ctf 密码学基础

原创

A1ienX 于 2018-05-26 00:56:09 发布
分类专栏： 密码学 ctf 文章标签： 密码学 ctf

最近想要学习ctf密码学的部分，于是开始了从百度百科的入门之旅。

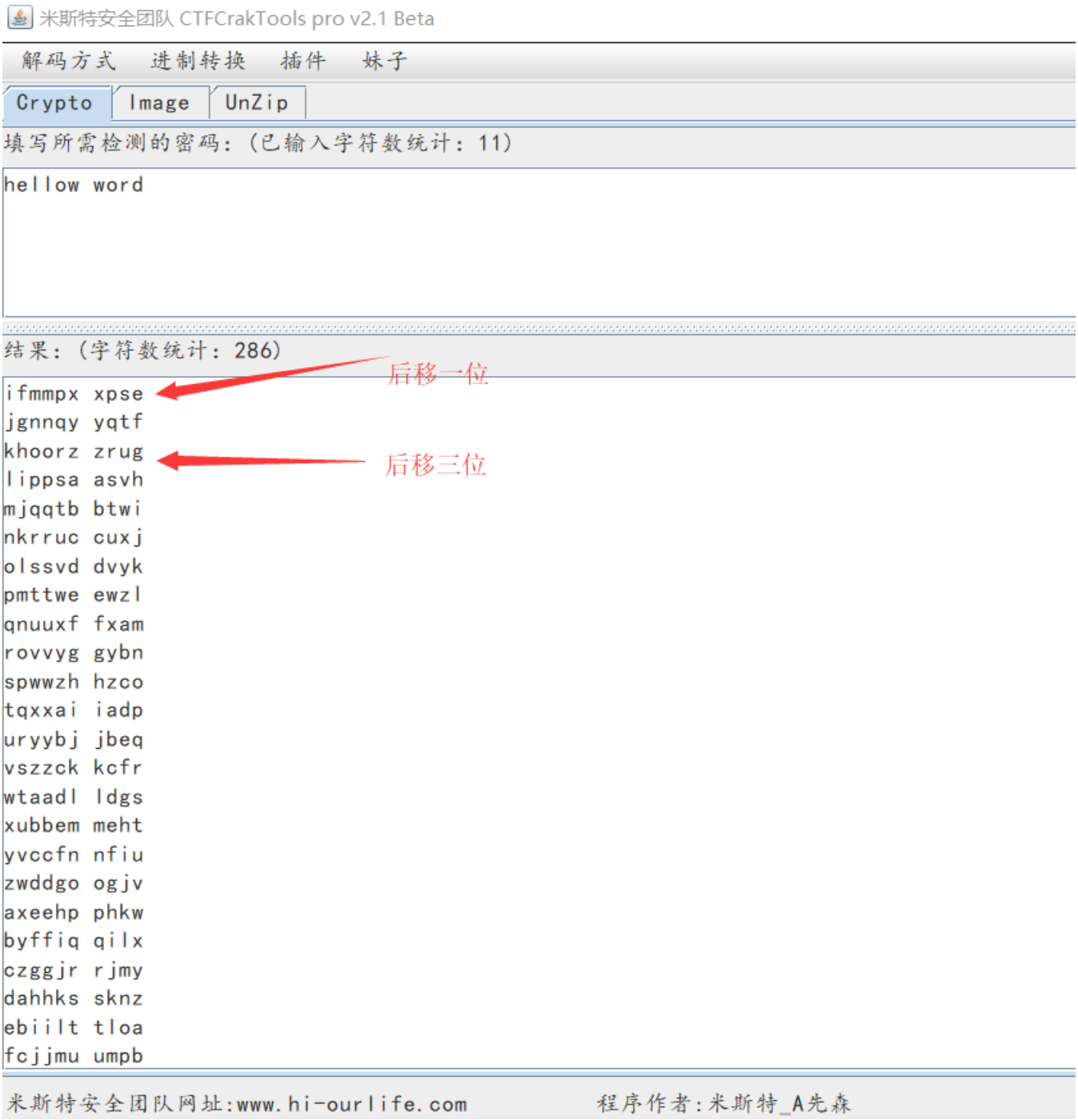
这里借助米斯特安全团队的一款工具来进行实际编码解码。

一，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。----百度百科

其实就是按照特定的顺序将字母替换。比如后移2位，那么a就变成c，d就变成f。

下面更直观的看一下：



二， Rot13加密

这种加密与凯撒十分类似，就是凯撒密码中移动13位的结果

三，培根密码

这是利用a和b(A和B也行)来表示二进制中的0和1，并以此来表示26个字母，培根密码5位一组。

这个加密方式主要特征是只有两种不同的字母，都可转换为ab形式

def就可以表示为AAABBAABAAAABAB

四，栅栏密码

百科上的例子：

一般比较常见的是2栏的栅栏密码。

比如 明文 ：THERE IS A CIPHER

去掉空格后变为：THEREISACIPHER

两个一组，得到：TH ER EI SA CI PH ER

先取出第一个字母：TEESCPE

