CNS HW2 REPORT

Question 1. Encryption Algorithm

	Symmetric	**Asymmetric**
Time	Fast	Slow
Key size	Small	Big
example	RC4, DH	RSA, ElGamal
mechanism	share same key	public key, private key
Digital Signature	Yes	No

速度，Assymetric Cryptographic 因為需要大量次方運算與求餘數，所以較耗時。

KEY SIZE，因為Assymetric Cryptographic如RSA需要兩個非常大的質數來確保不易被分解，而Symmetric有許多擴充方法，以達到類似Perfect XOR的效果。

Question 2. Three-way Diffie-Hellman

A cyclic group with generator g

1. A, B, C each generate their private keys x_A, x_B, x_C
2. A, B, C each calculate $y_A = g^{x_A}, y_B = g^{x_B}, y_C = g^{x_C}$
3. A sends y_A to B, B sends y_B to C, C sends y_C to A.
4. A calculates $z_{CA} = y_C^{x_A}$, B calculates $z_{AB} = y_A^{x_B}$, C calculates $z_{BC} = y_B^{x_C}$.
5. A sends z_{CA} to B, B sends z_{AB} to C, C sends z_{BC} to A.
6. A calculates $k_{BCA} = z_{BC}^{x_A}$, B calculates $k_{CAB} = z_{CA}^{x_B}$, C calculates $k_{ABC} = z_{AB}^{x_C}$.

   It's trivial that $k_{BCA} = k_{ABC} = k_{CAB}$ and it's the shared key

---

Question 3. ElGamal threshold decryption

KeyGen

- Choose a prime p such that $p = 2q + 1$, q is also a prime
- Find a generator g of order q
- Choose a random $a \in Z_q$ and compute $\beta = g^a$, $y = \beta^x$
- Compute a random degree t - 1 polynomial $f(x) = a + \sum_{i=1}^{t-1} \alpha_i x^i \ mod \ p$
- Compute n shares of a: $s_i = f(x_i)$ for each user i

public key $pk = (p, g, \beta)$, private key $sk = x$

Encrypt

- Choose a random $k \in Z_q$ and compute $c_1 = g^k \ mod \ p$
- Compute $c_2 = m\beta^k \ mod \ p$

The ciphertext is $c = (c_1, c_2) = (g^k, m\beta^k)$

Decrypt

$d_i = c_1^{s_i} = (g^k)^{s_i} \ mod \ p$ for each user i

$$d = \prod_{i \in I} d_i^{Y_i} \equiv \prod_{i \in I} g^{kf(x_i)Y_i} \equiv g^{k\sum_{i \in I} f(x_i)Y_i} \equiv g^{kf(0)} \equiv g^{ka} \ mod \ p$$

$$m = c_2 d^{-1} = (m\beta^k)(g^{ka})^{-1} = mg^{ka}g^{-ka} = m \ mod \ p$$

Question 4. ECB Encryption

> BALSN{W0w_y0u*4r3*r3411y*4*cu7*4nd_p4st3*m4st3r}

Solution

因為系統是採用ECB加密，也就是以同一組Key去加密所有東西，因此，我們可以透過多次詢問去拼貼出想要的buffer。

先送`username = "bals", password="gg3be"`，得到`login=bals&role=user&pwd=gg3be`的encode，然後我們只取前十六個也就是`login=bals&role=`。接下來，我們送`username="balsnbalsnadmin", password="gg3be"`，取第十七個字以後的全部，得到`admin&role=user&pwd=gg3be`，跟前面的合起來，則可與`username="blas", password="gg3be"`一起submit。

Question 5. Beginner's RSA

Part 1

> BALSN{V3RYW311}

因為加密的長度不長，因此猜測可以暴力破解。透過`RsaCtfTool`即可得解。

```
$ ./RsaCtfTool.py --publickey ./public.pem --uncipher ./flag.enc
```

Part 2

> BALSN{Forty Years of Attacks on the RSA Cryptosystem}

因為二者共用n，且我們有一組private key。假設我們的private key為d，public key為e，Alice的public key 為 e1，cipher為c。

$e * d \equiv 1 \ (mod \ \phi(n))$, $e * d - 1 = k\phi(n)$

若我們求出e1在模$k\phi(n)$空間下的反元素為b，則$c^{e*b} \equiv m^{k\phi(n)+1} \equiv m \ (mod \ n)$

Part 3

> BALSN{Keep calm and count prime numbers}

因為Alice跟Bob共用n，且我們有二者的cipher。假設Alice的cipher為c_A，public key為e_A，Bob的cipher為c_B，public key為e_B。

$c_A = m^{e_A} \ (mod\ n)$, $c_B = m^{e_B} \ (mod\ n)$

根據Extended Euclidean algorithm，我們可以求出a,b使$a * e_A + b * e_B = gcd(e_A, e_B) = 1$

則 $c_A^a * c_B^b \equiv m \ (mod\ n)$

Question 6 Digital Certificate

> BALSN{b451c_s3!f_51gn3d_c3rt1fic4t3}

簡單來說就是要自己產一個Certificate。

`openssl req -new -x509 -days 6666 -out -ca.crt`然後再base64 encode即可。

Question 7 I need your help

> BALSN{Now_you_know_the_secret}

ref: https://crypto.stackexchange.com/questions/21102/what-is-the-ssl-private-key-file-format

