**0x01 目录**

1. 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
2. 各种文本加密
3. 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)
4. 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. Bazeries密码(Bazeries 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)
5. 其他有趣的机械密码:
   1. 恩尼格玛密码
6. 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
7. 相关工具
8. 参考网站

**0x02 正文**

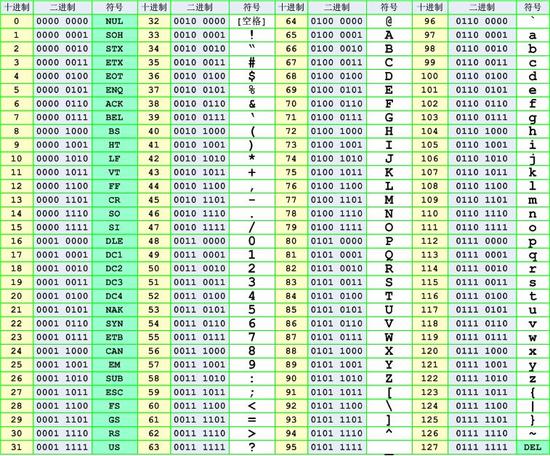
**常见编码**

**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;



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



ASCII编码对应十进制：

#!shell

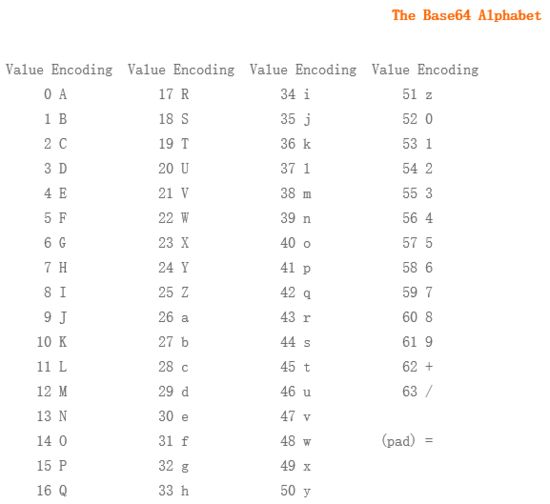
84 104 101 32 113 117 105 99 107 32 98 114 111 119 110 32 102 111 120 32 106 117 109 112 115 32 111 118 101 114 32 116 104 101 32 108 97 122 121 32 100 111 103

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位一个字节的形式，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：



举个例子(base64)：

源文本：T h e

对应ascii码:84 104 101

8位binary：01010100 01101000 01100101

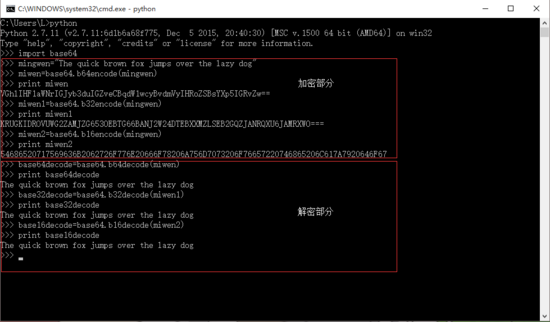
6位binary：010101 000110 100001 100101

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：



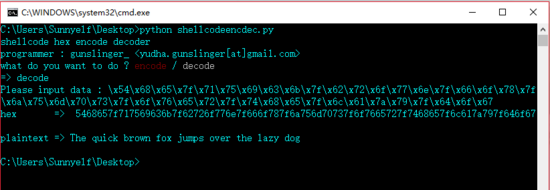
**3.shellcode编码**

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

编码后：

#!shell

\x54\x68\x65\x7f\x71\x75\x69\x63\x6b\x7f\x62\x72\x6f\x77\x6e\x7f\x66\x6f\x78\x7f\x6a\x75\x6d\x70\x73\x7f\x6f\x76\x65\x72\x7f\x74\x68\x65\x7f\x6c\x61\x7a\x79\x7f\x64\x6f\x67



**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) 。



源文本： 敏捷的棕色狐狸跳过了懒惰的狗

编码后：

#!shell

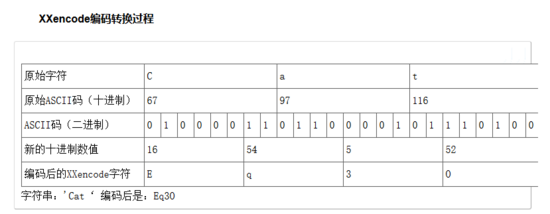
=E6=95=8F=E6=8D=B7=E7=9A=84=E6=A3=95=E8=89=B2=E7=8B=90=E7=8B=B8=E8=B7=B3=E8

=BF=87=E4=BA=86=E6=87=92=E6=83=B0=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。



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

编码后： hJ4VZ653pOKBf647mPrRi64NjS0-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(&1O9PH\*

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encodeURI、encodeURIComponent，PHP有 urlencode、urldecode等。编码方法很简单，在该字节ascii码的的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：

#!shell

%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)

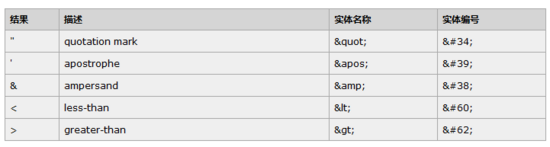
**9.Escape/Unescape编码**

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**



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，不同点是是用K字母被整合到C中。

敲击码表:

#!shell

1 2 3 4 5

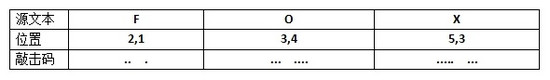
1 A B C/K D E

2 F G H I J

3 L M N O P

4 Q R S T U

5 V W X Y Z



**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：

#!shell

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:

#!shell

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

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=kTrscV5j5AsZq5zvBpr2jdkEJW8LqgrkkKsddwWA3YlXmgeqh_be95nMxqbFPOYoVBVy3A6lutlcXVDYLdZ-3iRawJpc0VZ71as07FnxtGS) 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



编码解码 [链接](http://www.qqxiuzi.cn/bianma/wenbenjiami.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： hqikrwfxupoeteayo

密文： Teucbonojmsvrhlzdghqikrwfxupoeteayo

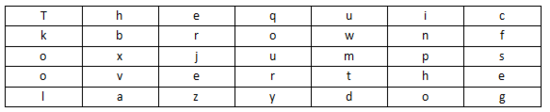
加解密 [传送门](http://www.practicalcryptography.com/ciphers/classical-era/rail-fence/)

**2.曲路密码**

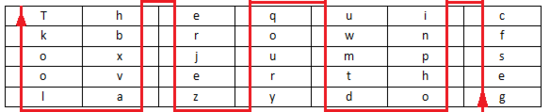
曲路密码(Curve Cipher)是一种换位密码，需要事先双方约定密钥(也就是曲路路径)。

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)



加密的回路线(事先约定填充的行列数)



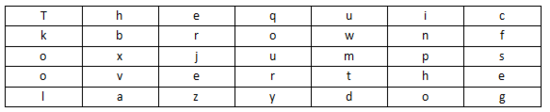
密文： gesfc inpho dtmwu qoury zejre hbxva lookT

**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)



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：



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/amsco-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**

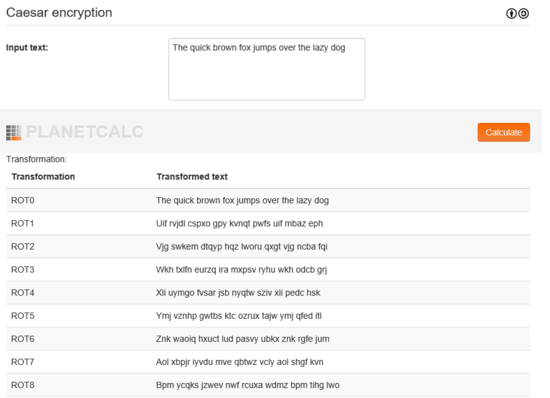
凯撒密码(Caesar Cipher或称恺撒加密、恺撒变换、变换加密、位移加密)是一种替换加密，明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。例，当偏移量是3的时候，所有的字母A将被替换成D，B变成E，以此类推，更多 [参考](https://en.wikipedia.org/wiki/Caesar_cipher) 。

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



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。

#!python

>>> from pycipher import Caesar

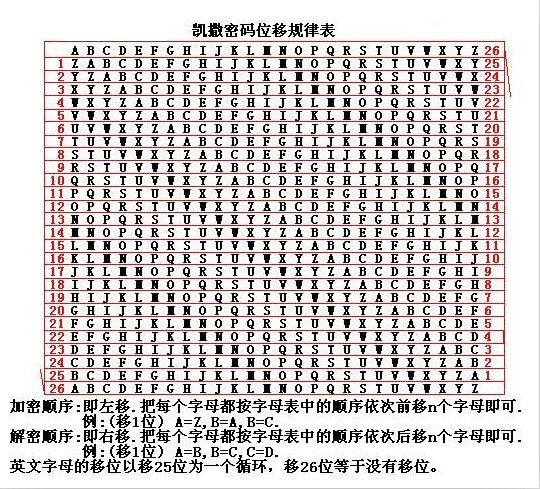
>>> Caesar(key=1).encipher('The quick brown fox jumps over the lazy dog')

'UIFRVJDLCSPXOGPYKVNQTPWFSUIFMBAZEPH'

>>> Caesar(key=1).decipher('UIFRVJDLCSPXOGPYKVNQTPWFSUIFMBAZEPH')

'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'

参考表(这里是向后移位加密，向前移位解密)：



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以例。

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：

#!shell

明文字母 : abcdefghijklmnopqrstuvwxyz

明文字母 : phqgiumeaylnofdxjkrcvstzwb

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**

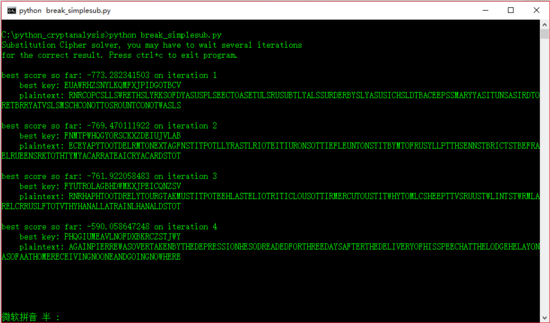
当密文数据足够多时这种密码我们可以通过字频分析方法破解或其他方法破解，比较好的在线词频分析网站 <http://quipqiup.com/index.php> (翻= =墙)，这里推荐一篇通过"爬山算法"来破解简单替换密码 [文章](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-simple-substitution-cipher/) ，基于文中的算法实现的工具来破解示例。

