	134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	μ	246	÷
0111	7	135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
1000	8	136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
1001	9	137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	•
1010	A	138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
1011	B	139	ï	155	¢	171	½	187	╗	203	╦	219	█	235	δ	251	√
1100	C	140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	n
1101	D	141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	2
1110	E	142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■
1111	F	143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	BLANK FF

注：表中的ASCII字符可以用：ALT + “小键盘上的数字键”输入

转换网站
ASCII 在线转换器
ASCII编码转换

2、base64,32编码

Base64是网络上最常见的用于传输8Bit字节码的编码方式之一

- 基于64个可打印字符来表示二进制数据的方法
- 3个字节可表示4个可打印字符
- 如果要编码的字节数不能被3整除：当最后剩余一个八位字节（一个byte）时，最后6位的base64字节块有四位是0值，最后附加上两个等号；如果最后剩余两个八位字节（2byte）时，最后一个6位的base字节块有两位是0值，最后附加一个等号

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		

加密：

```
>>> import base64
>>> encode = base64.b64encode(b'I love you')
>>> encode
b'SSBsb3ZlIHlvdQ=='
```

解密：

```
>>> import base64
>>> decode = base64.b64decode(b'SSBsb3ZlIHlvdQ==')
>>> decode
b'I love you'
```

base32

只有大写字母（A-Z）和数字234567

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	9	J	18	S	27	3
1	B	10	K	19	T	28	4
2	C	11	L	20	U	29	5
3	D	12	M	21	V	30	6
4	E	13	N	22	W	31	7
5	F	14	O	23	X	padding	=
6	G	15	P	24	Y		
7	H	16	Q	25	Z		
8	I	17	R	26	2		

跟base64相似就是将base64.b 64encode变成base64.b 32encode
加密：

```
>>> import base64
>>> encode = base64.b32encode(b'I love you')
>>> encode
b'JEQGY33WMUQHS33V'
```

解密：

```
>>> import base64
>>> decode = base64.b32decode(b'JEQGY33WMUQHS33V')
>>> decode
b'I love you'
```

网站
Base64加密解密
base编码

3、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在该字节ascii码的的16进制字符前面加%编码处理

- js：有encodeURI、encodeURIComponent
- PHP有 urlencode、urldecode等

url编码和双重编码是绕过时常用手段

网站
UrlEncode编码/解码
URL编码

4、Unicode编码

unicode编码

- 是一种所有符号的编码，现在的规模可以容纳100多万个符号，https://home.unicode.org/
- 只规定了符号的二进制代码，却没有规定这个二进制代码应该如何存储
- 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

UTF-8

- 对于单字节的符号，字节的第一位设为0，后面7位为这个符号的 Unicode 码。因此对于英语字母，UTF-8 编码和 ASCII 码是相同的
- 于n字节的符号（n > 1），第一个字节的前n位都设为1，第n + 1位设为0，后面字节的前两位一律设为10。剩下的没有提及的二进制位，全部为这个符号的 Unicode 码

Unicode 十六进制码点范围	UTF-8 二进制
0000 0000 - 0000 007F	0xxxxxxx
0000 0080 - 0000 07FF	110xxxxx 10xxxxxx
0000 0800 - 0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
0001 0000 - 0010 FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

二者的转换方式

- 首先找到该Unicode编号所在的编号范围，进而可以找到与之对应的二进制格式
- 然后将该Unicode编号转化为二进制数（去掉高位的0）
- 最后将该二进制数从右向左依次填入二进制格式的X中，如果还有X未填，则设为0

网站
Unicode编码转换

5、HTML实体编码

喜闻乐见的`&#`
HTML 中的预留字符必须被替换为字符实体
一些在键盘上找不到的字符也可以使用字符实体来替换

可参考
HTML 字符实体
HTML 符号实体参考手册
HTML ISO-8859-1 参考手册

网站
在线HTML编码器
HTML编码

6、敲击码

敲击码(Tap code)

- 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
- 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

```
  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
```

源文本	F	O	X
位置	2,1	3,4	5,3
敲击码	.. .	... .	....

网站
敲击码

7、摩尔斯电码（摩斯密码）

大名鼎鼎的morse电码

- 早期的数字化通信形式
- 不同于现代只使用0和1两种状态的二进制代码
- 代码包括五种：
  - 点（.）
  - 划（-）
  - 每个字符间短的停顿（在点和划之间的停顿）
  - 每个词之间中等的停顿
  - 句子之间长的停顿

字母

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	− − . .				

数字

字符	电码符号	字符	电码符号	字符	电码符号	字符	电码符号
0	− − − − −	1	. − − − −	2	. . − − −	3	. . . − −
4	. . . . −	5	. . . . .	6	−	7	− − . . .
8	− − − . .	9	− − − − .				

标点符号

字符	电码符号	字符	电码符号	字符	电码符号	字符	电码符号
.	. − . − . −	:	− − − . . .	,	− − . . − −	;	− . − . − .
?	. . − − . .	=	− . . . −	'	. − − − − .	/	− . . − .
!	− . − . − −	−	− . . . . −	_	. . − − . −	"	. − . . − .
(	− . − − .	)	− . − − . −	$	. . . − . . −	&	. . . .
@	. − − . − .						

非英语字符

字符	电码符号	字符	电码符号	字符	电码符号	字符	电码符号
à或å	. − − . −	ä或æ	. − . −	ch	− − − −	ç或ĉ	− . − . .
ð	. . − − .	é	. . − . .	è	. − . . −	ĝ	− − . − .
ĥ	− . − − .	ĵ	. − − − .	ñ	− − . − −	ö或ø	− − − .
ŝ	. . . − .	þ	. − − . .	ü或ŭ	. . − −		

特殊符号

字符	电码符号	字符	电码符号	字符	电码符号	字符	电码符号
AR	. − . − .	AS	. − . . .	K	− . −	SK	. . . − . −
BT	− . . . −						

网站

中文摩斯密码 Morse莫尔斯电码加密解密

摩尔斯密码在线翻译

Morse code

8、Quoted-printable编码

Quoted-printable编码

- 多用途互联网邮件扩展（MIME) 一种实现方式
- 帮助非ASCII编码的信件传输通过SMTP
- 每个未编码的二进制字符被编码成三个字符，即一个等号和一个十六进制的数字，如'=AB'

编码方法

任何一个8位的字节值可编码为3个字符：一个等号”=”后跟随两个十六进制数字(0-9或A-F)表示该字节的数值.例如，ASCII码换页符（十进制值为12）可以表示为”=0C”，等号”=”（十进制值为61）必须表示为”=3D”. 除了可打印ASCII字符与换行符以外，所有字符必须表示为这种格式.

所有可打印ASCII字符(十进制值的范围为33到126)可用ASCII字符编码来直接表示，但是等号”=”(十进制值为61)不可以这样直接表示.ASCII的水平制表符(tab)与空格符，十进制为9和32，如果不出现在行尾则可以用其ASCII字符编码直接表示。如果这两个字符出现在行尾，必须QP编码表示为”=09” (tab)或”=20” (space).

如果数据中包含有意义的行结束标志，必须转换为ASCII回车(CR)换行(LF)序列，既不能用原来的ASCII字符也不能用QP编码的”=”转义字符序列。 相反，如果字节值13与10有其它的不是行结束的含义，它们必须QP编码为=0D与=0A.

quoted-printable编码的数据的每行长度不能超过76个字符. 为满足此要求又不改变被编码文本，在QP编码结果的每行末尾加上软换行(soft line break). 即在每行末尾加上一个”=”, 但并不会出现在解码得到的文本中.