再取出第二个字母：HRIAIHR

连在一起就是：TEESCPEHRIAIHR

还原为所需密码。

而解密的时候，我们先把密文从中间分开，变为两行：

T E E S C P E

H R I A I H R

再按上下上下的顺序组合起来：

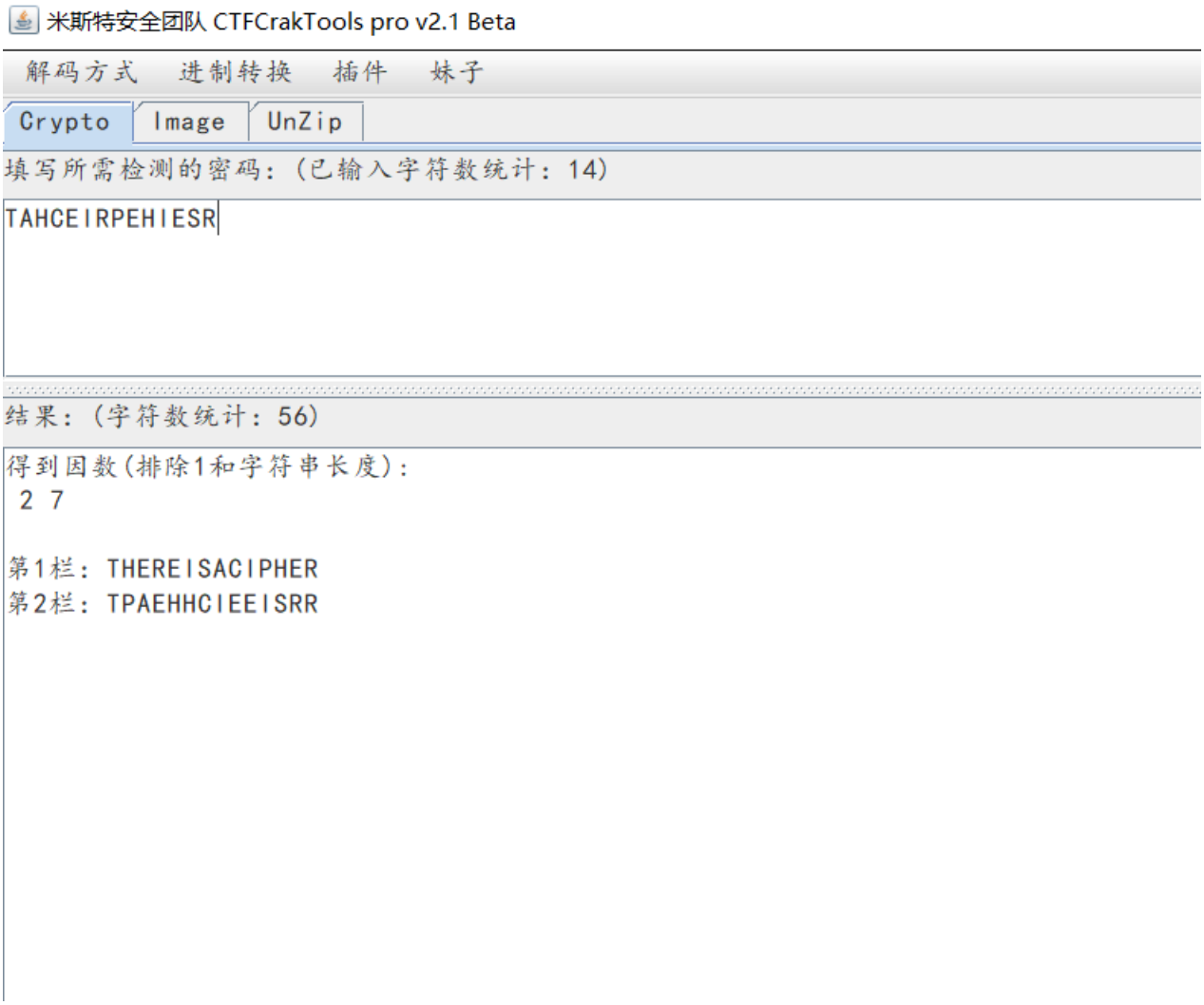
THEREISACIPHER

分出空格，就可以得到原文了：

THERE IS A CIPHER

但也存在不是两栏的情况，就需要对密文的总字数分解因数，尝试分栏的种类。

比如密文 TAHCEIRPEHIESR 14个字母，可以考虑2栏或7栏，解密如下：



根据不同情况，找到有意义的一组。

五，Base64编码

还是先看百科上的例子

转码过程例子：

3*8=4*6

内存1个字节占8位

转前： s 1 3

先转成ascii：对应 115 49 51

2进制： 01110011 00110001 00110011

6个一组（4组） **011100**110011**000100**110011

然后才有后面的 011100 110011 000100 110011

然后计算机是8位8位的存数 6不够，自动就补两个高位0了

所有有了 高位补0

科学计算器输入 **00**011100 **00**110011 **00**000100 **00**110011