密文：

#!shell

pmpafxaikkitprdsikcplifhwceigixkirradfeirdgkipgigudkcekiigpwrpucikceiginasikwduearrxiiqepcceindgmieinpwdfprduppcedoikiqiasafmfddfipfgmdafmfdteiki

解密：



(ps:score值越小越准确)

密钥： PHQGIUMEAVLNOFDXBKRCZSTJWY

明文：

#!shell

AGAINPIERREWASOVERTAKENBYTHEDEPRESSIONHESODREADEDFORTHREEDAYSAFTERTHEDELIVERYOFHISSPEECHATTHELODGEHELAYONASOFAATHOMERECEIVINGNOONEANDGOINGNOWHERE

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

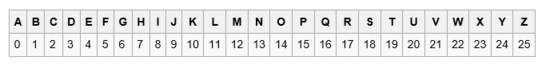
**5.希尔密码**

**（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×n的矩阵相乘，再将得出的结果MOD26。更多 [参考](https://en.wikipedia.org/wiki/Hill_cipher)

**（2）加密**

明文： ACT

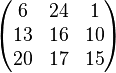


明文对应矩阵：

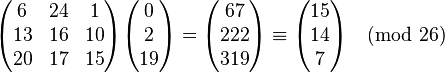


加密密钥： GYBNQKURP

加密矩阵：



计算过程：

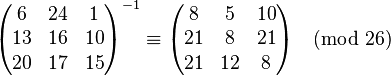


密文： FIN

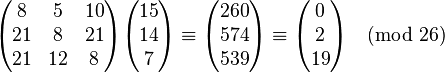
**（3）解密**

密文： FIN

计算加密矩阵的逆矩阵：



解密计算：



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-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：

#!shell

XUKEXWSLZJUAXUNKIGWFSOZRAWURORKXAOSLHROBXBTKCMUWDVPTFBLMKEFVWMUXTVTWUIDDJVZKBRMCWOIWYDXMLUFPVSHAGSVWUFWORCWUIDUJCNVTTBERTUNOJUZHVTWKORSVRZSVVFSQXOCMUWPYTRLGBMCYPOJCLRIYTVFCCMUWUFPOXCNMCIWMSKPXEDLYIQKDJWIWCJUMVRCJUMVRKXWURKPSEEIWZVXULEIOETOOFWKBIUXPXUGOWLFPWUSCH

解密：

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

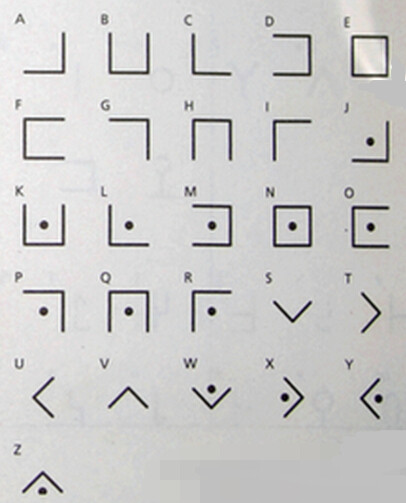
在线加解密 [传送门](http://www.practicalcryptography.com/ciphers/hill-cipher/)

**6.猪圈密码**

**（1）介绍**

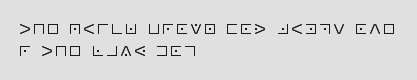
猪圈密码(Pigpen Cipher或称九宫格密码、朱高密码、共济会密码或共济会员密码)，是一种以格子为基础的简单替代式密码。更多 [参考](https://en.wikipedia.org/wiki/Pigpen_cipher)

明文字母和对应密文：



明文： THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

密文：

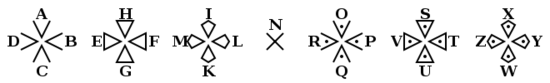


在线加密 [传送门](http://www.simonsingh.net/The_Black_Chamber/pigpen.html)

**（2）变种**

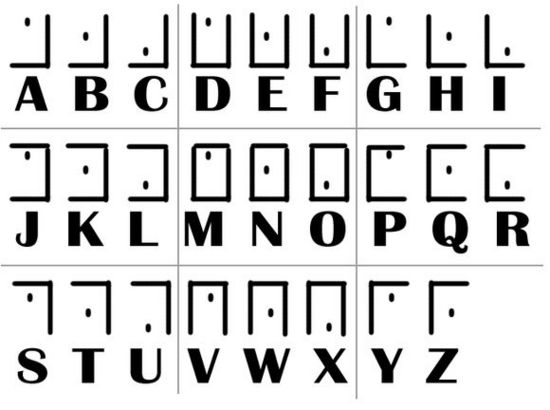
圣堂武士密码(Templar Cipher)是共济会的“猪圈密码”的一个变种，一直被共济会圣殿骑士用。

