HW1_Report

R05922164 柯百嶽

此份作業與 P06922004 洪軒治　R06922103 鍾明諺　討論過

Handwriting

1. CIA
    - confidentiality

      保密性:要保證未認證用戶不會得到我們不想給的訊息

      example:媽媽傳密碼給我，叫我幫它設定東西，如果透過網路傳輸。

    - integrity

      完整性:要確認我們的訊息在傳輸途中不能被變更 example:要跟別人告白的時候，如果訊息被改掉，只能哭哭了。

    - availability

      可用性:可以確保訊息可以完整的傳送到目的地 example:定機票的時候，訪問網站，但最後付款的 訊息無法傳送的話，就無法定機票了。

2. Hash Function
    - one-wayness

      很難從結果推出hash前的值

    - weak collision resistance

      給定a，不能容易找到b!=a使得h(a) == h(b)

- strong collision resistance

  不能隨便找到a,b(b!=a)使得h(a) == h(b)

3. ElGamal Threshold Decryption

   將secret keyz放在Shamir’s的常數項，使用t-1次方的Shamir’s來作切割，解的的時候就需要t out of n個人來才能解密。

Capture The Flag

4. How2Crypto

   這裡面有一個確認關卡與六個傳統密碼學關卡

   Q0.

```
##################
### how2crypto ###
##################

: What? You want the flag?
: Too bad, I seperate the flag into 6 pieces.
: I heard you are one of the CNS student.
: Try to collect all the flag pieces yourself :)

### Warmup ###

: Huh? This is your first CTF challenge?
: Lets start with something simple.
: Please send me back the text below.

m1 = to recover the original information only
with intended
```

   簡單的把m1傳回去就能開始挑戰後面的六個關卡

Ans: `to recover the original information only with intended`

Q1.

```
### Round 1 ###

: I love alphabets.
: I love numbers too!

c1 = 1506 200805 040919030916120914051 9 1506
1301200805130120090319 0315131621200518
m1 = ?
```

大概手動試了一下，發現是每兩個數字代表英文字母的號碼，空格不變。因此只要每次讀兩個數字，並對應到小寫的就可以了

Ans: `of the disciplines of mathematics computer`

Q2.

```
### Round 2 ###

: I am pretty sure you know about this.
: Unless you felt asleep in class :P

m1 =
presence of third parties called adversaries
More generally

c1 =
rtgugpeg qh vjktf rctvkgu ecnngf cfxgtuctkgu
Oqtg igpgtcnna


c2 =
```

```
etarvqitcrja kpenwfg gngevtqpke eqoogteg
ejkrdcugf rcaogpv ectfu
```

首先發現，空格沒變，位置也沒變，並且同一個字在同一次只會變成一樣的。看起來是每次連線位移都不一樣的凱薩。因此在每次連線時，看一次位移多少，再位移回來就好了。

Ans: `cryptography include electronic commerce chipbased payment cards`

Q3.

```
### Round 3 ###

: Too easy for you...
: This round is still for beginner.

c1 = ohscdc kd dro sxdobcomdsyx yp dro
nscmszvsxoc yp
m1 = ?
```

這次只有給一串密文，但一樣的，空格位置沒變，因次在各種猜測與嘗試後，憤怒的把0~25位移都印出來看，發現會有一個很像英文，最後用detect_langs的套件看看像不像英文

Ans: `exists at the intersection of the disciplines of`

Q4.

```
### Round 4 ###

: BALSN{...
: Nope, not so fast :P
: Getting harder now...
```

```
m1 =
thereby precluding unwanted persons from

c1 =
igzmzsl qmzdytoruv tuphuizo qzmxfux kmfj


c2 =
ehmrftx hxqzdix ru rukfmjhirfu xzdtmril xtdg hx
m2 = ?
```

首先空格沒變，同一個字每次也對應到一個一樣的東西，先試一下固定位移，發現沒有用，只好對對應建表，但每次總會少幾個字。經過一番觀察(google)之後，在變換對應的古典密碼學發現仿射密碼(affine cipher)，試了一下，發現就是使用這個cipher，於是得到答案。

