

密码学解密网站汇总

md5解密;<https://www.somd5.com/>

DES加密

<http://tool.chacuo.net/cryptdes>

程默的博客IP域名便民

加密解密工具

在线

对称性加密解密

» DES加密解密

» 3DES加密解密

» AES加密解密

» chacha20加解密

» RC2加密解密

» RC4加密解密

» RC5加密解密

» RC6加密解密

» Blowfish加密解密

» Twofish加密解密

» Serpent加密解密

» Gost加密解密

» Rijndael加密解密

» Cast加密解密

» Xtea加密解密

非对称性加密解密

» rsa公钥加密解密

» rsa私钥加密解密

» RSA密钥对

» RSA私钥密码清除

» RSA私钥密码修改

» PKCS#1转PKCS8

» 校验RSA密钥对

» 私钥中提取公钥

» Rsa公私钥解析

» DSA密钥对

广告 X

16进制解码

<http://www.cyeam.com/tool/hexdecode>

🔗 格式化 json

🔗 json 转 go 结构体

🔗 xml 转 go 结构体

🔗 SQL Create语句生成 go 结构体

🔗 JSON 转 Thrift

🔗 Url Encode 编码

🔗 Url Decode 解码

🔗 Base 32 编码

🔗 Base 32 解码

🔗 Base 64 编码

🔗 Base 64 解码

🔗 Hex 十六进制编码

🔗 Hex 十六进制解码

🔗 格式化 xml

🔗 MessagePack 转 JSON

🔗 字符串压缩

声网Agora

Agora快速实现超低延时互动直播

广告

73646E6973635F32303138

打开

sdnisc_2018

0x00.综合

网站中包含大多编码的解码。
<http://web2hack.org/xssee/>
<https://www.sojson.com/>
<http://web.chacuo.net/>

python的反编译
<https://tool.lu/pyc/>

0x01.文字倒序排列

<http://www.qqxiuzi.cn/zh/daoxu/>

0x02.cmd5解密

<http://www.cmd5.com/>
<http://pmd5.com/>
<http://tool.chinaz.com/Tools/MD5.aspx>
<http://md5jjami.51240.com/>

0x03.

①base64解密

<http://tool.chinaz.com/Tools/Base64.aspx>
<http://www1.tc711.com/tool/BASE64.htm>
<http://base64.supfree.net/>
<http://web.chacuo.net/charsetbase64>

②base32解密

http://tomeko.net/online_tools/base32.php?lang=en

0x04.ROT5、ROT13、ROT18、ROT47位移编码

<http://www.qqxiuzi.cn/bianma/ROT5-13-18-47.php>

0x05.颜文字 (aadecode)

打开谷歌浏览器->F12->点击Console->粘贴复制回车就会出现答案

<https://cat-in-136.github.io/2010/12/aadecode-decode-encoded-as-aaencode.html>

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

0x06.邮件编码 (UUencode: M=75E;F-O9&7HOYGDN*KED)

<http://web.chacuo.net/charsetuuencode>

<http://www.mxcz.net/tools/UUEncode.aspx>

0x07.查尔斯加密 (playfair解密)

<http://rumkin.com/tools/cipher/playfair.php>

aHR0cDovL3d3dy56aG9uZ2d1b3NvdS5jb20vem9uZ2hlL21vZXJzaWNvZGVjb252ZXJ0ZXluYXNweA

0x08.在线编码解码

<http://bianma.911cha.com/>

0x09.进制间转换

<http://tool.oschina.net/hexconvert/>

0x010.16进与文本转换

<http://www.5ixuexiwan.com/str/from-hex.php>

<http://www.5ixuexiwan.com/str/hex.php>

0x011.Unicode编码转换 (4种)

&#x [Hex]: The

&# [Decimal]: The

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

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

<http://tool.oschina.net/encode?type=3> (\u5927)

<http://tool.chinaz.com/tools/unicode.aspx> (\u5927)

<http://tool.oschina.net/encode> (安)

0xc12.Brainfuck (++++++>++++>)

<https://www.splitbrain.org/services/ook> (Ook!与Brainfuck)

<https://www.nayuki.io/page/brainfuck-interpreter-javascript>

<http://esoteric.sange.fi/brainfuck/impl/interp/i.html>

0x013.URL (%23)

