Github 账号：Nora-Qiu

实验摘要：

题目提供了一条经过 AES 加密，使用初始零向量和 01-00 填充的 CBC 模式 的消息。同时提供了能求出相应的密钥的字符串，但这个字符串并不完整。
实验目的是找到已给密文的明文。
实验分为三个步骤：
一、求出传输过程中丢失的字符？
二、求密钥 K_{ENC}
三、求出明文

实验题目

题目提供了一条经过 AES 加密，使用初始零向量和 01-00 填充的 CBC 模式 的消息。同时提供了相应的密钥，但这个密钥并不完整。
实验目的是找到已给密文的明文。
基于基本访问控制(Basic Access Control, BAC)协议的密钥 KENC 被生成并应用。为了解密，已经传输了以下字符，从中可以导出 KEN:
12345678<8<<<1110182<111116?<<<<<<<<<<<<<<<4
在传输过程中一个字符丢失了，并用?表示。为了求得 KENC 需要使？再次可现。经过 AES 加密的消息包含一个要作为解决方案输入的词。

通过题目中所给的参考资料计算得到 K_{ENC} 并解密以下 64 编码的明文：
9MgYwmuPrjiecPMx61O6zIuy3MtIXQQOE59T3xB6uOGyf1gYs2i3K9Jx
aaOzj4gTMazJuApwd6+jdyeI5iGHvhQyDHGVlAuYTgJrbFDrfB22Fpil2N fNnWFBTXyf7SDI

实验内容

根据题目信息，为了求出 K_{ENC} 并解得明文，按照以下步骤进行实验：

一、求出传输过程中丢失的字符？

MRZ character positions (line2)	*Field no. in VIZ*	*Data element*	*Specifications*	*Number of characters*	*References and notes**
10		Check digit	Shall be calculated as specified in Doc 9303-3 and positioned as specified in paragraph 4.2.4.	1	Notes b, d.
11 to 13	08	Nationality	As a three-letter code representing the holder's nationality as listed in Doc 9303-3. Spaces are replaced by filler characters.	3	Notes a, d, f.
14 to 19	9	Date of birth	See Doc 9303-3 for details.	6	Notes b, d, i.
20		Check digit	Shall be calculated as specified in Doc 9303-3 and positioned as specified in paragraph 4.2.4.	1	Notes b, d.
21	11	Sex	F = female; M = male; < = unspecified.	1	Notes a, d.
22 to 27	16	Date of expiry	See Doc 9303-3 for details.	6	Notes b, d, i.
28		Check digit	Shall be calculated as specified in Doc 9303-3 and positioned as specified in paragraph 4.2.4.	1	Notes b, d.

根据[1]中给出的字符编码类型表可以确定已传输字符串中的？是第 28 位，为 22 到 27 位的校验位。

4.9 Check Digits in the MRZ

A check digit consists of a single digit computed from the other digits in a series. Check digits in the MRZ are calculated on specified numerical data elements in the MRZ. The check digits permit readers to verify that data in the MRZ is correctly interpreted.

A special check digit calculation has been adopted for use in MRTDs. The check digits shall be calculated on modulus 10 with a continuously repetitive weighting of 731 731 ..., as follows.

Step 1. Going from left to right, multiply each digit of the pertinent numerical data element by the weighting figure appearing in the corresponding sequential position.

Step 2. Add the products of each multiplication.

Step 3. Divide the sum by 10 (the modulus).

Step 4. The remainder shall be the check digit.

For data elements in which the number does not occupy all available character positions, the symbol < shall be used to complete vacant positions and shall be given the value of zero for the purpose of calculating the check digit.

When the check digit calculation is applied to data elements containing alphabetic characters, the characters A to Z shall have the values 10 to 35 consecutively, as follows:

根据[2]中所给的信息，得到求出？值的方法：校验位用模 10 重复加权 731 731…

步骤 1：从左到右依次把数据中的元素和对应位权数相乘。

步骤 2：把他们相乘的结果相加

步骤 3：把相加和除以 10

步骤 4：余数为校验位数

代码实现如下：

```python
#计算丢失的字符，使？可见
def visible(remainder):
    a = [1, 1, 1, 1, 1, 6]
    b = [7, 3, 1, 7, 3, 1]
    c = 0
    for i in range(0,6):
        c = c + a[i]*b[i]
    remainder = c% 10
    remainder = str(remainder)
    return(remainder)
```