例如： If you believe that truth=beauty, then surely mathematics is the most beautiful branch of philosophy. 编码后结果是

```
If you believe that truth=3Dbeauty, then surely=20=
mathematics is the most beautiful branch of philosophy.
```


网站
QuotedPrintable编码
Quoted-printable编码

9、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0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

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				O											

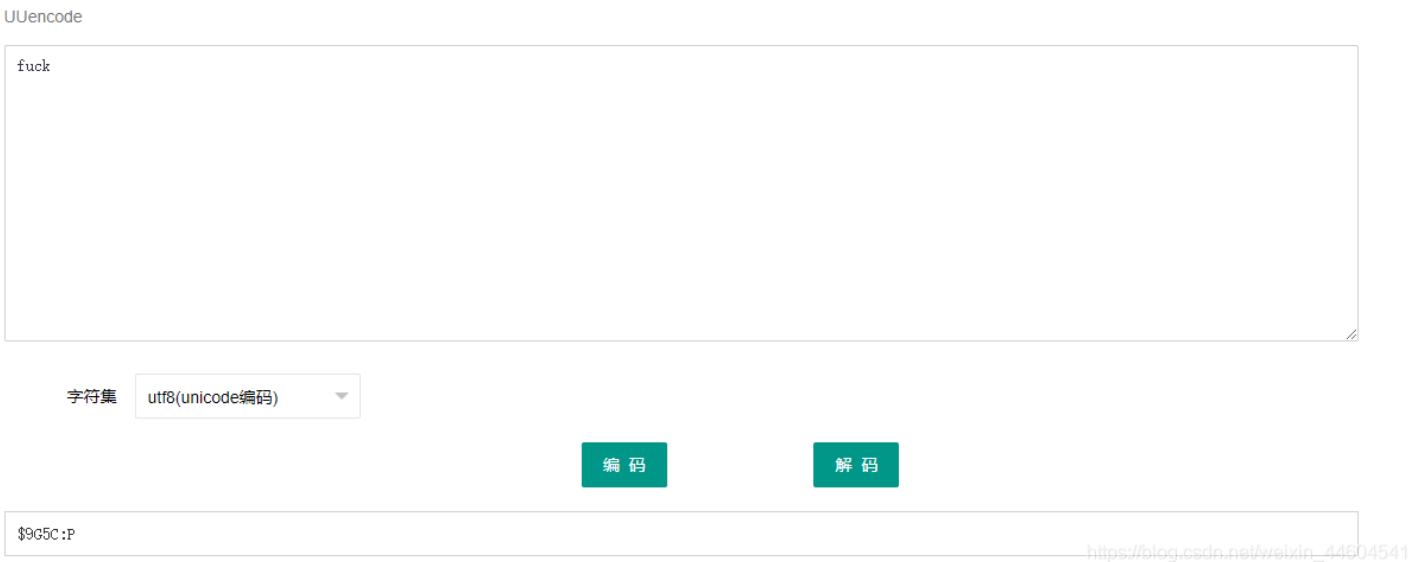
字符串：'Cat' 编码后是：Eq3O

网站
在线XXencode编码
XXencode

10、UUencode编码

UUencode编码

- 起先用在unix网络中，早期在电子邮件中使用较多
- 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0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



网站
在线UUencode编码
UUencode

11、Escape/Unescape编码

Escape/Unescape

- 又叫%u编码，采用UTF-16BE模式，16进制表示方式前面加%u
- 如：字符“中”，UTF-16BE是：“6d93”，因此Escape是“%u6d93”
- 因为目前%字符，常用作URL编码，所以%u这样编码已经逐渐被废弃了

网站
在线Escape编码/加密
Escape编码

12、md5

md5

- 被广泛使用的密码散列函数，可以产生出一个128位（16字节）的散列值（hash value），用于确保信息传输完整一致
- 值范围在`0-9，a-f`

按位补充数据

在MD5算法中，首先需要对信息进行填充，这个数据按位(bit)补充，要求最终的位数对512求模的结果为448。也就是说数据补位后，其位数长度只差64位(bit)就是512的整数倍。即便是这个数据的位数对512求模的结果正好是448也必须进行补位。补位的实现过程：首先在数据后补一个1 bit；接着在后面补上一堆0 bit, 直到整个数据的位数对512求模的结果正好为448。总之，至少补1位，而最多可能补512位 [8] 。

扩展长度

在完成补位工作后，又将一个表示数据原始长度的64 bit数(这是对原始数据没有补位前长度的描述，用二进制来表示)补在最后。当完成补位及补充数据的描述后，得到的结果数据长度正好是512的整数倍。也就是说长度正好是16个(32bit) 字的整数倍 [8] 。

初始化MD缓存器

MD5运算要用到一个128位的MD5缓存器，用来保存中间变量和最终结果。该缓存器又可看成是4个32位的寄存器A、B、C、D，初始化为 [8] :

A： 01 23 45 67

B： 89 ab cd ef

C： fe dc ba 98

D： 76 54 32 10

处理数据段

首先定义4个非线性函数F、G、H、I，对输入的报文运算以512位数据段为单位进行处理。对每个数据段都要进行4轮的逻辑处理，在4轮中分别使用4个不同的函数F、G、H、I。每一轮以ABCD和当前的512位的块为输入，处理后送入ABCD(128位) [8] 。

输出

信息摘要最终处理成以A, B, C, D 的形式输出。也就是开始于A的低位在前的顺序字节，结束于D的高位在前的顺序字节 [8]

md5碰撞

```
import hashlib

for i in range(10000, 10000001):
    s = hashlib.md5(str(i).encode()).hexdigest()[0:5]
    if s == "5fe45":
        print(i)
        break
```

网站

CMD5

xmd5

二、换位密码

1、栅栏密码

把要加密的明文分成N个一组，然后把每组的第1个字连起来，形成一段无规律的话

以2栏栅栏加密为例

- 明文：THE LONGEST DAY MUST HAVE AN END
- 把将要传递的信息中的字母交替排成上下两行。
  T E O G S D Y U T A E N N
  H L N E T A M S H V A E D
- 密文：
  将下面一行字母排在上面一行的后边。
  TEOGSDYUTAENN HLNETAMSHVAED

网站
Rail-fence Cipher
栅栏密码

2、简单换位密码

密文k=“`3124`”
明文m=“`flag{easy_easy_crypto}`”

移位密码首先以k的长度（也就是len(k)=4）切分m，具体如下：

```
flag    {eas    y_ea    sy_c    rypt    o}
```

总共分成了6个部分，然后按照密钥3124的顺序对每一部分都进行密钥变化。如下是变化规则

明文字符位置	1	2	3	4
密文字符位置	3	1	2	4

变化之后，如下：

```
flag    {eas    y_ea    sy_c    rypt    o}

lafg    ea{s    _eya    y_sc    yprt    }o
```

所以密文为：`lafgea{s _eyay_scyprt}o`

3、列移位密码

明文 `The quick brown fox jumps over the lazy dog`
密钥 `how are u`

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)
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

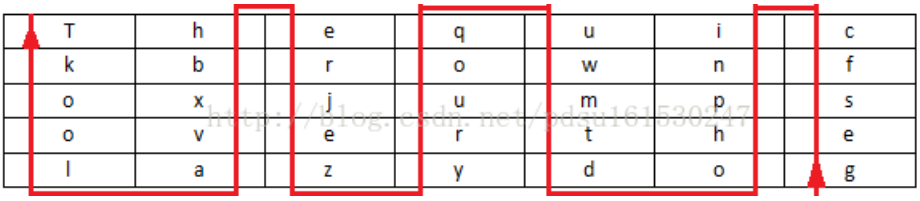
h	o	w	a	r	e	u
3	4	7	1	5	2	6
T	h	e	q	u	i	c
k	b	r	o	w	n	f
o	x	j	u	m	p	s
o	v	e	r	t	h	e
l	a	z	y	d	o	g