<http://tool.chinaz.com/tools/urlencode.aspx>

<http://web.chacuo.net/charseturlencode>

<http://tool.oschina.net/encode?type=4>

0x014.二维码生成/解码器

<http://tool.chinaz.com/qrcode/>

<http://jiema.wwei.cn/>

0x015.摩斯密码

<http://www.zou114.com/mesm/>

<http://www.jb51.net/tools/morse.htm>

<http://rumkin.com/tools/cipher/morse.php>

0x016.quipqiup

<https://www.xarg.org/tools/caesar-cipher/>

0x017.与佛论禅

<http://www.keyfc.net/bbs/tools/tudoucode.aspx>

0x018.xxencode(Ri64NjS0-eRKpkQm-jRaJm6)

<http://web.chacuo.net/charsetxxencode>

0x019.jsfuck ([])(![]+[])[+[]]

注：用！+加密

打开谷歌浏览器->F12->点击Console->粘贴复制回车就会出现答案

<http://discogscounter.getfreehosting.co.uk/js-noalnum.php?ckattempt=1&i=1>（放在下面的大框里，点击eval(-v))

0x020.UTF-8编码(安)

<http://tool.chinaz.com/Tools/UTF-8.aspx>

<http://tool.oschina.net/encode?type=2>

<http://www.3464.com/Tools/HtmlToOther/>

0x021.DES(要密钥)

<http://encode.chahuo.com/>

<http://tool.oschina.net/encrypt/>

0x022.凯撒

凯撒密码最早由古罗马军事统帅盖乌斯·尤利乌斯·凯撒在军队中用来传递加密信息，故称凯撒密码。这是一种位移加密方式，只对26个字母进行位移替换加密，规则简单，容易破解。下面是位移1次的对比：

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

将明文字母表向后移动1位，A变成了B，B变成了C.....，Z变成了A。同理，若将明文字母表向后移动3位：

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

则A变成了D，B变成了E.....，Z变成了C。

字母表最多可以移动25位。凯撒密码的明文字母表向后或向前移动都是可以的，通常表述为向后移动，如果要向前移动1位，则等同于向后移动25位，位移选择为25即可。

<http://www.zjslove.com/3.decode/kaisa/index.html>

<http://planetcalc.com/1434/>

<https://www.qqxiuzi.cn/bianma/kaisamima.php>

<http://www.metools.info/code/c70.html>

0x023.Rabbit（兔子）解密

注：此编码与和base64很相似

<http://tool.chinaz.com/Tools/TextEncrypt.aspx>

<http://tool.oschina.net/encrypt/>

0x024.AES

部分例子：U2FsdGVkX1+qtU8KEGmMjwGgKcPUK3XBTdM+KhNRLHSCQL2nSXaW8++yBUkSylRp

<http://tool.oschina.net/encrypt/>

<http://aes.online-domain-tools.com/>

0x025.JS解密

<http://www.haokuwang.com/jsdecode.htm>

<http://www.dheart.net/decode/index.php>

0x026.ASP解密

部分例子：#@EQAAAA==VXlj4UmkaYAUmKN3bAYAAA==#@

<http://adophper.com/encode.html>

0x027.Snake解密（要密钥）

<http://serpent.online-domain-tools.com/>

0x028.GRB与十六进制的转换

<http://www.sioe.cn/yingyong/yanse-rgb-16/>

<http://www.atool.org/colorpicker.php>

0x029.html隐写在线解密

注：密钥和网址

<http://fog.misty.com/perry/ccs/snow/snow/snow.html>

0x030.shellcode编码 (\x54\x68\x65\x7f)

<http://www.jb51.net/article/10399.htm>

0x031.Quoted-printable编码

我们收邮件，查看信件原始信息，经常会看到这种类型的编码！（=E6=95=8F=E6=8D=B7=E7=9A）

```
function quoted_printable_encode($string) {  
    return preg_replace('/[\\r\\n]{73}=[\\r\\n]{2}/', "KaTeX parse error: Can't use function '\\r' in math mode at position 3: 0=\\r\\n", str_replac...string));  
}
```

0x032.Escape/Unescape编码 (%u0054%u0068%u0065)