明文字母和对应密文：

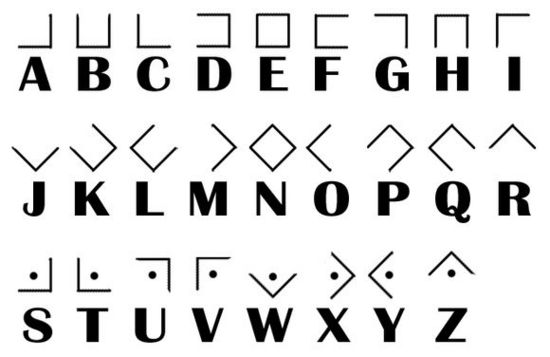


**（3）其他变种**

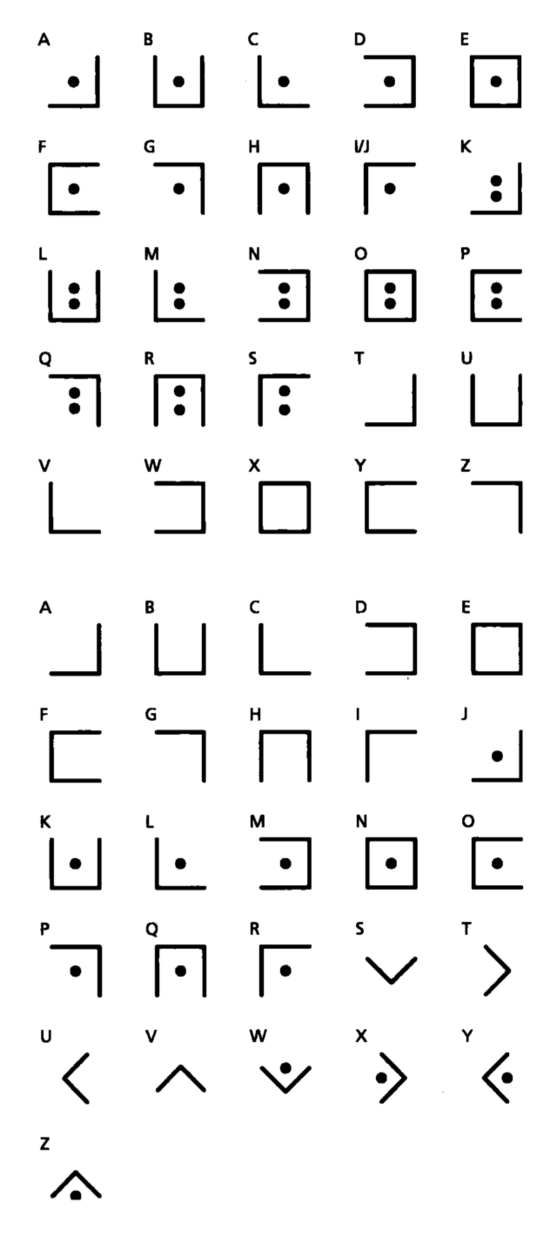
明文字母和对应密文：



明文字母和对应密文：



明文字母和对应密文：

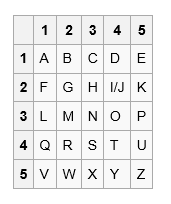


**7.波利比奥斯方阵密码**

**（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：



加密实例：

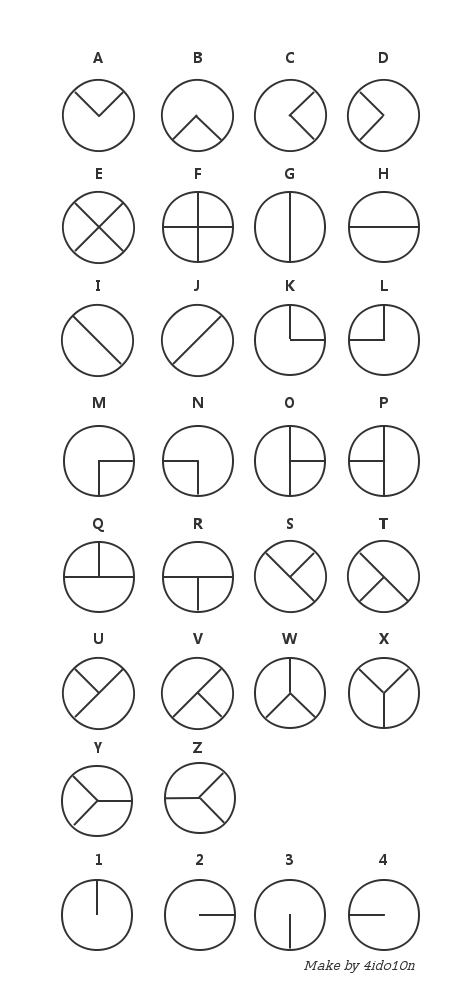
明文： THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

密文： 442315 4145241325 1242345233 213453 2445323543 442315 31115554 143422

**8.夏多密码(曲折加密)**

**（1）介绍**

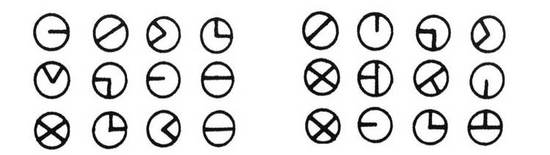
夏多密码是作者麦克斯韦·格兰特在中篇小说《死亡之链》塑造夏多这一英雄人物中所自创的密码，如下图所示：



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：



**9.普莱菲尔密码**

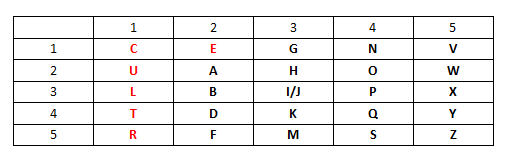
普莱菲尔密码(Playfair Cipher)是第一种用于实际的双字替换密码，用双字加密取代了简单代换密码的单字加密，很明显这样使得密文更难破译，因为使用简单替换密码的频率分析基本没有什么作用，虽然频率分析，通常仍然可以进行，但是有25×25=625种可能而不是25种可能，可以分为三个步骤，即编制密码表、整理明文、编写译文，下面我们以明文：

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:



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 "：

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" EG "；

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 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，得到密文 KUNDL HGTLF WUESP WLHSI NPCGC 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)：