得到 28 51 4 51

查对下照表 c z E z

核心思想在于将8位的二进制转换为6位的二进制。

而6位二进制共64种组合，可以表示10个数字，26个字母的大小写，外加符号+和/。一共64个。

对应编码

编号	字符	编号	字符	编号	字符	编号	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	/

有一点要特别指出，当明文个数不是三的倍数时，就会出现 明文数*8%6不为零的情况，

这是如果余数为2，就在末尾补两个等号，如果余数为4，就补一个等号。对于余下的2个或4个二进制

采用高位补零的方法。

比如密文：cmFubA== 的明文为ranl

因此我们可以得出base64编码后的特征：只含有数字，字母，和符号+ / =。

六， url编码

这是浏览器方便传输信息和加强安全性的编码。url编码是字符ASCII码的十六进制加上%的格式。

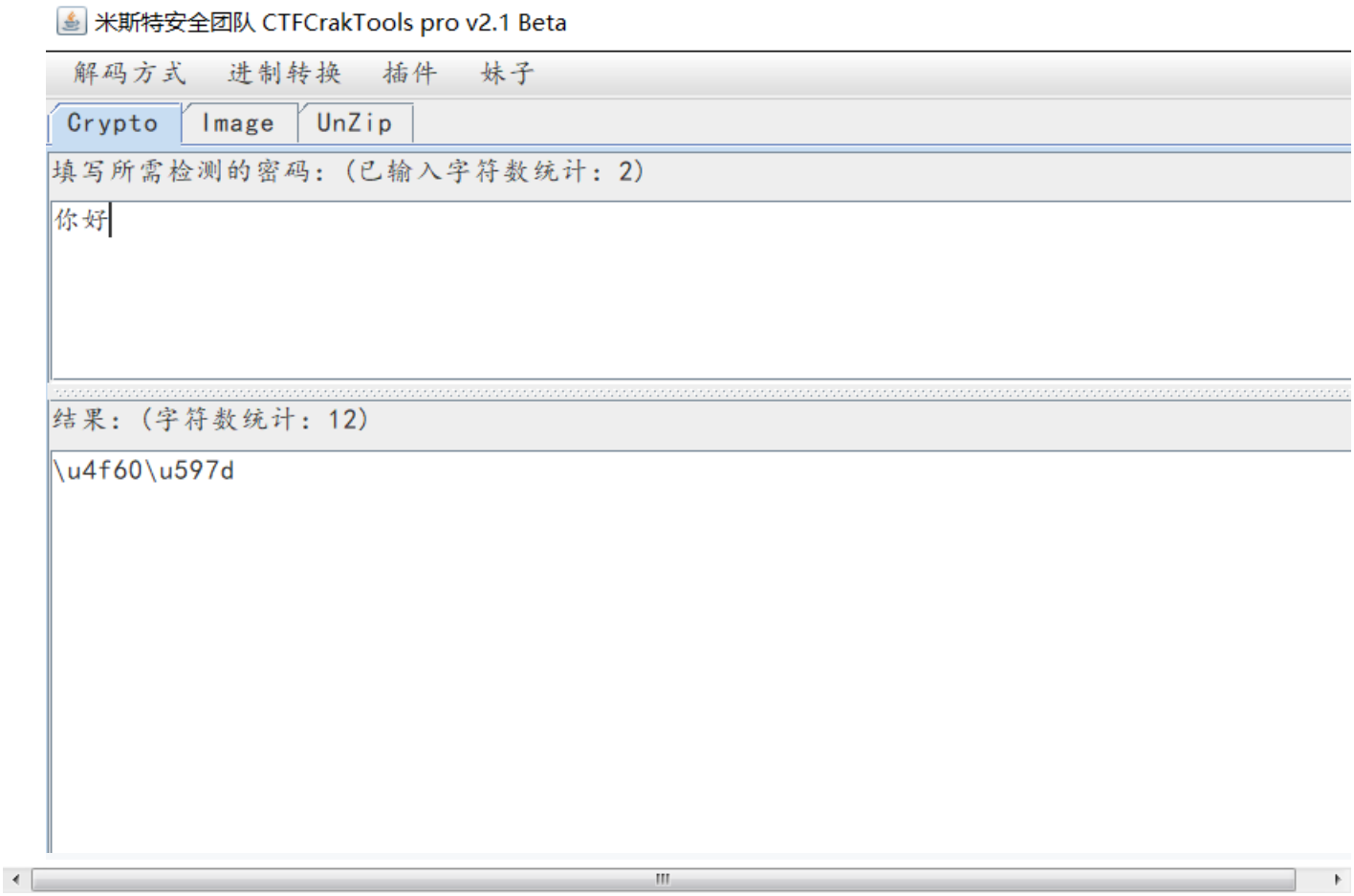
部分对应表：

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

但在url中不是对所有的字符都要进行url编码，一般的字母和数字是不会被编码的。会以原本形式传递。