<https://escape.supfree.net/>

0x032.敲击码(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
```

0x033.加密为汉字

<http://www.qqxiuzi.cn/bianma/wenbenjiami.php>

0x034.栅栏

<http://www.practicalcryptography.com/ciphers/classical-era/rail-fence/>

<http://www.atoolbox.net/Tool.php?Id=777>

0x035.曲路密码

需要事先双方约定密钥(也就是曲路路径)

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

0x036.列移位密码

h 3	o 4	w 7	a 1	r 5	e 2	u 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

https://blog.csdn.net/qq_41638851

密钥: how are u

密文: qoury inpho Tkool hbxva uwmt d cfseg erjez

0x037.埃特巴什码

ABCDEFGHIJKLMNOPQRSTUVWXYZ ZYXWVUTSRQPONMLKJIHGFEDCBA

<http://www.practicalcryptography.com/ciphers/classical-era/atbash-cipher/>

0x038.希尔密码 (利用矩阵)

<http://www.practicalcryptography.com/ciphers/hill-cipher/>

0x039.夏多密码(曲折加密)

技术分享图片

0x040.编制密码表

注：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"。

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

	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

https://blog.csdn.net/qq_41638851

<http://www.practicalcryptography.com/ciphers/classical-era/playfair/>

0x041.维吉尼亚密码

<http://planetcalc.com/2468/>（已知密匙）

<http://rumkin.com/tools/cipher/gronsfeld.php>（未知密匙）

<http://www.zjslove.com/3.decode/weijiniya/index.html>

0x042.自动密钥密码

<http://www.practicalcryptography.com/ciphers/classical-era/autokey/>

0x043.博福特密码

<http://www.practicalcryptography.com/ciphers/classical-era/beaufort/>

0x044.滚动密钥密码

<http://www.practicalcryptography.com/ciphers/classical-era/running-key/>

0x045.Porta密码

<http://www.practicalcryptography.com/ciphers/classical-era/porta/>

0x046.仿射密码

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

<http://www.practicalcryptography.com/ciphers/classical-era/affine/>

0x047.培根密码

<http://rumkin.com/tools/cipher/baconian.php>

<https://netair.xyz/tools/>培根密码加密解密.html

0x048.ADFGX密码

<http://www.practicalcryptography.com/ciphers/adfgx-cipher/>

0x049.双密码

明文: THE QUICK BROWN FOX=>5121542133 5435452521 3523311521 34 =>密文: WETED TKZNE
KYOME X

<http://www.practicalcryptography.com/ciphers/classical-era/bifid/>

0x050.ppencode

把Perl代码转换成只有英文字母的字符串

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

0x051.rrencode(把ruby代码全部转换成符号)

0x052.jother

可以在浏览器(IE可以)的控制台里输入密文即可执行解密 (用! + () [] { } 加密)

<http://tmxk.org/jother/> (编码)

0x053.CR4解密

<http://tool.oschina.net/encrypt>

<http://tool.chinaz.com/Tools/TextEncrypt.aspx>

0x054.VBScript

部分例子: #@TgAAAA=='[6*liLa6++p'aXvfiLaa6i[[avWi[[a*p[[[6*!!'[6cp'aXvXILa6fp[:6+Wp[:XvWi[[[6+XivRIAAA==#@

<http://www.dheart.net/decode/index.php>

0x055.执行php文件

<https://www.shucunwang.com/RunCode/php/>

0x056.图片/BASE64转换

<http://tool.oschina.net/encrypt?type=4>

0x057.QWE解密

进行QWE解密

【QWE加密表】

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

https://blog.csdn.net/qq_41638851

0x058.当铺密码

+

★

收藏

0

2

讨论

编辑

讨论

本词条缺少概述图，补充相关内容使词条更完整，还能快速升级，赶紧来编辑吧！

一种加密算法，在CTF比赛题目中出现过。

中文名

当铺密码

当铺密码^[1]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

https://blog.csdn.net/qq_41638851

0x059.RSA算法

<http://www.factordb.com/index.php>

0x060.词频分析

<https://www.quipqiup.com>

0x061.加盐密码算法

<https://blog.csdn.net/JBlock/article/details/78446604>

0x062.play fair密码

<http://rumkin.com/tools/cipher/playfair.php>

<https://www.cnblogs.com/xdjun/p/7469841.html>



3、解密规则

- (1) 若c1 c2在同一行，对应明文p1 p2分别是紧靠c1 c2 左端的字母。其中最后一列被看做是第一列的左方。
- (2) 若c1 c2在同一列，对应明文p1 p2分别是紧靠c1 c2 上方的字母。其中最后一行被看做是第一行的上方。
- (3) 若c1 c2不在同一行，不在同一列，则p1 p2是由c1 c2确定的矩形的其他两角的字母。

密文: KQ SA MF PA OP MF PA
明文: dk ay fn ir mo fn ir

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

https://blog.csdn.net/qq_41638851

CTF中常出现的密码题的不完全总结

1.Base64编码

例: `VGhpcyBpcyBhbibiBleGFtcGx1`

注: XXencode与Base64形似, UUencode会出现 '=' 字符而Base64不会

2.Base32编码:

例: `KRUGS4ZANFZSAYLOEBSXQYLNQBWGK===`

通常在其后会有三个等号, 比较好判断, 当然没有'='也可以解得出来 0.0

3.UUencode编码

例: `2Z+^9Y+BJYIBOY+BJY+Z+Y:V0`

往往会出现+, -, ; 等比较奇怪的符号

4.ook 解码

例:

```
1      ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook.
ook.
2      ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook!
ook?
3      ook! ook! ook. ook? ook. ook. ook. ook. ook. ook. ook. ook.
ook.
4      ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook.
ook.
```

5 ook. ook. ook? ook. ook? ook! ook. ook? ook. ook. ook. ook. ook.
ook.

6 ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook.
ook.

7 ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook! ook.
ook?

8 ook. ook. ook. ook. ook. ook. ook. ook! ook? ook! ook! ook. ook? ook.
ook.

9 ook. ook. ook. ook. ook? ook. ook? ook! ook. ook? ook. ook. ook. ook.
ook.

10 ook. ook. ook. ook! ook. ook? ook. ook. ook. ook. ook. ook. ook. ook.
ook.

11 ook! ook? ook! ook! ook. ook? ook! ook! ook! ook! ook! ook! ook! ook!
ook?

12 ook. ook? ook! ook. ook? ook! ook! ook! ook! ook! ook! ook! ook! ook!
ook!

13 ook! ook! ook! ook! ook! ook! ook! ook. ook! ook! ook! ook! ook! ook!
ook!

14 ook. ook? ook. ook. ook. ook. ook. ook. ook! ook? ook! ook! ook.
ook?

15 ook! ook! ook! ook! ook! ook! ook? ook. ook? ook! ook. ook? ook! ook!
ook!

16 ook! ook! ook! ook! ook! ook! ook! ook! ook! ook! ook. ook? ook. ook.
ook.

17 ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook! ook?
ook!

18 ook! ook. ook? ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook.
ook.

19 ook. ook. ook? ook. ook? ook! ook. ook? ook. ook. ook. ook. ook. ook.
ook.

20 ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook.
ook.

21 ook. ook. ook. ook. ook! ook. ook? ook. ook. ook. ook. ook. ook. ook.
ook.

22 ook. ook. ook. ook. ook. ook. ook. ook! ook? ook! ook! ook. ook? ook!
ook!

23 ook! ook! ook! ook! ook! ook! ook! ook! ook! ook! ook! ook! ook? ook.
ook?

24 ook! ook. ook? ook! ook! ook! ook! ook! ook! ook! ook! ook! ook! ook!
ook.

25 ook? ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook.
ook.

26 ook. ook! ook? ook! ook! ook. ook? ook. ook. ook. ook. ook. ook. ook.
ook.

27 ook. ook. ook. ook. ook. ook. ook? ook. ook? ook! ook. ook? ook. ook.
ook.

28 ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook. ook.
ook.

29 ook. ook. ook. ook. ook. ook. ook! ook. ook? ook. ook. ook. ook. ook.
ook.

30 ook. ook. ook. ook. ook. ook. ook. ook! ook? ook! ook! ook. ook? ook!
ook!

31 ook! ook! ook! ook! ook! ook! ook! ook! ook! ook! ook? ook. ook? ook!
ook.

32 ook? ook! ook! ook! ook! ook! ook. ook! ook! ook! ook! ook! ook! ook!
ook!

5.URL编码 (UrlEncode) :

例: %e8%bf%99%e6%98%af%e4%b8%aa%e4%be%8b%e5%ad%90

因为出现了%因此也更容易认出来

6.unicode中文编码

例:

```
1 \u0054\u0068\u0069\u0073\u0020\u0069\u0073\u0020\u0061\u006e\u0020\u0065\u0078
  \u0061\u006d\u0070\u006c\u0065
```

特点, 每一行都有'\u'

7.HTML编码

例子: 以“d e”类似为特征

8.摩斯电码

例子: -...- -...- / -...- -...- / -...- -...- / -...- -...- / -...-
-...- -...-

摩斯电码可以用很多方式, 编码里的每一个元素可以进行多种方式的替换, 将一种符号用别的代替, 也可以用音调的高低代替, 十分灵活。

9.unescape码

例子: %u8FD9%u662F%u4E2A%u4F8B%u5B50

%u 开头也很好辨认

10.asc码

例子: 这个不需要例子了吧。。。。

11.16进制编码

秘文里出现不超过F的字母和数字, 则有很大可能就是16进制, 不多做解释。

12.Unicode英文编码

例子: 这是个例子

13.brainfuck加密

例子:

```
1 +++++ +++++ +++++[ ->++++ +++++ +++++ +<]>+ +++++ +++++ +++++ +. <+ + [->+
2 ++<]> +++++. <++++ [->-- --<]> ----- --. - --.<+ ++[-> ---<] >--- --.<+
3 +++++ + [->+ +++++ +<]>+ +++++ +++++ ++.<+ +++++ + [->- ---<] >--- --.<+
4 <++++ +++[- >++++ +++<] >++++ +++++ ++.<+ +++++ + [->- ---<] >--- --.<+ .<
```

14.二进制

例子: 0101010110

15.shellcode

例子:

```
1 \x3d\xbf\x5d\x3c\x99\x9\x5e\x3e\xc\x2a\x41\x31\xa1\xcb\x63\xcc\xa1\x54\xbf\x71
  \x8\xbf\x78\x31\x4c\x37
```

16.Quoted-printable编码

例子: =E8=BF=99=E6=98=AF=E4=B8=AA=E4=BE=8B=E5=AD=90

CTF密码学总结

1.base64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，base64就是一种基于64个可打印字符来表示二进制数据的表示方法。由于2的6次方等于64，所以每6个比特为一个单元，对应某个可打印字符。三个字节有24个比特，对应4个base64单元，即3个字节可表示4个可打印字符。它可用来作为电子邮件的传输编码。在base64中的可打印字符包括字母A-Z、a-z、数字0-9，这样共有62个字符，此外两个可打印符号在不同系统中而不同。

例如：编码'Man'

文本	M						a						n											
ASCII编码	77						97						110											
二进制位	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
索引	19						22						5						46					
Base64编码	T						W						F						u					

在此例中，base64算法将3个字符编码转为4个字符

base64索引表：

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

如果要编码的字节数不能被3整除，最后会多出1个或2个字节，那么可以使用下面的方法进行处理：先使用0字节值在末尾补足，使其能够被3整除，然后再进行base64的编码。在编码后的base64文本后加上一个或两个“=”号，代表补足的字节数。也就是说，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

例如：

文本 (1 Byte)	A																							
二进制位	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
二进制位 (补0)	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Base64编码	Q								Q								=							
文本 (2 Byte)	http://blog.csdn.net/pdsul61530247																							
二进制位	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0		
二进制位 (补0)	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0		
Base64编码	Q								k								M							

当看到号的加密方式时，可以考虑base64。例：cTZ1NQ解码之后为q5u5。

关于base32：

base32中只有大写字母（A-Z）和数字234567

The RFC 4648 Base 32 alphabet							
Value	Symbol	Value	Symbol	Value	Symbol	Value	Symbol
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		
6	G	15	P	24	Y		
7	H	16	Q	25	Z		
8	I	17	R	26	2	pad	=

关于base16：

base16中只有数字0-9以及大写字母ABCDEF

Table 5: The Base 16 Alphabet

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	0	4	4	8	8	12	C
1	1	5	5	9	9	13	D
2	2	6	6	10	A	14	E
3	3	7	7	11	B	15	F

用python3使用base64、base32、base16编码

base64在线解密 摘自：CTF中Crypty（密码类）入门必看

2.凯撒密码

凯撒密码作为一种最为古老的对称加密体制，在古罗马的时候都已经很流行，他的基本思想是：通过把字母移动一定的位数来实现加密和解密。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。例如，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X将变成A，Y变成B，Z变成C。由此可见，位数就是凯撒密码加密和解密的密钥。现今又叫“移位密码”，只不过移动的位数不一定是3位而已。

下面我们总结一下：

明密对照表：

明文：ABCDEFGHIJKLMNOPQRSTUVWXYZ

密文：TUVWXYZABCDEFGHIJKLMNOPS

注：广义上的凯撒是位移的。

凯撒是没有密匙的，即使没有密匙也能将它破解出来，因为凯撒移位密码只有25种密匙，最多就是将这25种可能性挨个检测一下可以了，这就是我们所说的暴力破解法。也可在用软件破解，不过我提倡用人工的。

(有关凯撒密码的我了解还不是很详细，后续。。。)

3.栅栏密码

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

明文：The quick brown fox jumps over the lazy dog

去空格：Thequickbrownfoxjumpsoverthelazydog

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

第一组：Teucbonojmsvrhlzdg

第二组：hqikrwxupoeteayo

密文：Teucbonojmsvrhlzghqikrwxupoeteayo

在线栅栏密码解密链接

例：

一只小羊翻过了2个栅栏KYsd3js2E{a2jda}解密后为KEY{sad23jjdsa2}

4.摩斯密码

摩尔斯电码（英语：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。是由美国人艾尔菲德·维尔（Alfred Lewis Vail）与萨缪尔·摩尔斯（Samuel Finley Breese Morse）在1836年发明。

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0和1两种状态的二进制代码，它的代码包括五种：

- 1.点 (.)
- 2.划 (-)
- 3.每个字符间短的停顿 (在点和划之间的停顿)
- 4.每个词之间中等的停顿
- 5.以及句子之间长的停顿

摩尔斯电码字母与数字对应表:

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

例如：

原文本: I LOVE YOU TOO

加密后: .. / .- / --- / ... - / . / - . - / --- / .. - / - / --- / ---

5.ASCII编码

ASCII码大致可以分为三部分组成:

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。