密文：`qoury inpho Tkool hbxva uwmtd cfseg erjez`

网站
Columnar Transposition Cipher
列移位密码

4、曲路密码

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`
密文：`gesfc inpho dtmwu qoury zejre hbxva lookT`



三、替换密码

1、凯撒密码

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

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`

网站
Caesar cipher
凯撒密码

2、ROT5/13/18/47

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以例
明文：`the quick brown fox jumps over the lazy dog`
密文：`gur dhvpx oebja sbk whzcf bire gur ynml qbt`

网站
ROT5/13/18/47编码转换
Rot13密码

3、QWE加密

从电脑键盘上的字母从Q开始数，顺序是Q W E R T Y U I...
对应的字母顺序依次是A B C D E F G H 也就是说Q=A,W=B,E=C，依次类推

4、拼音九键加密

利用字母在九键上的位置进行加密
特点：数字为偶数个，且偶数位的数小于5(九键上一个键上的字母最多是四个)

例：335321414374744361715332
两个数为一组分开：33 53 21 41 43 74 74 43 61 71 53 32
对应九键进行查找：3键的第三个字母、5键的第3个字母，以此类推

5、埃特巴什码

以字母倒序排列作为特殊密钥的替换加密，也称也就是下面的对应关系：

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ZYXWVUTSRQPONMLKJIHGFEDCBA
```

网站
Atbash Cipher
埃特巴什码

6、培根密码

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)

```
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
```

网站
Baconian Cipher
培根密码

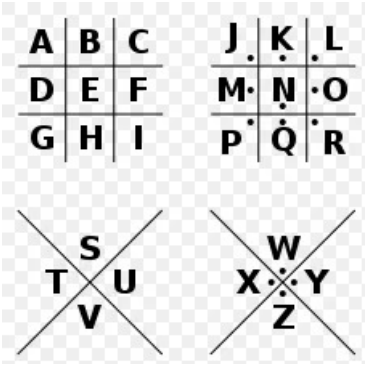
7、希尔密码

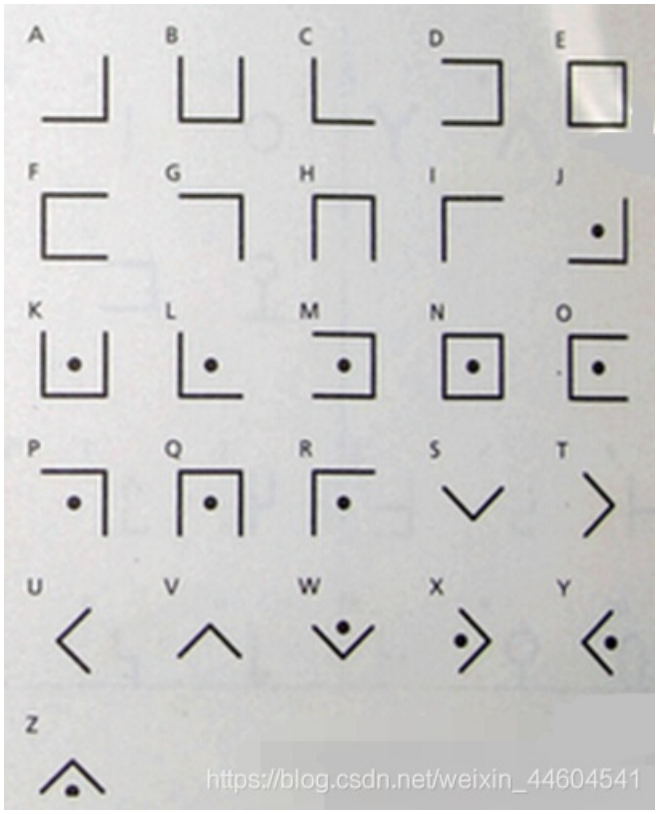
每个字母转换成26进制数字：A=0, B=1, C=2…Z=25
一串字母当成n维向量，跟一个n×n的矩阵相乘
再将得出的结果MOD26

网站
Hill Cipher
Cryptanalysis of the Hill Cipher

8、猪圈密码

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



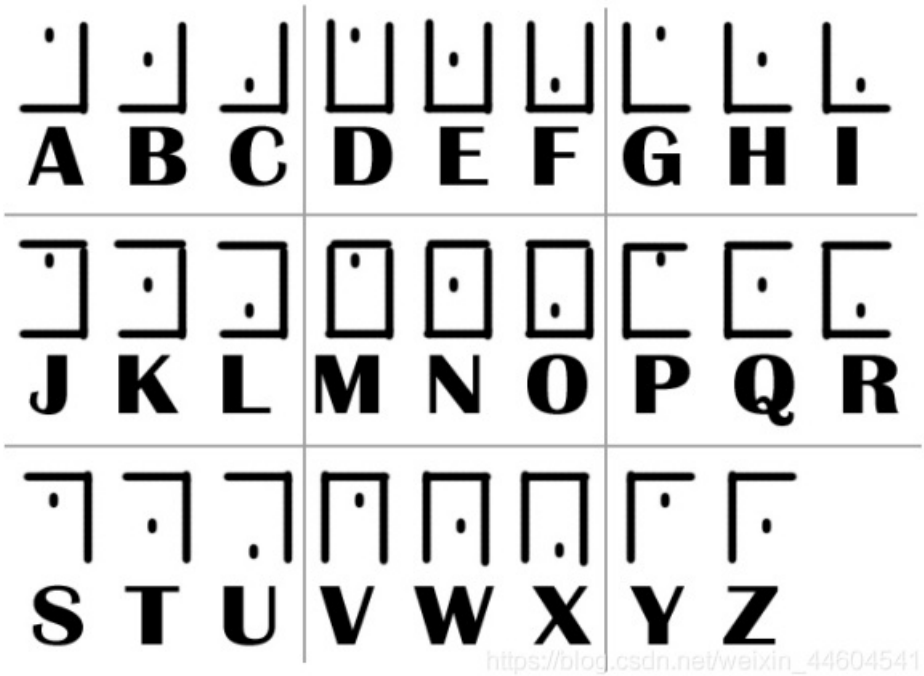


网站

The BLACK Chamber

猪圈密码

变种



9、圣堂武士密码

也算是猪圈密码的变种

10、银河字母

11、维吉尼亚密码

维吉尼亚密码(Vigenère Cipher)
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

	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`

网站

- Cryptanalysis of the Vigenere Cipher
- Vigenère cipher
- Vigenere Solver
- 维吉尼亚密码

实例

- 攻防世界 Crypto高手进阶区 3分题 shanghai

12、格罗斯费尔德密码

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(kasiski)来破译

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

网站
Gronsfeld Cipher
Gronsfeld密码

13、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`

网站
Cryptanalysis of the Autokey Cipher
Autokey Cipher
自动密钥密码

14、博福特密码

博福特密码(Beaufort Cipher)

- 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
- 最知名的应用是Hagelin M-209密码机
- 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`

网站
Beaufort Cipher
博福特密码

15、滚动密钥密码

滚动密钥密码(Running Key Cipher)

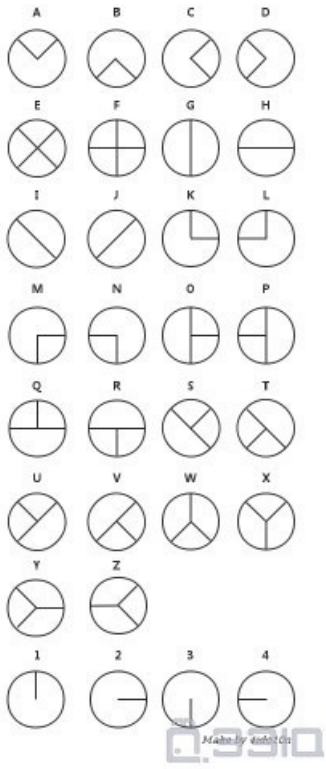
- 和维吉尼亚密码有着相同的加密机制，区别是密钥的选取，维吉尼亚使用的密钥简短，而且重复循环使用，与之相反，滚动密钥密码使用很长的密钥，比如引用一本书作为密钥
- 这样做的目的是不重复循环使用密钥，使密文更难破译，尽管如此，滚动密钥密码还是可以被攻破，因为有关于密钥和明文的统计分析模式可供利用，如果滚动密钥密码使用统计上的随机密钥来源，那么理论上是不可破译的，因为任何可能都可以成为密钥，并且所有的可能性都是相等的。

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`

网站
Running Key Cipher
滚动密钥密码

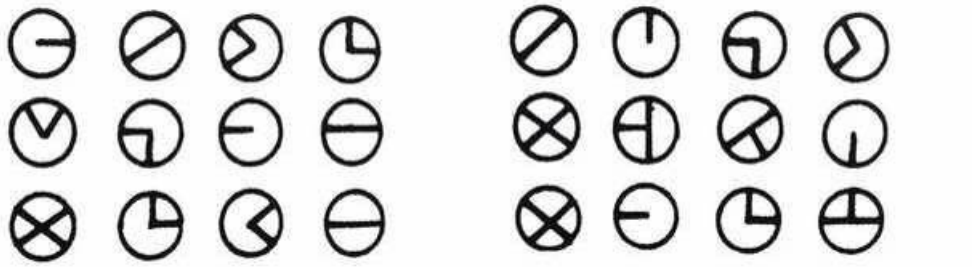
16、夏多密码（曲折加密）

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



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

例：信文：`I AM IN DANGER.SEND HELP`(我有危险，速来增援)
可以加密成



17、波利比奥斯方阵密码

波利比奥斯方阵密码（Polybius Square Cipher或称波利比奥斯棋盘）

- 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
  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	S	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`

18、普莱菲尔密码

普莱菲尔密码(Playfair Cipher)

- 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
- 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)

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

（1）编制密码表

- 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
- 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

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”I”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“：

- 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
- 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：

- 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”EG“；
- 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“；
- 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 CG 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

密文`KUNDL HGTLF WUESP WLHSI NPCGC RAGBU VZQAI V`

网站
Playfair Cipher
普莱菲尔密码

19、ADFGX密码

ADFGX密码(ADFGX Cipher)

- 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`
密文：`XF AD DA AF XD XG GA FG XA FX DX GX DG FA DX FF`

网站
ADFGX Cipher
ADFGX密码

20、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 0 q g 6
D | 4 m e a 1 y
F | l 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
```

