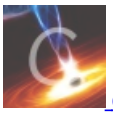
栅栏密码的变种WWW型

原创

影子019 于 2019-07-16 15:48:17 发布 4908 收藏 13

分类专栏： crypto 文章标签： ctf crypto

本文链接：https://blog.csdn.net/qinying001/article/details/96134356



栅栏密码：

所谓栅栏密码，就是把要加密的明文分成N个一组，然后把每组的第1个字连起来，形成一段无规律的话。 不过栅栏密码本身有一个潜规则，就是组成栅栏的字母一般不会太多。

传统型：

```
假如有一个字符串：123456789
取字符串长度的因数进行分组，假如key=3
1 2 3  \\分组情况，每三个数字一组，分为三组
4 5 6
7 8 9
然后每一组依次取一个数字组成一个新字符串：147258369 \\加密完成的字符串
```

解密方法：

遇到这种的栅栏加密的密文，解密的key值就是字符串的长度除以加密的key值 (de_key = len / key）再用de_key将密文字符串加密就可以得到原文字符串。

加密解密网站：http://www.zjslove.com/3.decode/zhalan/index.html

WWW型：

```
同样一个字符串：123456789
key=3
1----5----9  \\让数字以W型组织，同样是三组，但每组的数量不一定相同
-2--4-6--8
--3----7--
加密密文：159246837
```

加密脚本

```python
def encode(string, key):#需要加密的字符串以及加密栏数
    i = 0
    enlist = []
    for j in range(0, key):
        enlist.append('')#添加分组，列表的一个元素相当于一个分组

    while i < len(string):#分组重排进行加密
        for k in range(0, key):
            if i >= len(string):
                break
            enlist[k] += string[i]
            i += 1
        for k in range(1, key-1):
            if i >= len(string):
                break
            enlist[key-1-k] += string[i]
            i += 1
        enstr = ''
    for i in range(key):
        enstr += enlist[i]
    return enstr
```

解密方法：

WWW型的加密密钥就不只能是字符串长度的因子，小于其长度的任何一个数都可能是其key值，所以第一步也是确定密钥。

```
字符串：159246837
假设 key = 3 \\具体情况下可以遍历key值，每个值都算一次
确定key值之后就可以确定字符串的排部方法，即：
分组1：1---2---3 \\ 将第一个行第一个数到第一行的下一个数看作为一部分（相同数字），  其长度为这部分长度len = 2key-2
分组2：-1-1-2-2  \\确定长度的排布之后将密文带入解密即可
分组3：--1---2-
原文：123456789
```

解密脚本

```python
def decode(string, key):#解密字符串以及解密栏数
    de_key = 2*key - 2#一个部分的长度
    length = len(string)//de_key#确定有多少个完整部分
    r = len(string)%de_key#最后不完整部分的长度
    delist = []
    for i in range(key):
        delist.append('')#重新排布分组
    #确定第一个分组
    if r == 0:
        delist[0] += string[0:length]
        s = length
    else:
        delist[0] += string[0:length+1]
        s = length+1
    #确定第二个到第key-1个分组
    for i in range(1, key-1):
        l = length*2#这几个分组长度至少是完整部分数量的两倍
        #最后一个不完整部分对应当前分组有几个元素
        if r > i:
            l += 1
        if r > de_key-i:
            l += 1

        delist[i] += string[s:s+l]
        s = s+l
    #确定最后一个分组
    delist[key-1] += string[s:]
#排布分组确定原文字符串
    destr = ''
    j = 0
    for i in range(0, len(string)):
        destr += delist[j][0]
        delist[j] = delist[j][1:]
        if j == key-1:
            flag = 0
        if j == 0:
            flag = 1
        if flag:
            j += 1
        else:
            j -= 1
    return destr
```

加密解密网站：http://www.atoolbox.net/Tool.php?Id=777

新创建的网络安全公众号，欢迎各位朋友们的关注，一起学习