七，Unicode编码

这是对ASCII码表的扩展，可以表示更多的字符，采用2个字节16位的储存形式。解码后可以看到16进制的表示形式。



其特征为以\u开头

八，utf8编码

这是为了利用内存采用的一种编码格式，这里只是看看其特征 &#开头ascii表之外的字符，

对于能用ascii表示的字符仍用ascii表示还是一个长度，就节省了内存。

九，hash算法

hash函数是一系列单向函数，就是这种运算是不可逆的，只可以通过hash运算得到一个hash值，而不能通过hash值得到原始数据。并且经过hash运算后得到的hash值为固定长度。

十，加盐hash

网站后台一般只储存用户密码的hash值，但也存在一定风险，因为如果黑客拿到了密码的hash后，可以的通过对比已经存在的明文与哈希的对应数据，进行对比，获得明文密码。因此，有了加盐hash,就是随机的在用户密码后加上一段字符后再进行hash运算，由于黑客不知道加的盐是什么，依然无法得到密码。

十一，MD5算法

MD5是hash算法中一种特殊的算法而已，也是hash算法，最后的结果为128个字符，但一般取其中的64或者32位。

最后附上一个在线MD5解密网站，当然MD5是不可逆的，这个网站只是通过查询已经存在的明文密文对照，进行查询操作

http://www.xmd5.org/md5/