网站
ADFGVX密码

21、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
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`

网站
Bifid Cipher
Cryptanalysis of the Bifid cipher
双密码

22、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:

```
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：

```
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:

```
2313133132311323122322232232213123313311321132222132
N  O  O  N  W  G  B  X  X  L  G  H  H  W  S  K  W
```

23、四方密码

四方密码(Four-Square Cipher)

- 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
- 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   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 b c d e
C R H S A   f g h i k
X Y O G V   l m n o p
I T U E W   q r s t u
L Q Z K P   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`

网站
Four-Square Cipher
Cryptanalysis of the Foursquare Cipher
四方密码

24、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`

25、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`

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
+723093471324131335367574036777
-----------------------------
 550281743207850180863012240 4
```

在经过棋盘转换后：

```
550281743207850180863012240 4
ppfmyk n if  pfkyfyd hkmmcfc
```

最终得到密文：`ppfmyk n in pfkyfyd hkmmcfc`

网站
Straddle Checkerboard Cipher

26、云影密码

采用的是0作间隔，其他非0数隔开后组合起来相加表示26个字母

脚本

```python
#!/usr/bin/python
# -*- coding=utf8 -*-
"""
# @Author : pig
# @CreatedTime:2019-11-2423:54:02
# @Description :
"""


def de_code(c):
    dic = [chr(i) for i in range(ord("A"), ord("Z") + 1)]
    flag = []
    c2 = [i for i in c.split("0")]
    for i in c2:
        c3 = 0
        for j in i:
            c3 += int(j)
        flag.append(dic[c3 - 1])
    return flag

def encode(plaintext):
    dic = [chr(i) for i in range(ord("A"), ord("Z") + 1)]
    m = [i for i in plaintext]
    tmp = [];flag = []
    for i in range(len(m)):
        for j in range(len(dic)):
            if m[i] == dic[j]:
                tmp.append(j + 1)
    for i in tmp:
        res = ""
        if i >= 8:
            res += int(i/8)*"8"
        if i%8 >=4:
            res += int(i%8/4)*"4"
        if i%4 >=2:
            res += int(i%4/2)*"2"
        if i%2 >= 1:
            res += int(i%2/1)*"1"
        flag.append(res + "0")
    print ("".join(flag)[:-1])

c = input("输入要解密的数字串:")
print (de_code(c))
m_code = input("请输入要加密的数字串:")
encode(m_code)
```

例

```python
a = "884210122048022440401422402480122"
a = a.split("0")
flag = ''
for w in a:
    sum = 0
    for i in w:
        sum += int(i)
    flag += chr(sum + 64)
print(flag)
```

27、Porta密码

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

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`

网站
Porta Cipher
Porta密码

28、仿射密码

仿射密码(Affine Cipher)

- 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

以E(x) = (5x + 8) mod 26函数为例

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

解密用D(x)=21(x - 8) mod 26

网站
Affine Cipher
仿射密码

29、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  Y
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  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')
```

30、当铺密码

当铺密码 就是一种将中文和数字进行转化的密码，算法相当简单:当前汉字有多少笔画出头，就是转化成数字几

例如:

```
王夫 井工 夫口 由中人 井中 夫夫 由中大
67 84 70 123 82 77 125
```

四、现代密码

1、RSA

RSA是目前最有影响力和最常用的公钥加密算法，它能够抵抗到目前为止已知的绝大多数密码攻击，已被ISO推荐为公钥数据加密标准。

今天只有短的RSA钥匙才可能被强力方式解破。到2008年为止，世界上还没有任何可靠的攻击RSA算法的方式。只要其钥匙的长度足够长，用RSA加密的信息实际上是不能被解破的。目前普遍认为，模式n至少应该取1024位，最好是2048位。但在分布式计算和量子计算机理论日趋成熟的今天，RSA加密安全性受到了挑战和质疑。

