EN | ZH

总结

古典密码分析思路

CTF 中有关古典密码的题目，通常是根据密文求出明文，因此采用**唯密文攻击**居多，基本分析思路总结如下：

1. 确定密码类型：根据题目提示、加密方式、密文字符集、密文展现形式等信息。
2. 确定攻击方法：包括直接分析、蛮力攻击、统计分析等方法。对于无法确定类型的特殊密码，应根据其密码特性选用合适的攻击方法。
3. 确定分析工具：以在线密码分析工具与 Python 脚本工具包为主，以离线密码分析工具与手工分析为辅。

以上唯密文攻击方法的适用场景与举例如下：

攻击方法	适用场景	举例
直接分析法	由密码类型可确定映射关系的代换密码	凯撒密码、猪圈密码、键盘密码等
蛮力攻击法	密钥空间较小的代换密码或置换密码	移位密码、栅栏密码等
统计分析法	密钥空间较大的代换密码	简单替换密码、仿射密码、维吉尼亚密码等

实验吧 围在栅栏里的爱

题目描述

> 最近一直在好奇一个问题，QWE 到底等不等于 ABC?
>
> -.- .. --.- .-.. .-- - ..-. -.-. --.- --. -. ... --- ---
>
> flag 格式：CTF{xxx}

首先，根据密码样式判断是摩斯电码，解密后得到 `KIQLWTFCQGNSOO`，看着也不像 flag，题目中还有还有栅栏与 `QWE到底等不等于ABC`，两个都试了试之后，发现是先 QWE 然后栅栏可得到结

果。

首先键盘 QWE 解密，试着解密得到 `IILYOAVNEBSAHR`。继而栅栏解密得到 `ILOVESHIYANBAR`。

2017 SECCON Vigenere3d

程序如下

```
# Vigenere3d.py
import sys
def _l(idx, s):
    return s[idx:] + s[:idx]
def main(p, k1, k2):
    s = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz_{}"
    t = [[_l((i+j) % len(s), s) for j in range(len(s))] for i in
range(len(s))]
    i1 = 0
    i2 = 0
    c = ""
    for a in p:
        c += t[s.find(a)][s.find(k1[i1])][s.find(k2[i2])]
        i1 = (i1 + 1) % len(k1)
        i2 = (i2 + 1) % len(k2)
    return c
print main(sys.argv[1], sys.argv[2], sys.argv[2][::-1])

$ python Vigenere3d.py SECCON{************************} **************
POR4dnyTLHBfwbxAAZhe}}ocZR3Cxcftw9
```

解法一:

首先，我们先来分析一下 t 的构成 $$ t[i][j]=s[i+j:]+s[:i+j] \ t[i][k]=s[i+k:]+s[:i+k] $$

$t[i][j][k]$ 为 $t[i][j]$ 中的第 k 个字符，$t[i][k][j]$ 为 $t[i][k]$ 中的第 j 个字符。无论是 $i+j+k$ 是否超过 `len(s)` 两者都始终保持一致，即 $t[i][j][k]=t[i][k][j]$ 。

故而，其实对于相同的明文来说，可能有多个密钥使其生成相同的密文。

然而上面分析就是单纯地分析而已，，下面开始正题。

不难看出，密文的每一位只与明文的相应位相关，而且，密钥的每一位的空间最大也就是 s 的大小，所以我们可以使用爆破来获取密钥。这里根据上述命令行提示，可以知道密钥长度为 14，恰好明文前面 7 个字节已知。恢复密钥的 exp 如下

```
def get_key(plain, cipher):
    s = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz_{}"
    t = [[_l((i + j) % len(s), s) for j in range(len(s))]
         for i in range(len(s))]
    i1 = 0
    i2 = 0
    key = ['*'] * 14
    for i in range(len(plain)):
```

```
        for i1 in range(len(s)):
            for i2 in range(len(s)):
                if t[s.find(plain[i])][s.find(s[i1])][s.find(s[i2])] ==
cipher[
                        i]:
                    key[i] = s[i1]
                    key[13 - i] = s[i2]
    return ''.join(key)
```

恢复明文的脚本如下

```
def decrypt(cipher, k1, k2):
    s = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz_{}"
    t = [[_l((i + j) % len(s), s) for j in range(len(s))]
         for i in range(len(s))]
    i1 = 0
    i2 = 0
    plain = ""
    for a in cipher:
        for i in range(len(s)):
            if t[i][s.find(k1[i1])][s.find(k2[i2])] == a:
                plain += s[i]
                break
        i1 = (i1 + 1) % len(k1)
        i2 = (i2 + 1) % len(k2)
    return plain
```

得到明文如下

```
➜  2017_seccon_vigenere3d git:(master) python exp.py
SECCON{Welc0me_to_SECCON_CTF_2017}
```

解法二

关于此题的分析：

1. 考虑到在程序正常运行下，数组访问不会越界，我们在讨论时做以下约定：
   $arr[index] \Leftrightarrow arr[index\%len(arr)]$
2. 关于 python 程序中定义的 `_l` 函数，发现以下等价关系：
   $\_l(offset, arr)[index] \Leftrightarrow arr[index + offset]$
3. 关于 python 的 main 函数中三维矩阵 t 的定义，发现以下等价关系：
   $t[a][b][c] \Leftrightarrow \_l(a + b, s)[c]$
4. 综合第 2 第 3 点的观察，有如下等价关系：$t[a][b][c] \Leftrightarrow s[a + b + c]$
5. 我们将 s 视为一种编码格式，即：编码过程 s.find(x)，解码过程 s[x]。并直接使用其编码结果的数字替代其所代指的字符串，那么加密过程可以用以下公式表示：
6. $e = f + k1 + k2$
7. 其中，e 是密文，f 是明文，k1 与 k2 是通过复制方法得到、与 f 长度一样的密钥，**加法是向量加。**

所以我们只需要通过计算 `k1+k2` ，模拟密钥，即可解密。关于此题的解密 python 脚本：

```
# exp2.py
enc_str =
'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz_{}'
dec_dic = {k:v for v,k in enumerate(enc_str)}
encrypt = 'POR4dnyTLHBfwbxAAZhe}}ocZR3Cxcftw9'
flag_bg = 'SECCON{************************}'

sim_key = [dec_dic[encrypt[i]]-dec_dic[flag_bg[i]] for i in range(7)] # 破解
模拟密钥
sim_key = sim_key + sim_key[::-1]

flag_ed = [dec_dic[v]-sim_key[k%14] for k,v in enumerate(encrypt)] # 模拟密钥
解密
flag_ed = ''.join([enc_str[i%len(enc_str)] for i in flag_ed]) # 解码
print(flag_ed)
```

得到明文如下：

```
$ python exp2.py
SECCON{Welc0me_to_SECCON_CTF_2017}
```

消失的三重密码

密文

```
of zit kggd zitkt qkt ygxk ortfzoeqs wqlatzwqssl qfr zvg ortfzoeqs
yggzwqssl. fgv oy ngx vqfz zg hxz zitd of gft soft.piv dgfn lgsxzogfl qkt
zitkt? zohl:hstqlt eiqfut zit ygkd gy zit fxdwtk ngx utz.zit hkgukqddtkl!
```

使用 quipquip 直接解密。

评论