Ans : `various aspects in information security such as`

Q5.

```
### Round 5 ###

: You are so good...
: I bet you can't pass this!

m1 =
original information only with intended
recipients the

c1 =
o  oeicrdien rwtnnoim nnaiiihltnparfdlngteys
iteoe tih


c2 =
```

```
i dsiiteiocmanaautticdnoatfstcyyaue ntao ah
lrnsfi rd
```

首先觀察，這次空格的順序變了，應該不是之前那種。而且第一個字一定會一樣。經過一番觀察發現每次會在位置上位移d倍 大概是c[i] = m[(i * d)%len(m)] 因此得到答案

Ans: `information security such as data confidentiality data`

Q6.

```
    ### Round 6 ###

: Huh? You are still here...
: You must be lost...
m1 =
with intended recipients thereby precluding
c1 =
weiiindntptughisl c cietenrhrt epedr neeydb

c2 =
crehadisicscp  nbtdeanirsegremiudytc aa pl
```

看了一下完全沒有頭緒，只好打開wiki的classical cipher看一下類型，看了一看很像把明文橫著放，一定長度換一行，最後直的讀，但前面是對的，到第一直行後就會錯，後來發現除了第一行之後要第一行跟最後一行來回讀，因此先算出一行有幾個字，再依照上面規則，最後得到答案

Ans: `'chipbased payment cards digital currencies`

在每一個關卡會得到一段字串，結合起來是下面這段

`iVBORw0KGgoAAAANSUhEUgAAAS4AAAAeCAIAAAAdL2QNAAAFA0lEQVR4Xu2aPZbTMBCAx5zFoeBxguQECQ0VLZ1TJk26Lelo4nLT0VLRYJ9gcwIexdp3CYV+LI1GI9lOQuDNV21kaf40I8nyAgiCIAiCIAiCIAiCIA`

iCIAiCIAjXZHnsLpfL5dIdl6mugvAA5GXsG+bZQ7I8ftuVqU6CYNB1
4JRBTstNKHffGA3RUqyaC6Kpgk6m3MOHejgxBkkeeth2LyKqdej1/m
0JAH29KhZ70OodLVrIrYPK+n5dBj13UDabMG3uMRc3wboyKuTUqPN+
UazqHgDKt+/Z0RQoAWij3E4o4JFSpMSaPs4sBrWFCxaNMcqXYW3eCs
b360HOwvW1MevmWP5uKVKO6Bg6bTktE+c3WRFMjBMH1L5eFUVR6JqG
9UfPyU8fSoB2u20BoPzwKW2vGmGlWrkOfd8DrA+Zzp++qMVGKbfiv5
xSA+cy3vfx2LP4EK9i26ZGPQTttrAs9udU98ejetqVAH1dj5rfaaMS
oNMzdZhWbd1xqQueOFfGtlFybTDLBrXvxXZFTxOl01+lw9cDLDhzAe
R8rxrV4JvjbXBZWozp9EKKtTSB846H6s+mcq3QNgQbr5UQDR1HbINF
CeREjbHNH4sfoQh0XcSRcDtKtwz2ERtmFHZUcleMQRyN0GQM2UilcW
RKPLHoobXVlYw8iC8JZirCakP4pTGtFFnf/RS+NNW0Yxu7bBFacMRd
B+lIuEnjSkqFjiP01K+IMFEZ22iBfmYMrdcrRcq+5HylRsUWqSSRAM
UyN5aPhGIsGa3CTRX904wMouJMiqPQmXtPsWqYU4q878YaXBS2R56m
RAoEWlDIvV84cuh3MFl86DjipThI9ddM1jYudGEE6KxLFV7QgqXkzV
d6lDE4JQmBY4ADhhQFS3i8FMEfgRPCzZ3OHgD5UiSfEBa4TTNKMeF7
IChMz8AygiCkPoS5rkv+48BZPzp0FkVDx8H3GyIxGM7axoWOiEC8FM
Nu0ZZoDbHOZ4xKLGi53xXP+68tDLex1cc1AEC5e1GxedHf+vx7HY7z
fmEuIYgb3vP+c91DuXuqvMZFsW2pzzPn3x0AAHS/73BDMN/3LIxP+W
LP33/2ALA+HKvjYQ0A/c/vd4hHPst3C/PnlFv9iZw2+O6Ibzlt8GWT
aik23GVgcpS+hOvrVeQeK7cUTQL2r7+GHwR84lSNt7Y4UxOgan99OP
hdTj/azIn89aoufe1l7FLlJ7Q/nJAae83DNON915Y496DpmbWuAqyf
/R0jvnHrWix3z7uSqkQbCzIUA3mhG4vJxbbtsVdR2yaGziXcdXNaro
/6IN5+HX2fzL4rkmc5/0BFnCyaihbqn3nDAPmtQS+nlTqZxLRFPEwe
UNO+xw9OHjnzznhA2uGNoCPpYnp4jx0fEMnYsJ46ButOSl7CtiyBht
ARpCy75frQiesQ2xXN8cihr1dqOVL7Al5z9ZLM7Yvn/QJ9SFT/NRNZ
J9QpdTqnDf5u2W4Hbb70dpv3yW6a76dNnnTMaRN8SWy37Hc6s5WSO5
j+DGp+2M3FDYU69vChm0DVPK/tpmC+BrvvHzHbRoUudMRurM67S06L
kCayGwga8rbnHgewqTyybVck7Wb2u+KjoBawcvdyu3IkT2eXMMFncw
NF+h+O5r3VcdzA5n+Gqb4vj52+3DNbNcU/V4pzT63/NzevRGEGff15
xhlfEARBEARBEARBEARBEATHr/IHj+oKZBQsYE0AAAAASUVORK
5CYII=

