com.example.ggad.b中拿到PRESET\_VALUE

com.example.ggad.c中拿到另一个字符串

直接梭哈

# 给定的二进制字符串

ci = '0100001000110100010001000011001100110111001100110100010000110000001101000011001101000110001101100100010000110000'

c0 = '65489I5Y5S3FD0'

def vigenere\_decrypt(text, key):

decrypted\_text = []

key\_index = 0

for char in text:

if char.isalpha():

# Decrypt the character using Vigenère cipher

decrypted\_char = chr(((ord(char.upper()) - ord('A')) - (ord(key[key\_index % len(key)].upper()) - ord('A')) + 26) % 26 + ord('A'))

# Append decrypted char, maintaining original case

if char.islower():

decrypted\_text.append(decrypted\_char.lower())

else:

decrypted\_text.append(decrypted\_char)

key\_index += 1

else:

decrypted\_text.append(char)

return ''.join(decrypted\_text)

def decrypt(text, key):

return vigenere\_decrypt(text, key)

def a(cs):

# Assuming KeyManager.getKey() returns the decryption key as a string.

key = 'ExpectoPatronum'

return decrypt(cs, key)

c2 = a(c0)

c1 = ''

for i in range(0, len(ci), 8):

c1 += chr(int(ci[i:i + 8], 2))

c = ''

for i in range(14):

c+=c1[i]

c+=c2[i]

byte\_array = [int(c[i:i+2], 16) for i in range(0, len(c), 2)]

c3 = ''

for i in byte\_array:

a = bin(i)[2:]

c3 += ("{:0>8}".format(a))

str1 = ''

for i in c3:

if(i == '1'):

str1 += '0'

if(i == '0'):

str1 += '1'

c4 = []

for i in range(0, len(str1), 8):

c4.append(int(str1[i:i + 8], 2))

final\_cipher = bytes(c4)

def rc4\_ksa(key):

"""密钥调度算法 (KSA)

得到初始置换后的S表

"""

# 种子密钥key若为字符串，则转成字节串

if isinstance(key, str):

key = key.encode()

S = list(range(256)) # 初始化S表

# 利用K表，对S表进行置换

j = 0

for i in range(256):

j = (j + S[i] + key[i % len(key)]) % 256

S[i], S[j] = S[j], S[i] # 置换

return S

def rc4\_prga(S, text):

"""伪随机生成算法 (PRGA)

利用S产生伪随机字节流,

将伪随机字节流与明文或密文进行异或,完成加密或解密操作

"""

# 待处理文本text若为字符串，则转成字节串

if isinstance(text, str):

text = text.encode()

i = j = 0

result = []

count=0

for byte in text:

i = (i + 1) % 256

j = (j + S[i]) % 256

S[i], S[j] = S[j], S[i] # 置换

t = (S[i] + S[j]) % 256

k = S[t] # 得到密钥字k

# 将明文或密文与k进行异或,得到处理结果

result.append(byte ^ k)

return bytes(result)

S = rc4\_ksa('ExpectoPatronum')

res = rc4\_prga(S, final\_cipher)

flag = "ISCC{" + res.decode() +"}"

print(flag)