RSA算法基于一个十分简单的数论事实:将两个大质数相乘十分容易，但是想要对其乘积进行因式分解却极其困难，因此可以将乘积公开作为加密密钥

脚本

```
import libnum
from Crypto.Util.number import long_to_bytes

q = int("0xa6055ec186de51800ddd6fcbf0192384ff42d707a55f57af4fcfb0d1dc7bd97055e8275cd4b78ec63c5d592f567c66393a061324aa2e6a8d8fc2a910cbee1ed9",16)
p = int("0xfa0f9463ea0a93b929c099320d31c277e0b0dbc65b189ed76124f5a1218f5d91fd0102a4c8de11f28be5e4d0ae91ab319f4537e97ed74bc663e972a4a9119307",16)

e = int("0x6d1fdab4ce3217b3fc32c9ed480a31d067fd57d93a9ab52b472dc393ab7852fbcb11abbebfd6aaae8032db1316dc22d3f7c3d631e24df13ef23d3b381a1c3e04abcc745d402ee3a031ac2718fae63b240837b4f657f29ca4702da9af22a3a019d68904a969ddb01bcf941df70af042f4fae5cbeb9c2151b324f387e525094c41",16)

c = 0x7fe1a4f743675d1987d25d38111fae0f78bbea6852cba5beda47db76d119a3efe24cb04b9449f53becd43b0b46e269826a983f832abb53b7a7e24a43ad15378344ed5c20f51e268186d24c76050c1e73647523bd5f91d9b6ad3e86bbf9126588b1dee21e6997372e36c3e7428473474889182966508 6e0dc523ed23c386bb520

n = q*p

d = libnum.invmod(e, (p - 1) * (q - 1))
m = pow(c, d, n)   # m 的十进制形式
string = long_to_bytes(m)  # m明文
print(string)
```

网站
在线RSA公钥加密解密
在线RSA私钥加密解密

2、AES

在密码学中又称Rijndael加密法，是美国联邦政府采用的一种区块加密标准。这个标准用来替代原先的DES，已经被多方分析且广为全世界所使用。经过五年的甄选流程，高级加密标准由美国国家标准与技术研究院(NIST)于2001年11月26日发布于FIPS PUB 197，并在2002年5月26日成为有效的标准。2006年，高级加密标准已然成为对称密钥加密中最流行的算法之一。

aes密文包括字母 数字 + = /等

网站
AES加密
在线加密解密

3、DES

DES算法为密码体制中的对称密码体制，又被称为美国数据加密标准，是1972年美国IBM公司研制的对称密码体制加密算法。明文按64位进行分组，密钥长64位，密钥事实上是56位参与DES运算（第8、16、24、32、40、48、56、64位是校验位， 使得每个密钥都有奇数个1）分组后的明文组和56位的密钥按位替代或交换的方法形成密文组的加密方法

网站
DES算法原理完整版
在线DES加密解密

4、ECC

ECC（Elliptic Curves Cryptography，椭圆曲线密码编码学）
椭圆加密算法(ECC)是一种公钥加密体制，最初由Koblitz和Miller两人于1985年提出，其数学基础是利用椭圆曲线上的有理点构成Abel加法群上椭圆离散对数的计算困难性
公钥密码体制根据其所依据的难题一般分为三类:大整数分解问题类、离散对数问题类、椭圆曲线类
有时也把椭圆曲线类归为离散对数类

网站
ECC(椭圆曲线密码编码学)简介

5、RC4

RC4（来自Rivest Cipher 4的缩写）是一种流加密算法，密钥长度可变。它加解密使用相同的密钥，因此也属于对称加密算法。RC4是有线等效加密（WEP）中采用的加密算法，也曾经是TLS可采用的算法之一

网站
RC4算法总结
RC4加解密

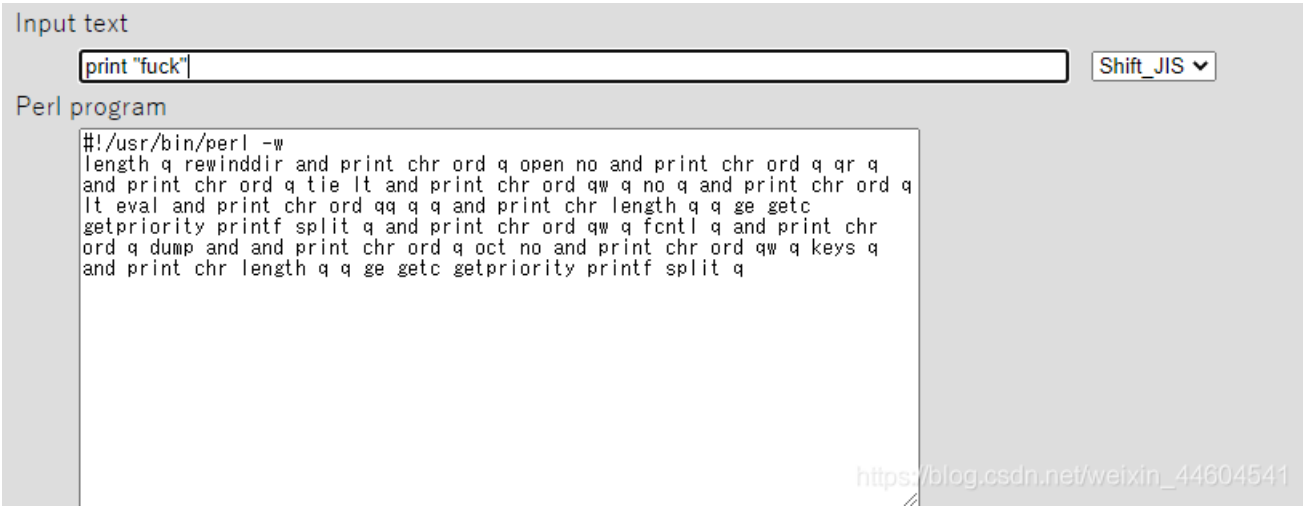
五、代码混淆加密

1、VBScript.Encode解密

VBScript.Encode解密

2、ppencode

ppencode-Perl把Perl代码转换成只有英文字母的字符串



网站
ppencode - JavaScript demo
PPencode

3、rrencode

rrencode可以把ruby代码全部转换成符号

暂时没找到网站
之前有个挂了

4、jjencode

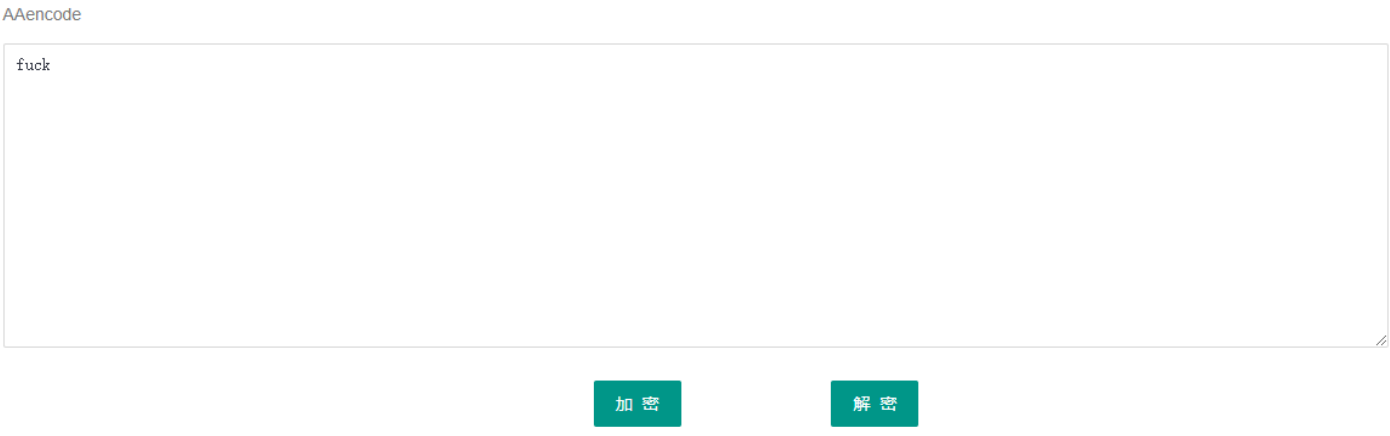
jjencode将JS代码转换成只有符号的字符串
jjencode的解密直接在浏览器的控制台里输入密文即可执行解密


```
$=~[];$={___:++$,$$$$:(![]+"")[$],__$:++$,$_$_:(![]+"")[$],_$_:++$,$_$$:({}+"")[$],$$_$:($[$]+"")[$],_$$:++$,$$$_:(!""+"")[$],$__:++$,$_$:++$,$$__:({}+"")[$],$$_:++$,$$$:++$,$___:++$,$__$:++$};$.$_=($.$_=$+"")[$.$_$]+($._$=$.$_[$.__$])+($.$$=($.$+"")[$.__$])+((!$)+"")[$._$$]+($.__=$.$_[$.$$_])+($.$=(!""+"")[$.__$])+($._=(!""+"")[$._$_])+$.$_[$.$_$]+$.__+$._$+$.$;$.$$=$.$+(!""+"")[$._$$]+$.__+$._+$.$+$.$$;$.$=($.___)[$.$_]
[$.$_];$.$($.$($.$+"\""+$.$$$$+$._+$.$$__+"\"")+$.__$+$.$_$+$._$$+"\"")())();
```

