实验吧——(crypto)古典密码 writeup

原创

嗯哼哈嘿 于 2019-05-13 23:32:53 发布 360 收藏

分类专栏： CTF 文章标签： CTF 古典密码 实验吧

先学习一些关于古典密码理论知识：

古典密码有两种：**置换和代换**

一个是换了个原来有的，一个是换了个原来没有的。学术点讲就是前者明文和密文空间一样，后者 不一样。

置换密码：列置换和周期置换

列置换：操作及输出都是以列为单位。将明文以**密钥长度**为列数形成矩阵，按照**密钥的顺序**进行列选出，然后一列一列的输出。

举个例子：

我们以字符串“hello-my-cipher”为例来演示加密过程

选择密钥，我们这里使用“4213”作为密钥。该密钥共4位，表示中间结果的矩阵共4列，4213表示按照第四列，第二列，第一列，第三列的顺序读出形成密文

生成中间结果矩阵(该行不够4个则用明文中不包含的固定字符填充，这里使用‘@’)

```
h e l l
o - m y
- c i p
h e r @
```

按照密钥所示的列顺序读出（按一列一列读出）

lyp@e-ceho-hlmir

至此加密完成。

解密

解密过程即按照密钥所示的长度顺序恢复出矩阵，再按行读取即可。

已知明文破解

根据置换密码算法的特点，破解的主要难点是确定密钥的长度。我们可以根据**密文的第一个和第二个字符在明文中的位置之差得出密钥的长度**，但这中方法并不总是有效，例如helloeye用密钥1234加密后密文前两个字符是he，而h，e在明文中的位置有1和4，再例如helloeye用3124加密后前两个字符是ly，而l，y在明文中的位置有4和5，所以这个方法无法完全确定密钥长度。假如我们确定了密钥长度，就可以把明文形成矩阵，然后比对密文和明文矩阵的列，便能确定按列读出的顺序，也就得到了密钥。（这里还有碰到明文矩阵中两列完全一样情况，这样就得不到完整的密钥）

例子：

```
明文：abcdefgh
密钥：2143
加密结果：bfaedhcg（8个字符）
根据bf在明文中的位置，第2和第6，得到密钥长度为4，则每列长度为8/4=2，然后生成矩阵
a b c d
e f g h
密文中前2个字符为bf，找到矩阵中bf列为第二列，密钥第一位是2，密文中接下来2个字符是ae，找到矩阵中ae
列为第一列，密钥第二位是1，按这个方法，就可以得到密钥为2143了
```

现在来看看**题目**：

密文内容如下{79 67 85 123 67 70 84 69 76 88 79 85 89 68 69 67 84 78 71 65 72 79 72 82 78 70 73 69 78 77 125 73 79 84 65}

请对其进行解密

提示：1.加解密方法就在谜面中

2.利用key值的固定结构

格式：CTF{}

我们先看到密文：应该是ASCII码，所以我们可以使用网页F12，使用javascript的函数将ASCII码转换为字幕字母

```
> String.fromCharCode(79,67,85,123,67,70,84,69,76,88,79,85,89,68,69,67,84,78,7
  1,65,72,79,72,82,78,70,73,69,78,77,125,73,79,84,65)
< "OCU{CFTELXOUYDECTNGAHOHRNFIENM}IOTA"
```

因为原字符串为35位：35=5*7

因为是列置换，一行5个的话，无法得到CTF{}这个格式，

所以：

```
OCU{CFT
ELXOUYD
ECTNGAH
OHRNFIE
NM}IOTA
```

由提示：利用key值的固定结构，我们可以尝试多次，因为第一行有两个C

第二列/第五列，第七列，第六列，第四列，还要考虑}（第三列）是最后一列

```
第一次尝试（2764153）
CTF{OCU
LDYOEUX
CHANEGT
HEINOFR
MATINO}
第二次尝试(2764513)
CTF{COU
LDYOUEX
CHANGET
HEINFOR
MATNIO}
得到答案
```

参考：

http://www.cnblogs.com/gebilaowang/p/4623290.html

https://blog.csdn.net/j5856004/article/details/78159723