#!python

>>>from pycipher import Playfair

>>>Playfair('CULTREABDFGHIKMNOPQSVWXYZ').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')

'UKDNLHTGFLWUSEPWHLISNPCGCRGAUBVZAQIV'

>>>Playfair('CULTREABDFGHIKMNOPQSVWXYZ').decipher('UKDNLHTGFLWUSEPWHLISNPCGCRGAUBVZAQIV')

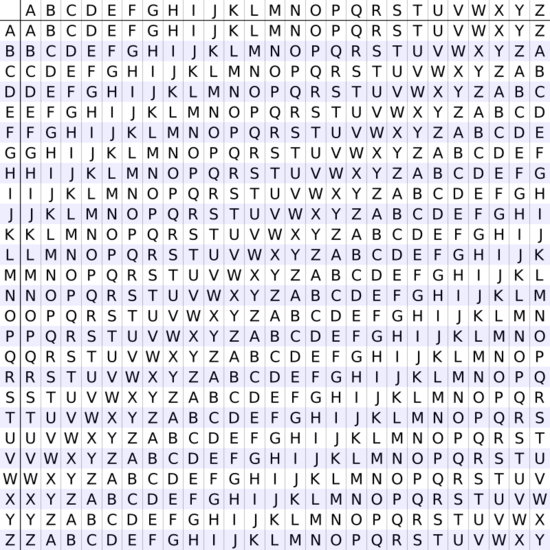
'THEQUICKBROWNFOXIUMPSOVERTHELAZYDOGX'

**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：



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**

#!python

>>>from pycipher import Vigenere

>>>Vigenere('CULTURE').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')

'VBPJOZGMVCHQEJQRUNGGWQPPKNYINUKRXFK'

>>>Vigenere('CULTURE').decipher('VBPJOZGMVCHQEJQRUNGGWQPPKNYINUKRXFK')

'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'

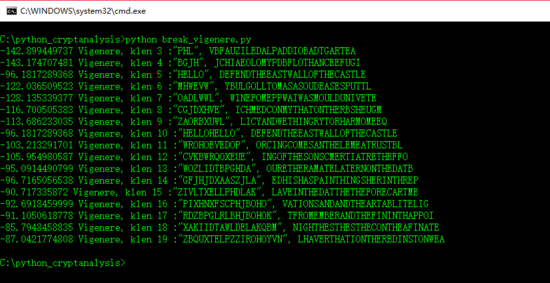
在线加密解密 [传送门](http://planetcalc.com/2468/)

**（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：

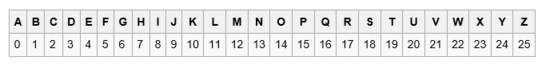


(ps:结合左边的值，密钥以及解出明文可以确定kien 5或者klen 10为准确的结果)

明文： DEFEND THE EAST WALL OF THE CASTLE

**（4）变种**

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.html) (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。



#!python

>>>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'

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 FWW ZVDP WVK

**（2）已知关键词加解密**

#!python

>>>from pycipher import Autokey

>>>Autokey('CULTURE').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')

'VBPJOZGDIVEQVHYYAIICXCSNLFWWZVDPWVK'

>>>Autokey('CULTURE').decipher('VBPJOZGDIVEQVHYYAIICXCSNLFWWZVDPWVK')

'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**

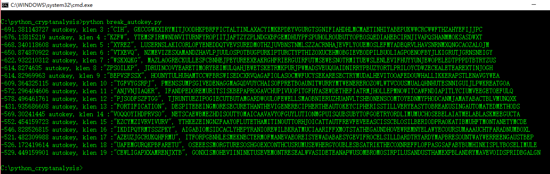
推荐去看这篇 [自动密钥密码分析文章](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-autokey-cipher/) ，基于文中的算法实现的工具来破解示例。

密文:

#!shell

isjiqymdebvuzrvwhmvysibugzhyinmiyeiklcvioimbninyksmmnjmgalvimlhspjxmgfiraqlhjcpvolqmnyynhpdetoxemgnoxl

解密



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

明文：

#!shell

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**

#!python

>>>from pycipher import Beaufort

>>>Beaufort('CULTURE').encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')

'JNHDAJCSTUFYEZOXCZICMOZHCBKARUMVRDY'

>>>Beaufort('CULTURE').decipher('JNHDAJCSTUFYEZOXCZICMOZHCBKARUMVRDY')

'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 OGTC 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：

#!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

已知密钥在线加解密 [传送门](http://www.practicalcryptography.com/ciphers/classical-era/porta/)

**（2）破解**

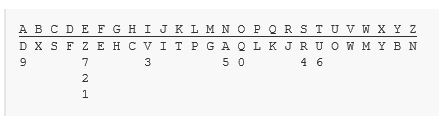
Porta密码可以被以 [维吉尼亚密码](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher-part-2/) 破解相类似方式进行自动攻破，破解Porta密码第一步是先确定密钥长度，这里推荐一篇关于使用 [重合指数算法](https://en.wikipedia.org/wiki/Index_of_coincidence) 确定为维吉尼亚密钥长度 [文章](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher/) 。