网站
JJencode
MALWARE ANALYSIS

5、aaencode/颜文字

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

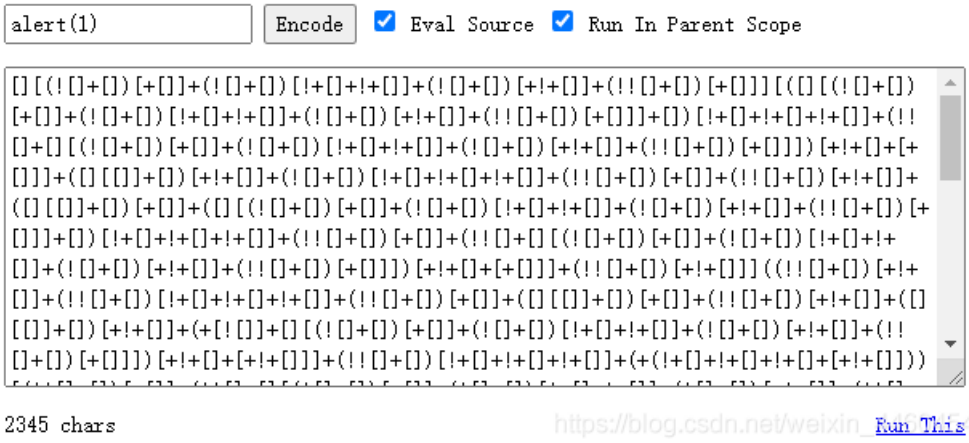



```
ﾟωﾟ/= /｀ｍ´）/ ~┻━┻   //*´∇｀*/ ['_']; o=(ﾟｰﾟ)  =_=3; c=(ﾟΘﾟ) =(ﾟｰﾟ)-(ﾟｰﾟ); (ﾟДﾟ) =(ﾟΘﾟ)= (o^_^o)/ (o^_^o);(ﾟДﾟ)=[ﾟΘﾟ: '_' ,ﾟωﾟ/ : ((ﾟωﾟ/==3) +'_') [ﾟΘﾟ] ,ﾟｰﾟ/ :(ﾟωﾟ/+ '_')[o^_^o -(ﾟΘﾟ)] ,ﾟДﾟ/:((ﾟｰﾟ==3) +'_')[ﾟｰﾟ] }; (ﾟДﾟ) [ﾟΘﾟ] =((ﾟωﾟ/==3) +'_') [c^_^o];(ﾟДﾟ) ['c'] = ((ﾟДﾟ)+'_') [ (ﾟｰﾟ)+(ﾟｰﾟ)-(ﾟΘﾟ) ];(ﾟДﾟ) ['o'] = ((ﾟДﾟ)+'_') [ﾟΘﾟ];(ﾟoﾟ)=(ﾟДﾟ) ['c']+(ﾟДﾟ) ['o']+(ﾟωﾟ/ +'_')[ﾟΘﾟ]+ ((ﾟωﾟ/==3) +'_') [ﾟｰﾟ] + ((ﾟДﾟ) +'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ ((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+((ﾟｰﾟ==3) +'_') [(ﾟｰﾟ) - (ﾟΘﾟ)]+(ﾟДﾟ) ['c']+((ﾟДﾟ)+'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ (ﾟДﾟ) ['o']+((ﾟｰﾟ==3) +'_') [ﾟΘﾟ];(ﾟДﾟ) ['_'] =(o^_^o) [ﾟoﾟ] [ﾟoﾟ];(ﾟε ﾟ)=((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+ (ﾟДﾟ) .ﾟДﾟ/+((ﾟДﾟ)+'_') [(ﾟｰﾟ) + (ﾟｰﾟ)]+((ﾟｰﾟ==3) +'_') [o^_^o -ﾟΘﾟ]+((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+ (ﾟωﾟ/ +'_') [ﾟΘﾟ]; (ﾟｰﾟ)+=(ﾟΘﾟ); (ﾟДﾟ)[ﾟε ﾟ]='\\'; (ﾟДﾟ).ﾟΘﾟ/=(ﾟДﾟ+ ﾟｰﾟ)[o^_^o -(ﾟΘﾟ)];(oﾟｰﾟo)=(ﾟωﾟ/ +'_')[c^_^o];(ﾟДﾟ) [ﾟoﾟ]='\"';(ﾟДﾟ) ['_'] ( (ﾟДﾟ) ['_'] (ﾟε ﾟ+(ﾟДﾟ)[ﾟoﾟ]+(ﾟДﾟ)[ﾟε ﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟε ﾟ]+(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟε ﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ (o^_^o)+ (ﾟДﾟ)[ﾟε ﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (o^_^o)+ (ﾟДﾟ)[ﾟε ﾟ]+(ﾟΘﾟ)+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟoﾟ]) (ﾟΘﾟ)) ('_');
```

网站
aaencode demo
aadecode - Decode encoded-as-aaencode JavaScript program. ['_']
AAencode