在线ASCII码与进制互相转换

ASCII表																									
(American Standard Code for Information Interchange 美国标准信息交换代码)																									
高四位	ASCII控制字符												ASCII打印字符												
	0000						0001						0010	0011	0100	0101	0110	0111							
	0						1						2	3	4	5	6	7							
低四位	十进制	字符	Ctrl	代码	转义	字符解释	十进制	字符	Ctrl	代码	转义	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	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
注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入。																									
2013/08/08																									

例：

突然天上一道雷电gndk€rlqhmtkwwp}z分析gndk€rlqhmtkwwp}z这个格式有点像flag{***}?

我们比较一下"gndk"与"flag"的ASCII码

gndk的10进制的ASCII码分别是：103 110 100 107

flag的10进制的ASCII码分别是：102 108 97 103

发现ASCII以此减少 1 2 3 4，所以以此类推解密得flag{lei_ci_jiami}

6.Brainfuck/Ook!编码

例：

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.
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! 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.

为Ook!编码，在线转文本后为flag{ok-ctf-1234-admin}

1.

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

2.

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

3.

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

4.

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

5.

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

6.

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

7.

++,<

这个为brainfuck，在线解码网站解码即可。

附上链接: Brainfuck/Ook!在线编码解码

还有许多类型的编码, 请参照: 一段base64的解体过程

7.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 丰富一下知识面2333。。。

8.培根密码

培根密码中的ab,代表的是数学二进制中的0和1.通过下列的密码表进行加密和解密:

第一种方式:

A aaaaa B aaaab C aaaba D aaabb E aabaa F aabab G aabba H aabbb I abaaa J abaab
K ababa L ababb M abbaa N abbab O abbba P abbbb Q baaaa R baaab S baaba T baabb
U babaa V babab W babba X babbb Y bbaaa Z bbaab

第二种方式

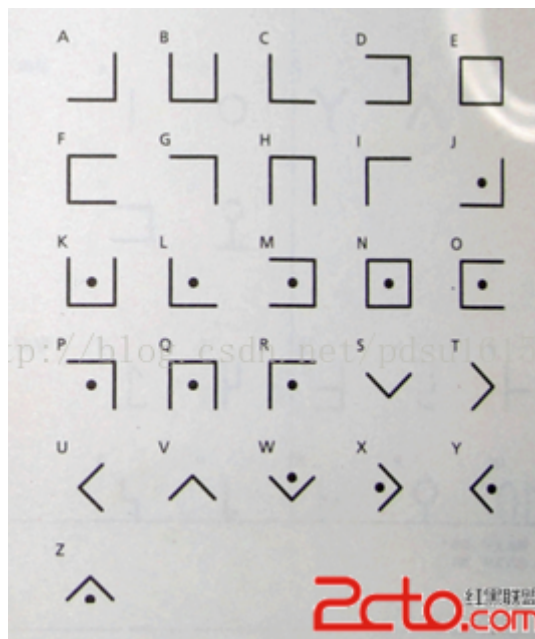
a AAAAA g AABBA n ABBA t BAABA
b AAAAB h AABBB o ABBAB u-v BAABB
c AAABA i-j ABAAA p ABBBA w BABAA
d AAABB k ABAAB q ABBBB x BABAB
e AABAA l ABABA r BAAAA y BABBA
f AABAB m ABABB s BAAAB z BABBB

感兴趣的可以参考大牛的文章: 培根密码

9.猪圈密码

猪圈密码(亦称朱高密码、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。即使使用符号, 也不会影响密码分析, 亦可用在其它替代式的方法。

明文字母和对应密文:



明文: THE QUICK BROWN FPX JUMPS OVER THE LAZY DOG

密文：

>ΠΟ Γ<ΓΛΩ ΠΓΕΥΘ ΓΕ> Γ<ΓΓΥ ΓΛΟ
Γ >ΠΟ Γ<ΓΛΩ ΠΓΕΥΘ ΓΕ> Γ<ΓΓΥ ΓΛΟ

10.unicode编码

例如：原文本：You had me at hello

编码后

\u0059\u0066\u0075\u0020\u0068\u0061\u0064\u0020\u006d\u0065\u0020\u0061\u0074\u0020\u0068\u0065\u006c\u006c\u006f

11.URL编码

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

12.ROT5/13/18/47编码

ROT5、ROT13、ROT18、ROT47 编码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，而不让其理解其意。

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

例：Good Day经过ROT47加密后为v@@5 s2JP ROT5/13/18/47在线加解密

13.Rabbit加密

就给出几个例子参考下好了。。。

明文I Love You小可爱无密匙加密后密文为U2FsdGVkX1/ouFei55jKdzY1fWNS4jxHVNf/AfKWjnBrOGY=

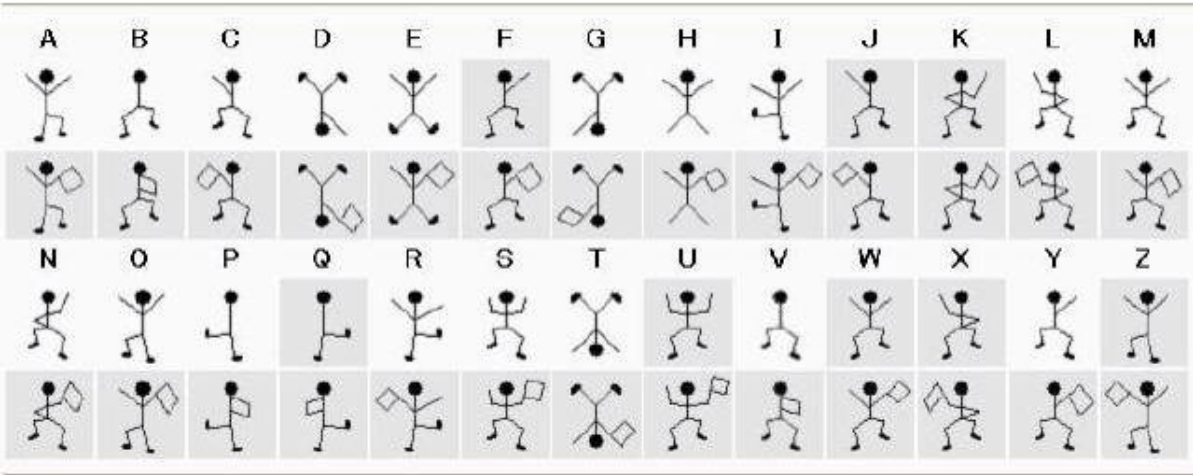
明文I Love You 521无密匙加密后密文为U2FsdGVkX19DvuEo5PvBA8TuLrM2t+EZBvUkzIAa

明文I Love You 521密匙为666加密后密文为U2FsdGVkX18w6vxXxux/ivRVwo3xMzTxmUyk7cHz

Rabbit在线加解密

14.跳舞的小人

来自夏洛克福尔摩斯在《归来记》中侦探案件使用的一种加密方式。



对应的明文是

1	AT ELRIGES （住在埃尔里奇）	COME ELSIE （来吧 埃尔茜）	NEVER （决不）
	ELSIE PREPARE TO MEET THY GOD （埃尔茜 准备见上帝吧）		

15.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。