**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：



明文: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

密文(其中一种)： 6CZ KOVST XJ0MA EQY IOGL4 0W1J UC7 P9NB F0H

**（2）破解**

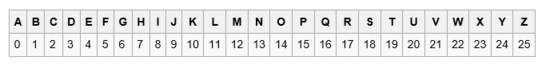
如果同音替换密码的同音词个数很多，那么破解它难度很大，通常的方法采取类似破解替换密码的"爬山算法"，除了找到一个明文字母映射几个字符之外，我们还需要确定映射了那些字符，可以尝试 [2层嵌套"爬山算法"](http://www.cs.sjsu.edu/faculty/stamp/RUA/homophonic.pdf) 来破解，外层确定映射的数量，内层确定映射字符。

**16.仿射密码**

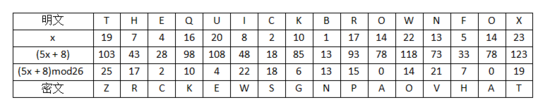
**（1）介绍**

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。更多 [参考](https://en.wikipedia.org/wiki/Affine_cipher)

常见的字母对应关系：



下面我们以E(x) = (5x + 8) mod 26函数为例子



至于解密我们知道



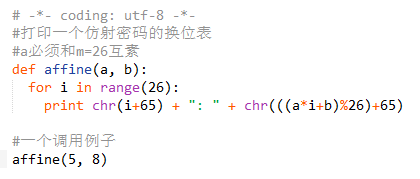


即可得出解密结果



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



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)：

#!shell

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：

#!shell

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

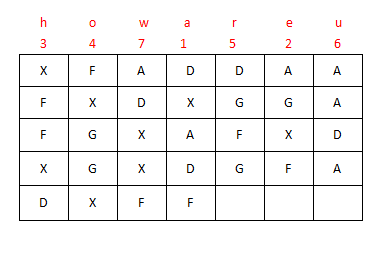
明文： THE QUICK BROWN FOX

结果矩阵加密：

#!shell

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

列移位密钥： how are u



密文： DXADF AGXF XFFXD FXGGX DGFG AADA ADXXF

已知密钥加解密：

#!python

>>>from pycipher import ADFGX

>>>a = ADFGX('phqgmeaynofdxkrcvszwbutil','HOWAREU')

>>>a.encipher('THE QUICK BROWN FOX')

'DXADFAGXFXFFXDFXGGXDGFGAADAADXXF'

>>>a.decipher('DXADFAGXFXFFXDFXGGXDGFGAADAADXXF')

'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。

#!shell

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

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。

#!python

>>>from pycipher import ADFGVX

>>>a = ADFGVX('ph0qg64mea1yl2nofdxkr3cvs5zw7bj9uti8','HOWAREU')

>>>a.encipher('THE QUICK BROWN FOX')

'DXXFAFGFFXGGGFGXDVGDVGFAVFVAFVGG'

>>>a.decipher('DXXFAFGFFXGGGFGXDVGDVGFAVFVAFVGG')

'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:

#!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

**（2）已知密阵加解密**

#!python

>>>from pycipher import

>>>Bifid('phqgmeaylnofdxkrcvszwbuti',5).encipher('THE QUICK BROWN FOX')

'WETEDTKZNEKYOMEX'

>>>Bifid('phqgmeaylnofdxkrcvszwbuti',5).decipher('WETEDTKZNEKYOMEX')

'THEQUICKBROWNFOX'

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

**（3）未知密阵破解**

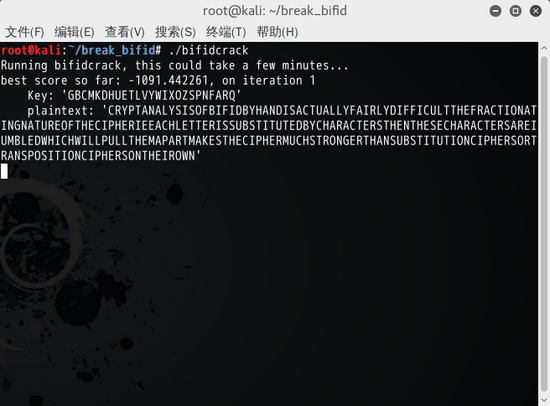
手工分析破解双密码是有一定难度的，每个字母都是同过3个数字进行非线性代替转换，而且之后还会对字母顺序进行打乱，这样使双密码比一些替换密码和换位密码更难破解。然而，现在是计算机时代，这张加密方式没有安全性可言，通过 [模拟退火](http://baike.baidu.com/link?url=mkceUr0W4L7B7UVQxc-dUkXKPJbj9v4YyBh_hrskt5iXk99UdnjW6mZ_YxoJO1PkT1zdjEZD2hd7TCMiSxpOma) 算法就能快速找到双密码的密阵。 这里推荐一篇详细的 [双密码破解分析](http://www.practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-bifid-cipher/) 的文章，基于模拟退火算法实现的工具破解示例：

密文:

#!shell

KWTAZQLAWWZCPONIVBTTBVQUZUGRNHAYIYGIAAYURCUQLDFTYVHTNQEENUPAIFCUNQTNGITEFUSHFDWHRIFSVTBISYDHHASQSROMUEVPQHHCCRBYTQBHWYRRHTEPEKHOBFSZUQBTSYRSQUDCSAOVUUGXOAUYWHPGAYHDNKEZPFKKWRIEHDWPEIOTBKESYETPBPOGTHQSPUMDOVUEQAUPCPFCQHRPHSOPQRSSLPEVWNIQDIOTSQESDHURIEREN

解密：



得到加密矩阵:

#!shell

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:

#!shell

CRYPTANALYS IS 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 CIPHER SON 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3×3×3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.

经过密阵转换：

#!shell

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):

#!shell

THEQU ICKBR OWNFO X.