6、JSfuck

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



网站

JSFuck

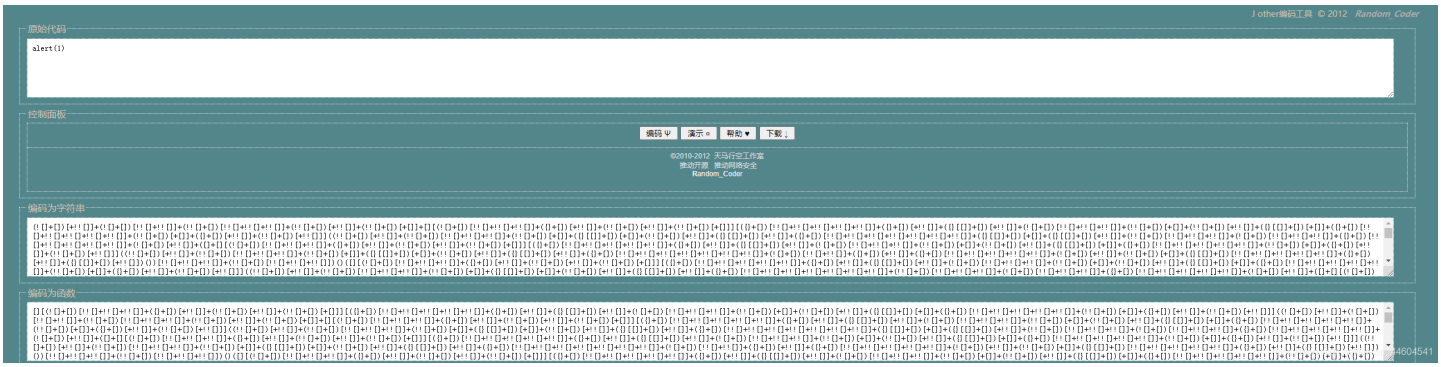
JSfuck

7、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

直接在浏览器(f12)的控制台里输入密文即可执行解密



网站

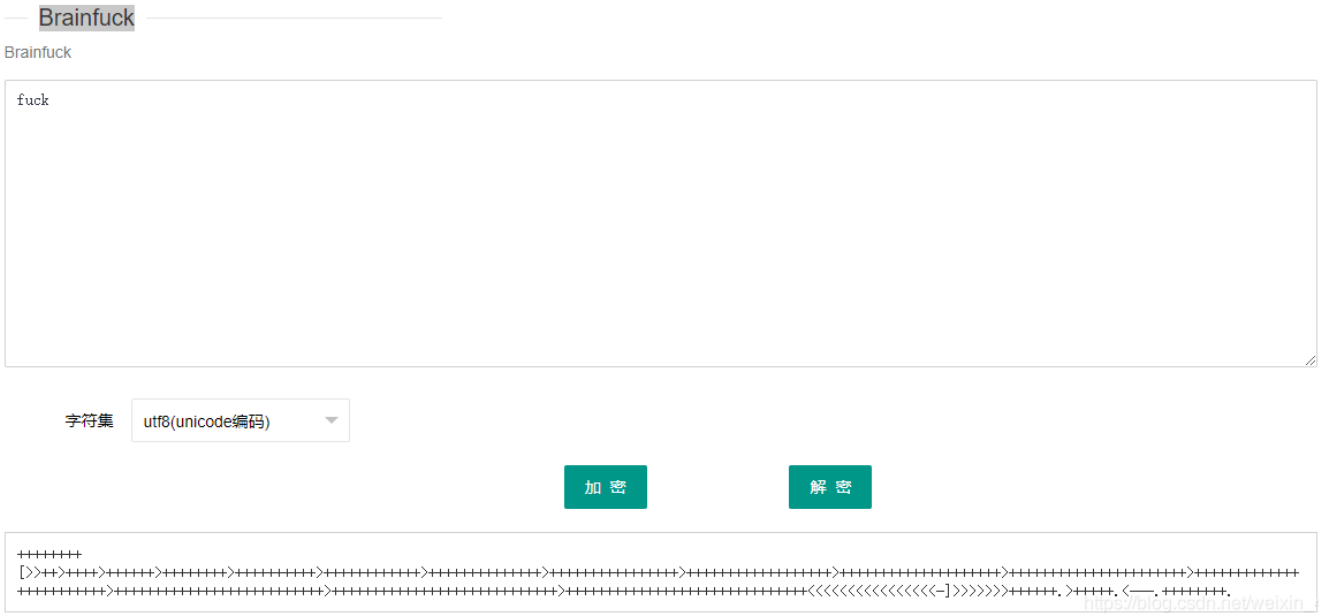
jother

8、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

Brainfuck只有八种符号，所有的操作都由这八种符号`> < + - . , [ ]`的组合来完成



网站

Brainfuck/Ook! Obfuscation/Encoding

Brainfuck

9、Ook!

Ook! 与Brainfuck类似, 但用单词“Ook！”，“Ook.” 和“Ook?”代替

A part of "Hello World" example:

LINENUMBERZEROCODEPRINTZEROGOTOONELINENUMBERONECODEPRINTONEGOTOONEZEROLINENUMBE

RONEZEROCODEPRINTZEROGOTOONEONELINENUMBERONEONECODEPRINTZEROGOTOONEZEROZEROLINE

NUMBERONEZEROZEROCODEPRINTONEGOTOONEZEROO

Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.

Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook.

Ook! Ook. Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.

Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook?

Ook! Ook! Ook? Ook! Ook? Ook. Ook. Ook. Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook.

Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook! Ook. Ook. Ook. Ook. Ook.

Ook. Ook. Ook! Ook. Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook.

Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook.

Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook. Ook! Ook.

Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.

Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook.

Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.

Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook.

Ook? Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook.

Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook.

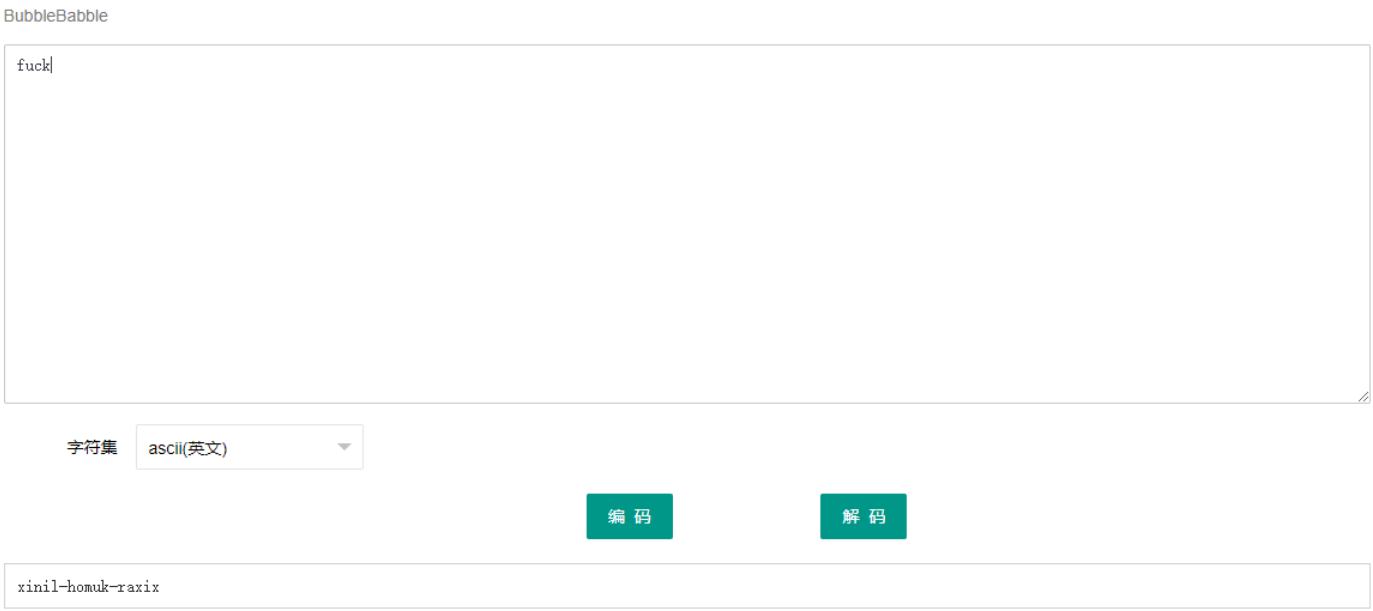
网站

Brainfuck/Ook! Obfuscation/Encoding

10、Bubble Babble

Bubble Babble

- 由Antti Huima创建的一种编码方法，可以把二进制信息表示为由交替的元音和辅音组成的伪词（pseudo-words）
- 主要用于密码指纹，其编码也具有内置的纠错和冗余
- 编码格式每5个字符中间以`-`来分隔，作者的原意就是想把难以记得的二进制数据表示为难忘的伪词