用base64解密後會看到有PNG的header 因此存成檔案後可以看到下圖

BALSN{You_Are_Crypto_Expert!!!^_^}

FLAG = BALSN{You_Are_Crypto_Expert!!!^_^}

5. Mersenne RSA 在mersenne-rsa.txt裡面

```
N =
364615485029501136970713101143871109540079913994
317049087258562868354903436255206595580958951461
147024129894416770392933752888490885711614193520
646632973108751496411205454301933653621610762952
359760633015466919606414418247273955697450246240
243890311584572563094642894376854071409826472706
802673042403357882788691676170142926495057389918
6177
e = 65537
flag =
581788861819028849604486057941619390966415254053
077878653962134906354590013338110809005469428056
022879067145295587559787690231746970639508058101
186849163634545251666072807237318293232040804252
379754066724988796875005542128501130627797311445
331915041017484093388581376972683044166349427306
646525252846187992860040468618963138756087975061
397
```

首先把N拿去找p,q，我是先放到factordb上面找，找到是(2**521)-1跟(2**607)-1

我們只要先找出mod運算的反元素，再算出pow(c,d,N)，最後把數字轉成hex再轉成字元

FLAG : BALSN{if_N_is_factorized_you_get_the_private_key}

6. OTP 題目提示說明文是英文，因此我用xortool來解 首先用`xortool -x -c ' ' otp.txt`

告訴它常用的字元是空白這個字元 它會產生可能的明文 發現裡面藏了一個FLAG

FLAG : BALSN{NeVer_U5e_0ne_7ime_PAd_7wIcE}

7. Double AES 使用meat in the middle攻擊，因為依照code描述，其中前後兩個key的範圍只在0~2**23以內。　因此分別從前面與後面爆破所有可能並尋找解出來的中間的是否一樣。這裡存key與中間文字的字典並比對，大概花費一分鐘可以得到key，得到key即可以解出FLAG