23131 31223 31233 22

33132 22232 13311 21

31132 23221 32113 32

合并：

#!shell

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

在经过密阵转换后密文:

#!shell

231313313231132312232223223221312331331132113222132

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) 。

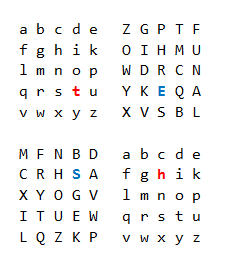
**21.四方密码**

**（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：



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**

#!python

>>>from pycipher import Foursquare

>>>fs = Foursquare('zgptfoihmuwdrcnykeqaxvsbl','mfnbdcrhsaxyogvituewlqzkp')

>>>fs.encipher('THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG')

'ESZWQAFHGTDKWHRKUENYQOLMQTUNWMBPTGHQ'

>>>fs.decipher('ESZWQAFHGTDKWHRKUENYQOLMQTUNWMBPTGHQ')

'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-searching/cryptanalysis-foursquare-cipher/) 文章，如果有足够多的密文那么四方密码可以轻易被破解，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-searching/cryptanalysis-foursquare-cipher/) 破解示例：

密文(密文最好在200个字符以上)：

HMMKEQESDTMDHLAWFWMNKSOSFOMRFNLWLKHNSQGGEKXEOLLVDXNRSQQGARTFKSAVNUDLFNDHESPZGQ TWESAGPGSQSQSTPKUSBBQLQHESAGPGSQSQGXLNAVHTPMHMKKNYGSUGDMTPDGFNKYAVHXLWGEKRILESLZ ZOFNAVIHRHRKAGHSMYUGEGNSRGAVMVOQPRLNKRXLMYLQPXILESQYBNRHRKAGKYQXDIHMPGPYOERZOLBEZ LURFWLWUOLDDPNSQYAGMUQPQWESBEZVEQESDTMDBQLWDIUSHB

用法：

#!shell

gcc -O3 -lm foursquarecrack2.c scoreText\_2.c -o fsc

./fsc

输出结果：

#!shell

Running foursquarecrack, this could take a few minutes...

best score so far: -1239.505249, on iteration 1

Key: 'KFMLUGWSQEPOZTNRBHDAVXCIY','UGSVKFIZMOYXPQRWTHLNCABED'

plaintext: 'THECIPHERTEXTSQUARESCANBEGENERATEDUSINGAKEYWORDDROPPINGDUPLICAT

ELETTERSTHENFILLTHEREMAININGSPACESWITHTHEREMAININGLETTERSOFTHEA

LPHABETINORDERALTERNATIVELYTHECIPHERTEXTSQUARESCANBEGENERATEDCO

MPLETELYRANDOMLYTHEFOURSQUAREALGORITHMALLOWSFORTWOSEPARATEKEYSO

NEFOREACHOFTHETWOCIPHERTEXTMATRICESX'

**2.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：

#!shell

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:

#!shell

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):

#!shell

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

#!shell

8372983729837298372983729837

+7230934713241313536757403677

-----------------------------

5502817432078501808630122404

在经过棋盘转换后:

#!shell

5502817432078501808630122404

ppfmyk n if pfkyfyd hkmmcfc

最终得到密文: ppfmyk n if pfkyfyd hkmmcfc

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:

#!shell

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。

明文： THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

(类似)摩尔斯电码:

#!shell

-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)

**25.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:

#!shell

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:

#!shell

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

使用密钥矩阵替换:

#!shell

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：

#!shell

1 2 3 4 5 6 7 8 9

D I G R A F D 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

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

明文: THE QUICK BROWN FOX

密表转换(以4分组为例):

#!shell

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:

#!shell

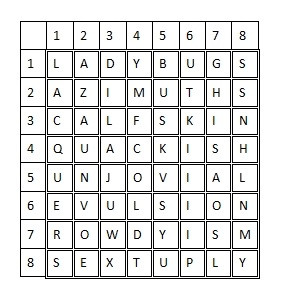
213 975 724 924 876 716 566 516

Ip #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:



#!shell

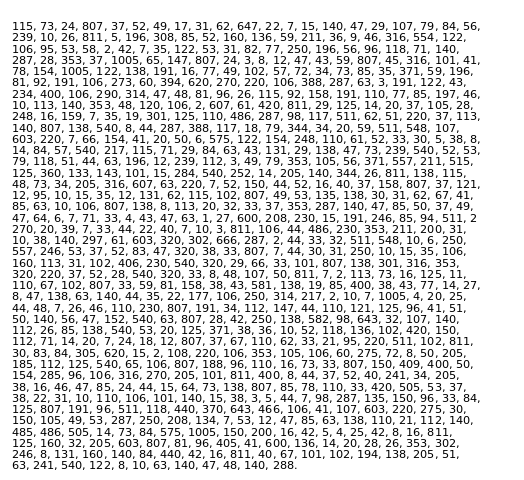
明文:T H E Q U I C K B R O W N F O

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%A0%81) 。

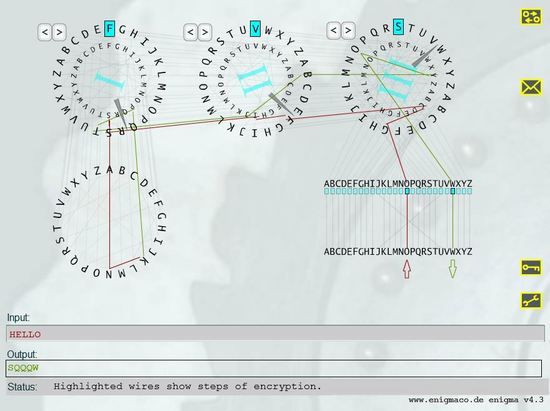
**29.键盘密码**

一般用到的键盘密码就是手机键盘和电脑键盘两种，2014 0ctf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/0ops-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%A0%BC%E7%8E%9B%E5%AF%86%E7%A0%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](http://www.zhaoyuanma.com/aspfix.html)