网站
bubblepy
BubbleBabble

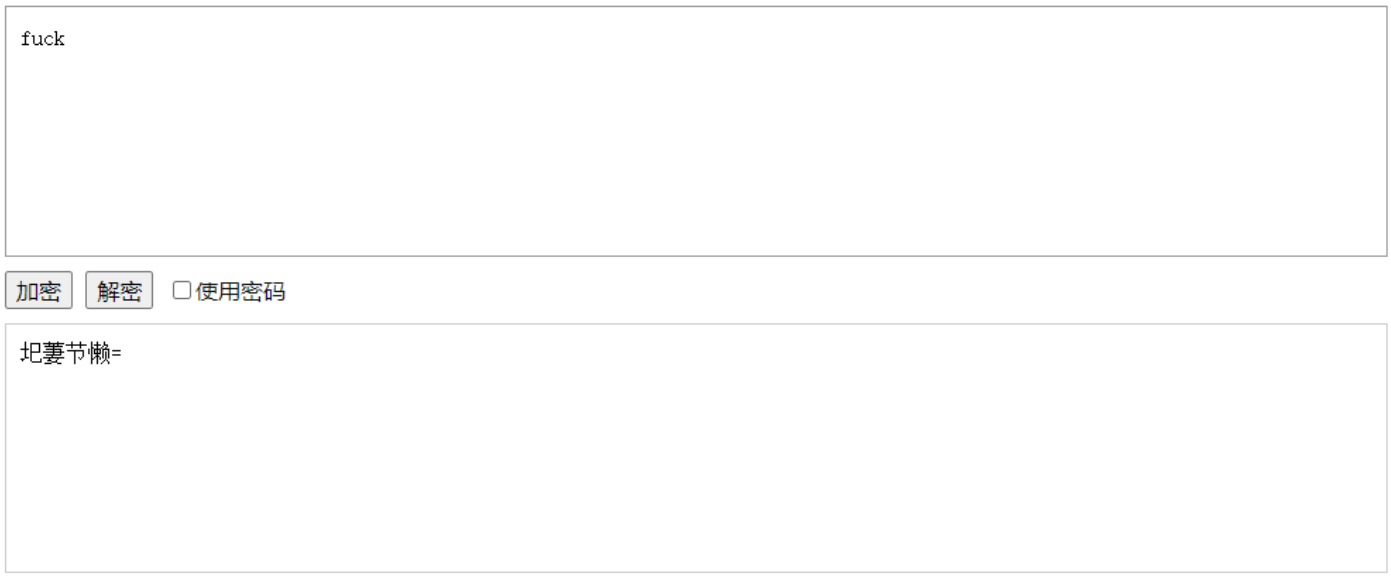
六、其他

1、与佛论禅

与佛论禅

2、文本加密

文本加密



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。

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

3、核心价值观密码

核心价值观密码

4、恩尼格码密码

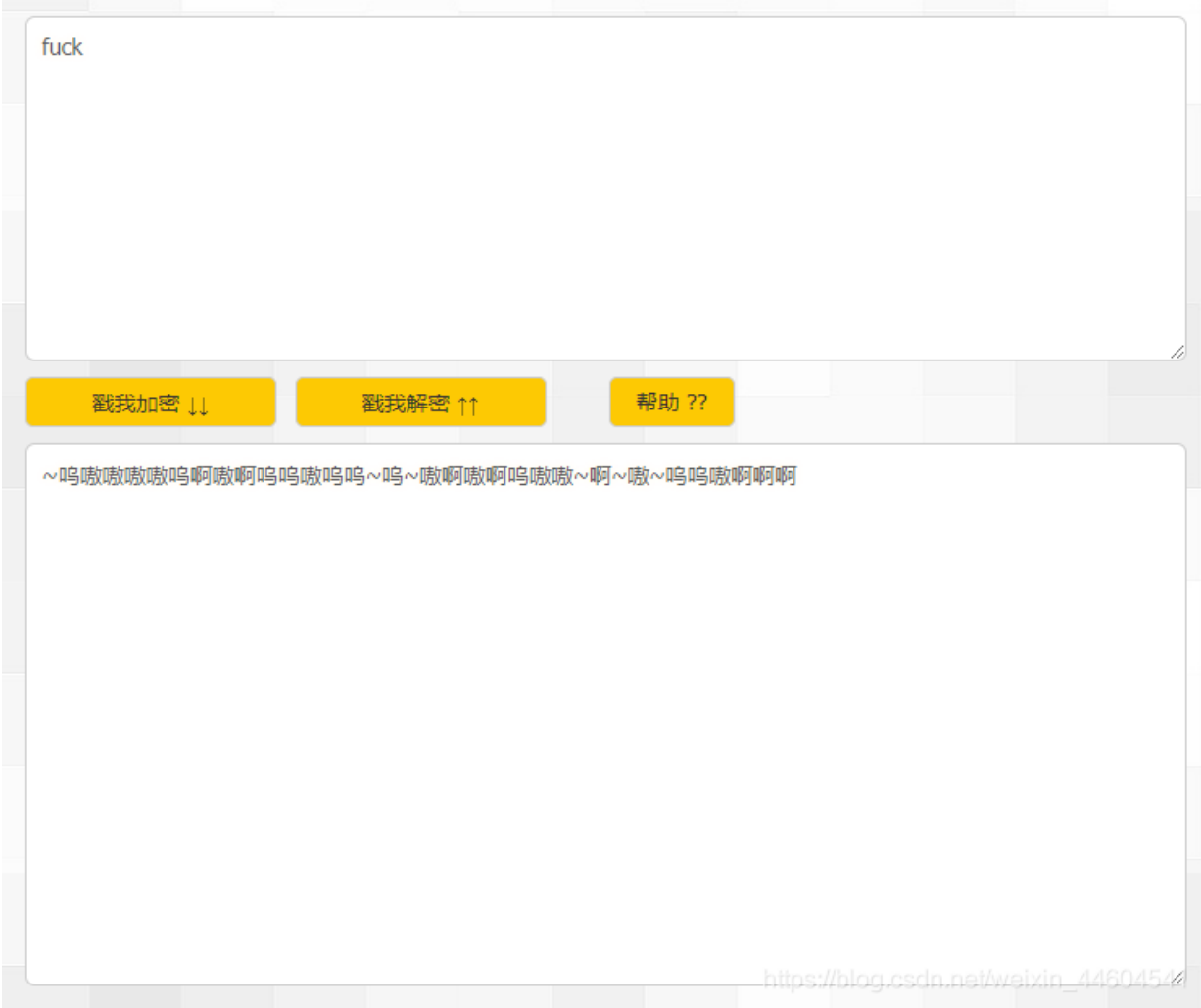
恩尼格玛密码机（德语：Enigma，又译哑谜机，或“谜”式密码机）是一种用于加密与解密文件的密码机。确切地说，恩尼格玛是对二战时期纳粹德国使用的一系列相似的转子机械加解密机器的统称，它包括了许多不同的型号，为密码学对称加密算法的流加密

模拟

5、兽音译者

给我笑死

兽音译者



结语

对常见的编码和密码做了个归纳

一些好用的网站和工具

- https://web2hack.org/xssee/
- json在线
- 程默的博客
- CaptfEncoder
- python_cryptanalysis
- Kryptos and Cryptanalysis Information
- Cipher Tools
- 在线加密解密

持续更新