   FLAG :BALSN{so_2DES_is_not_used_today}

8. Time Machine

   看程式，它有兩關，第一關先叫我傳一個sha1的hash以某個6個特定的東西結尾。第二關叫你傳兩個不一樣的東西，但要跟前面的hash一樣。 查了一下sha1的碰撞，在google 的demo有提供兩個pdf，稍微看了一下他的paper，只要拿前面320byte，後面append一樣的東西，hash值也會一樣

   因此把兩個pdf前面byte拿出來，然後隨機append東西在後面，一直試到符合它給的後面六個一樣，傳過去過第一關

   再來第二關就傳兩個append一樣東西的pdf過去就可以得到flag了

   FLAG: BALSN{P0W_1s_4_w4st3_0f_t1m3_4nd_3n3rgy}

9. Future Oracle 首先會有一堆驗證，回傳一個隨機數字跟mac

```
cipher = sha256(password + '||' + ID + '||' + Nc
+ '||' + "login")
print Ns + '||' + base64.b64encode(cipher)
```

   回傳的東西就是上面的

   接下來會驗證`sha256(password+"||"+ID+"||"+Ns+"||"+action)`
   因此先開一個連線，身份打admin拿到隨機數字，再開一個連線，

message部份(Nc)打剛剛的隨機數字，這樣它就會幫我們弄好mac了

如此一來可以得到 `Welcome! admin`

但還是無法拿到flag，因為這樣後面的action會是login

後來看到code

```
if ID == 'admin' and plaintext[-1] ==
"printflag":
        print 'The flag is: ' + secret.FLAG
```

在判斷action的地方是用split後取第 -1 個參數，這樣不就可以用append多一個東西嗎？

進而聯想到上課老師說過hash在作mac的時候，沒弄好可以用length extended attack

在上面用第二個連線拿到的
`sha256(password+"||"+ID+"||"+Ns+"||"+action)`

可以append東西在後面，這樣可以造出第-1個參數是pritflag的情況 而且提示密碼在30個字以內，而長度擴展攻擊是要知道key的長度，也就是password的長度

網路上找了一個hashpump的工具把密碼長度從0~30爆破一次，看哪一次會過

最後算出長度是21

FLAG : BALSN{Wh4t_1f_u_cou1d_s33_th3_futur3}

10. Digital Saving Account

連過去後，可以註冊或登入，但登入後只有admin可以繼續下去

因此稍微看了一下它處理的方式

它會把`pt = 'login=' + name + '&role=guest' + '&pwd=' +pwd`的地方拿去AES_ECB加密，blocksize是16

因此它每一段的block是可以分開計算的，所以可以用它提示的cut-and-paste attack

我們先構造兩個name跟admin

```
login=1234&role= guest&pwd=admin
==>3NwGVOE8Rr1ynlGg9zYE014vlXQjurxhwztPKYouODA=
login=1234567890 admin&role=gues t&pwd=admin
==>+KQeLB+tFVt7AZCiycWhkXkRlV9ovsoJoAx6qJaTYrfYz
AWC11hXHQSY/mB0yXeb

login=1234&role= admin&role=gues t&pwd=admin
==>3NwGVOE8Rr1ynlGg9zYE03kRlV9ovsoJoAx6qJaTYrfYz
AWC11hXHQSY/mB0yXeb
```

把第一個的第一個block跟第二個的剩下block接在一起

由於他在判斷角色的時候有點模糊

```
if 'admin' in data['role']:
```

只判斷有沒有admin而已，所以會過

之後會跑dsa，也就是數位簽章，它會給我們兩筆明文與這個明文簽過得樣子還有他的public key。

首先不知道它數位簽章用什麼hash fuction，因此寫一個驗證知道他是使用sha1

再來觀察兩筆簽名，它的r都一樣，表示兩筆共用k

如此一來可以用兩筆s計算的公式相減，先得到k

再來依照計算r的方式，可以得到x，也就是密鑰

這樣一來可以傳送sign("FLAG")

得到flag

FLAG: BALSN{s3nd_m3_s0m3_b1tc01n_p13as3}