先將`private.key`中的`-`換成`x`decode然後dump成hex。根據標準，我們知道`02`為每個整數的開頭，並且緊接著的是代表數有多長。

得到

```
028201
c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c7
1c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c
71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71

c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c7
1c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c
71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c71c
718181
00
e06f755205d1063502ac8044127f4aeada8ac5adf8830c61f5ea0c71714c8ea1a8106a2a6949807d2d0f7dc03877f35221
b69726cdc2e6ab351152dbab547d52c1719b7e7976ab2d5ae662a2dfe2e57151ef1fcedca7e8e8947c346d32b66e764c44
93cb71f548a4ddc07b8d63e2fe8d8bfcaacbe0529a22fb8acd8f7bdfc923
028181
00
d080094320bef16c0cd45538d1136b1328a68ac990f338d3077de18e036718b39a17478496fbffd89341fb39838a00350c
436bf31a7bc073bb6cc1deeff4b0379878b543a3c6c190052489725b0246f1116152cb141bc67a17bdeffae7108dc55ea9
2f2be0fa76a4aba72c9b8b12d6a03eb55b57378706c7eeaca86268dca47b
028181
00
caacfca690a81d51dbd34995af9a925e19f33de70847d7f3d2ee844421cbbff64e5e5c7166594116498df6c2927c0818c0
673282914813a4c2ac9d45d0a7e0f0cdce395c7275eec96b9027bcec2feb81753b5b5f24b6e146bf6896b3921b5b0fab7a
36797fb4c0e0597d0c5637c0f29d82b02ed124079438492e24ca11549b3f
028180
0a03dc6e0908a2f819b5a9524d58ad700227ddcac8d7a6071cf902f89b593c6a84205223204d828098b236ab1092746817
eab528bc40ed81a1a31bbce5b1cf351c71cfe32bdec43572c9ca805fb6c0499c181cadfc8d48ff5c5c97466a0af5846183
a6ec68a61f44d1a9fa0e8ea39d039d7f809fb3df1c884602ad23ecfd8c39

028181
00
9862c02f29dd2c9a5a03e117f9c83f1d6eeb475b88aefe5d42a301b5273c78ba583830fad22db60ef7164742887baae915
ad97a878f6cea9fb58bc8c39e4e9c184ddd30a0be27281d72e09fc0e141fe72f0838e121579426fc4a3be76e23bfbf8df6
4c91ed53aa320a7ab03e7d80c373ed1cf387e040c1474f42e60db24cf230
```

其中，718181那組為猜測，因為如果長度大於`0xFF`需要先在前面加個`81`，因此`81`也是個重要的指標，而`71`為x所產生出的結果，所以有可能是`02`。

結果，那個欄位的確是`02`。

因此我們有`p`跟`q`，先利用套件`rsatool`重新製造出`private.pem`

```
./rsatool.py -p
0xe06f755205d1063502ac8044127f4aeada8ac5adf8830c61f5ea0c71714c8ea1a8106a2a6949807d2d0f7dc03877f352
21b69726cdc2e6ab351152dbab547d52c1719b7e7976ab2d5ae662a2dfe2e57151ef1fcedca7e8e8947c346d32b66e764c
4493cb71f548a4ddc07b8d63e2fe8d8bfcaacbe0529a22fb8acd8f7bdfc923 -q
0xd080094320bef16c0cd45538d1136b1328a68ac990f338d3077de18e036718b39a17478496fbffd89341fb39838a0035
0c436bf31a7bc073bb6cc1deeff4b0379878b543a3c6c190052489725b0246f1116152cb141bc67a17bdeffae7108dc55e
a92f2be0fa76a4aba72c9b8b12d6a03eb55b57378706c7eeaca86268dca47b -o private.pem
```

然後透過openssl解密，即得到flag。

`openssl rsautl -decrypt -inkey private.pem -in flag.enc -out flag.bin`

Question 8 I will look for you, and I will find you

Part1

> BALSN{Don't underestimate the power of the Dark Web}

Download tor browser and connect to it.

Part2

> BALSN{128.199.198.162}

因為使用https，所以有certificate，然後發現email為`balsn@servers.tw`用whois看一下擁有者，是曾助教的名字。因此`dig servers.tw`即得到ip。

Part3

> BALSN{140.112.31.96}

這題的關鍵是SSH的fingerprint。

首先`torsocks ssh-keyscan ztczadd4tipwhwyl.onion`找到所有可能的fingerprint。但是因為預設是SHA256，所以再用 `ssh-keygen -l -E md5 -f 檔名` `轉。最後把結果丟到shodan.io。

Part4

> BALSN{104.198.2.240}

ftp有active mode。可以讓ftp server對client連線，因此可以得到ip。

先在linux1.csie.org 用`python3 -m http.server 9487`架設一個http server用來接收資料。然後`torsocks ncat 7zysy3slgt7qxhek.onion 21`連上FTP server。

```
USER anonymous
PORT 140,112,30,32,37,15
LIST
```

然後就會得到IP

Part5 (Bonus)

因為執行`cat <(echo Knock, knock) - | torify ncat su7twnwqamobiytx.onion 31337`會得到回覆，因此目前想到的方法是暴搜全部的ip，判斷回覆。但是因為目前人住學校，如果做大規模的掃瞄會被ban，因此仍在想辦法。