可得？=7

完整的字符串为：12345678<8<<<1110182<1111167<<<<<<<<<<<<<<

二、求密钥 K_{ENC}

1) The inspection system reads the "MRZ_information". The "MRZ_information" consists of the concatenation of Document Number, Date of Birth and Date of Expiry, including their respective check digits, as described in Doc 9303-4, Doc 9303-5 or Doc 9303-6 for document form factors TD3, TD1 and TD2, respectively, from the MRZ using an OCR-B reader. Alternatively, the required information can be typed in; in this case it SHALL be typed in as it appears in the MRZ. The most significant 16 bytes of the SHA-1 hash of this "MRZ_information" are used as key seed to derive the Document Basic Access Keys using the key derivation mechanism described in Section 9.7.2.

根据[3]中的信息可知道，MRZ_information 的 sha-1 哈希散列中 16 字节被用作密钥种子，并用以下密钥派生机制来产生密钥。

求密钥种子 K_seed 代码：

```python
def seed(MRZ):

    sha1_MRZ = hashlib.sha1(MRZ.encode('utf-8')).hexdigest()
    #print(sha1_MRZ)
    K_seed = sha1_MRZ[0:32]
    print('K_seed is:', K_seed)
    return(K_seed)
```

9.7.2 Document Basic Access Keys

The computation of two key 3DES keys from a key seed (K) is used in the establishment of the Document Basic Access Keys K_{Enc} = **KDF**(K,1) and K_{MAC} = **KDF**(K,2).

根据[4]可知，通过密钥种子 K_seed 计算得出的两个 3DES 密钥被用来建立 K_{ENC}。

函数 KDF（K，1）在 9.7.1 中给出

9.7.1 Key Derivation Function

The key derivation function **KDF**(K,c), is defined as follows:

Input: The following inputs are required:

- The shared secret value K (REQUIRED)
- A 32-bit, big-endian integer counter c (REQUIRED)

Output: An octet string keydata.

Actions: The following actions are performed:

- keydata = **H**(K || c)
- Output octet string keydata

The key derivation function **KDF**(K,c) requires a suitable hash function denoted by **H**(), i.e the bit-length of the hash function SHALL be greater or equal to the bit-length of the derived key. The hash value SHALL be interpreted as big-endian byte output.

Note.— The shared secret K is defined as an octet string. If the shared secret is generated with ECKA [TR-03111], the x-coordinate of the generated point SHALL be used.

KDF 函数输入共享密钥 K_seed 和一个 32 位的整数计数器，输出八位字节的数据串。
具体步骤如下：
步骤 1：K_seed 和计数器 c 级联
步骤 2：对级联结果用 SHA1 加密（这里要求哈希函数的位长大于或者等于产生密钥的长度）
步骤 3：输出 8 位字节串 keydata
生成 keydata 代码如下

```
seed=seed(MRZ)
C = '00000001'
outcome = seed+C
c = binascii.a2b_hex(outcome) #转换成ASCii编码的字符串
keydata = hashlib.sha1(c).hexdigest()
print('output keydata:', keydata)
```

9.7.1.1 3DES

To derive 128-bit (112-bit excluding parity bits) 3DES [FIPS 46-3] keys the hash function SHA-1 [FIPS 180-2] SHALL be used and the following additional steps MUST be performed:

- Use octets 1 to 8 of keydata to form keydataA and octets 9 to 16 of keydata to form keydataB; additional octets are not used.
- Adjust the parity bits of keydataA and keydataB to form correct DES keys (OPTIONAL).

两个 3DES 密钥通过如上方式得出：

- KDF 函数输出的八位字节串 1 到 8 字节用来生成 K_A，9 到 16 字节用来生成 K_B，其余的字节不被使用。
- 调整 K_A 和 K_B 的奇偶校验位生成正确的 DES 密钥

```
#生成KA,KB

K_A = keydata[0:16]
K_B = keydata[16:32]
```

```
#奇偶校验位的判断
def parity_check(x):
    k = []
    a = bin(int(x,16))[2:]
    for i in range(0,len(a),8):
        if (a[i:i+7].count("1"))%2 == 0:#1的个数为偶数，添加1
            k.append(a[i:i+7])
            k.append('1')
        else :
            k.append(a[i:i+7])#1的个数为奇数
            k.append('0')
    a1 = hex(int(''.join(k),2))
    print("this is " + x + "---" +a1)
    return a1[2:]
k_1 = parity_check(K_A)
k_2 = parity_check(K_B)
key = k_1 + k_2
```