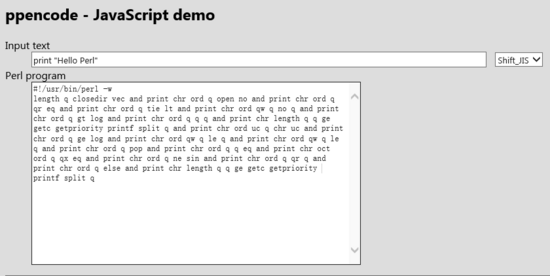
2.[php language="混淆加密"][/php] [118](http://www.zhaoyuanma.com/phpjmvip.html)

3.[css language="/js混淆加密"][/css] [119](http://tool.css-js.com/)

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

**5.ppencode**

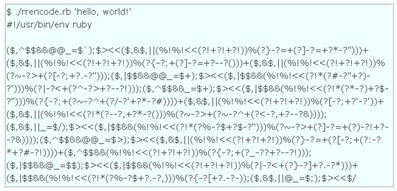
ppencode-Perl把Perl代码转换成只有英文字母的字符串。



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

**6.rrencode**

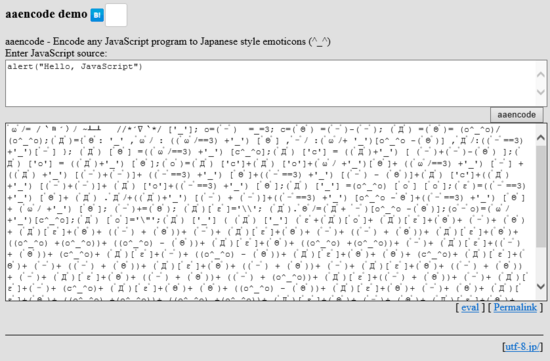
rrencode可以把ruby代码全部转换成符号。



rrencode [传送门](http://www.lab2.kuis.kyoto-u.ac.jp/~yyoshida/rrencode.html)

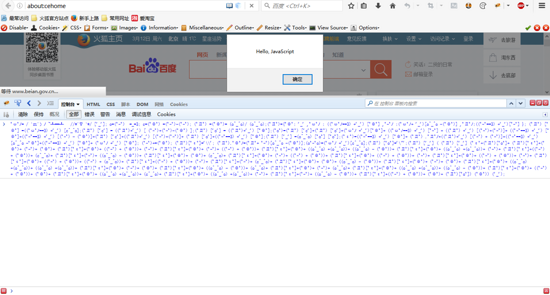
**7.jjencode/aaencode**

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。



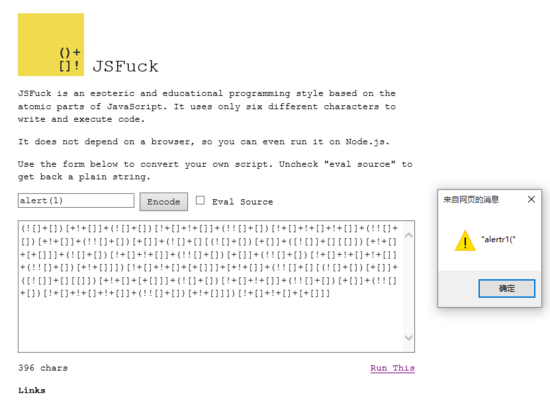
aaencode [传送门](http://utf-8.jp/public/aaencode.html)

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。



**8.JSfuck**

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

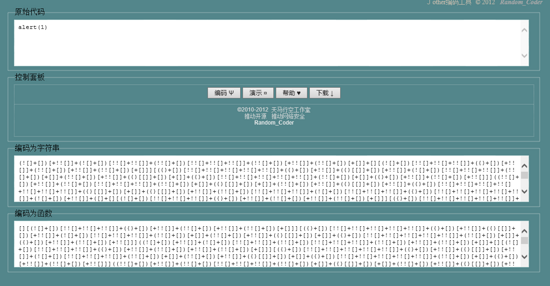


JSfuck [传送门](http://www.jsfuck.com/)

**9.jother**

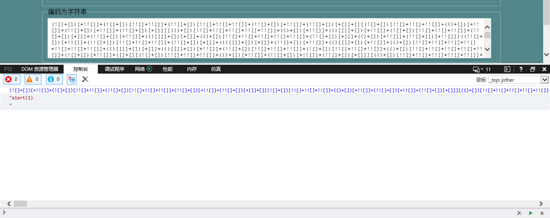
jother是一种运用于javascript语言中利用少量字符构造精简的匿名函数方法对于字符串进行的编码方式。其中8个少量字符包括： ! + ( ) [ ] { } 。只用这些字符就能完成对任意字符串的编码。

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



jother编码 [传送门](http://tmxk.org/jother/)

jother直接在浏览器(IE可以)的控制台里输入密文即可执行解密：



**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!

#!shell

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

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

----- --<]> ----- ----- .<

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)