三、求出明文

因为所给的消息是 base64 加密的，所以要先对其 base64 解码：

```
s='9MgYwmuPrjiecPMx61O6zIuy3MtIXQQ0E59T3xB6u0Gyf1gYs2i3K9J\
xaa0zj4gTMazJuApwd6+jdyeI5iGHvhQyDHGVlAuYTgJrbFDrfB22Fpil2NfNnWFBTXyf7SDI'
cipher = base64.b64decode(s.encode('utf-8'))
```

最后用密钥解密就得到明文啦

```
m = AES.new(binascii.unhexlify(key),AES.MODE_CBC,binascii.unhexlify('00000000000000000000000000000000'.encode()))
print('the base64-decoded message is:',cipher)
print('the plaintext is :',m.decrypt(cipher))
```

运行结果是：

```
The full str: 12345678<8<<<1110182<1111167<<<<<<<<<<<<<<
K_seed is: a095f0fdfe51e6ab3bf5c777302c473e
output keydata: eb8645d97ff725a998952aa381c5307909962536
parity check:eb8645d97ff725a9-->0xea8645d97ff725a8
parity check:98952aa381c53079-->0x98942aa280c43179
key= ea8645d97ff725a898942aa280c43179
the base64-decoded message is: b"\xf4\xc8\x18\xc2k\x8f\xae8\x9ep\xf31\xeb5\xba\xcc\x8b\xb2\xdc\xcbH]\x044\x13\x9f5\xdf\x10z\xbbA\xb2\x7fX\x18\xb3h\xb7+\xd2qi\xad3\x8f\x88\x131\xac\xc9\xb8\npw\xaf\xa3
the plaintext is : b'Herzlichen Glueckwunsch. Sie haben die Nuss geknackt. Das Codewort lautet: Kryptographie!\x01\x00\x00\x00\x00\x00\x00\x00'
```

但令人遗憾的是，当我在 MTC3 网站输入解得的明文后显示

不知道我的问题出现在哪里

实验总结

刚开始做的时候 google 到了英文版的参考资料，但是没有找到解密的示例。代码慢悠悠写完之后发现结果完全不对，以下是几个我犯错的地方：

①把 K_seed 和计数器级联之后没有转换成 ASCii 编码的字符串，在 MTC 论坛上看到有相关提问于是改了过来

```
outcome = seed+C
c = binascii.a2b_hex(outcome) #转换成ASCii编码的字符串
```

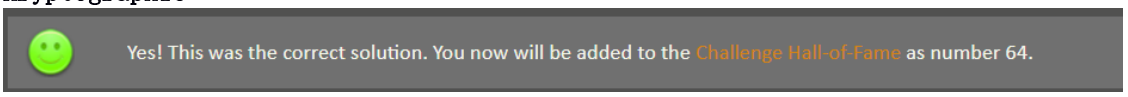
② 对 MRZ_information 选择错了，一开始以为是完整的字符串 12345678<8<<<1110182<1111167<<<<<<<<<<<<<<4，后来在论坛看到应该选择护照信息第二行，删除国家的 3 位数字和性别的 1 位数字，并且最后一位应该是 2

> the last digit problem:
> when you calculate the last digit correctly, then the result is 2, as you mentioned.
>
> for the correct calculations, you take the second line, remove the 3 digits of the country and 1 digit of the gender. then you calculate the checksum over the remaining 39 digits.
>
> when you calculate over all 43 digits, then the checksum is 4. i think, that happened during the creation of the challenge.
>
> best regards,
> jomandi

结果还是不对，看到同学的中文参考资料才知道应该只用前 28 位并删除国家的 3 位数字和性别的 1 位数字。正确的 MRZ_information 是 12345678<811101821111167

后来为了检查仔细是否还遗漏了信息，再看了一遍英文参考资料，发现其实在说明 K——seed 如何得到的段落中明确说了 MRZ_information 由证件号码，出生日期以及到期日构成。

③题目中说了加密的消息包含一个要作为解决方案输入的词。所以应该输入的是单词 Kryptographie



如果根据参考资料步骤进行解题，过程应该是相当清晰的。由于自己的疏忽，在求密钥种子种子那里浪费了相当长时间，以后一定仔细读题和文献！

参考文献

- https://www.mysterytwisterc3.org/en/mtc3forum/viewtopic.php?f=5&t=217&start=10
- https://www.icao.int/publications/Documents/9303_p11_cons_en.pdf
- https://www.icao.int/publications/Documents/9303_p3_cons_en.pdf