Fall 2024 CIS 3362 Homework #6: Public Key Encryption Solutions

1) (10 pts) In the Diffie-Hellman Key Exchange, let the public keys be p = 67, g = 13, and the secret keys be a = 28 and b = 51, where a is Alice's secret key and b is Bob's secret key. What value does Alice send Bob? What value does Bob send Alice? What is the secret key they share? Use a program or calculator to quickly simplify the modular exponentiations that arise, but show what each calculation is.

Solution:

Alice sends Bob 13^{28} mod 67 ≣ 16
Bob sends Alice 13^{51} mod 67 ≣ 58

*Using IDLE Python IDE will make this easy

Calculate the secret keys:

16^{51} mod 67 ≣ 64
58^{28} mod 67 ≣ 64
*Remember that these should be the same

Therefore, we've calculated that the shared secret key is **64**.

2) (10 pts) In an RSA scheme, p = 31, q = 23 and e = 139. What is d? Show the work by hand, but for any complicated calculation, do it on a calculator or use a program. (So, show each step of the Extended Euclidean Algorithm, but feel free to use a calculator to quickly get quotients and remainders.)

Solution:

n = p*q = 31*23 = 713
ϕ(713) = ϕ(31)*ϕ(23) = (31-1)*(23-1) = 660

*Since p & q are prime, this computation is easy

d = 139^{-1} mod 660 *Now use EEA to solve this equation

660 = 139(4) + 104
139 = 104(1) + 35
104 = 35(2) + 34
35 = 34(1) + 1 *End of Euclid's Algorithm

35 – 1(34) = 1
35 – 1(104 – 2(35)) = 1
 35 – 104 + 2(35) = 1
3(35) – 104 = 1
 3(139 – 104) – 104 = 1
 3(139) – 3(104) – 104 = 1
3(139) – 4(104) = 1
3(139) – 4(660 – 4(139)) = 1
 3(139) – 4(660) + 16(139) = 1
19(139) - 4(660) = 1 *End of EEA

19(139) - ~~4(660)~~ ≣ 1 mod 660
139^{-1} ≣ 19 mod 660

So **d = 19**.

3) (50 pts) The following message was encrypted via RSA encryption. The public keys are as follows:

n = 576025912082114341909169
e = 395065083027011624330977

Each integer in the ciphertext corresponds to a plaintext of 16 letters. This ciphertext was generated by the program rsa3.py. You may use any of the posted code as necessary to decrypt the message. In your write up, describe in detail what steps you took and which code (if you used any of the posted code) you used, or how you adapted it. Turn in attachments of any original code you wrote or anything that had non-trivial adaptations of the code posted for the class. Here is the ciphertext:

```
488798928261625380184161
533946500611718831345802
411942882720703143384960
20068354290376977207914
252864055600177840617225
144565738643838496733483
98121155489099542089269
377474600037914621137040
```

Solution:

In order to solve this we need to find some way to break `n` into its component parts of `p` and `q`. However standard factorization methods won't cut it. Luckily, since `n` is small enough, we can use Pollards Rho factoring to factorize `n`

n = 832176222161 X 692192226529

Now that we know `p` and `q` we can use it find our decryption key `d`. After this we can use that to decrypt the cipher text.
We will use `rsabreak.py` to pass in the known information. Basically, we will take the known information: n, e, p, q, $\phi(n)$ to apply the computation of C^d mod n to decrypt each block of cipher text.

Once decrypted, we will convert each block back to english by reversing the base-26 character encoding.

Doing this we get:

GOTOHECTWOFOURSIXTHEREISAKNIGHTTHEREITSARMISHOLLOWLOOKINSIDEOFITSRIGHTFOREARMTOGETANOTEBRINGTHENOTETOMEFORYOURPRIZEXXXXXXXXXXXXX

Cleaned up we get:

GO TO HEC TWO FOUR SIX THERE IS A KNIGHT THERE ITS ARM IS HOLLOW LOOK INSIDE OF ITS RIGHT FOREARM TO GET A NOTE BRING THE NOTE TO ME FOR YOUR PRIZE

Refer to pollardrho.py & rsabreak.py

4) (30 pts) The following ciphertext below was created with the El Gamal cryptosystem with the following public elements:

q = 310000037 (prime)
g = 52216224 (primitive root)
Ya = 32298658 (Alice's public exponent)

You also know that the plaintext was written in all lowercase letters and split into blocks of 6 characters and the value of each 6 character block is simply equal to its base 26 equivalent, treating A = 0, B = 1, …, Z = 25.

Use this information to decrypt the following ciphertext: (Note: This will also be given to you in a text file, for ease of processing and as mentioned, each line represents the encryption of one block of 6 characters.)

```
56495539 72767212
62083516 76971521
181398440 263421160
149867850 72743477
14826439 190288780
113953407 197793189
117331466 185360595
291767686 140312582
97578813 288144131
66782213 277003739
189849901 192777619
147582903 21503450
154299245 242826784
86211909 200694188
31309028 293758361
21217580 3535169
79019712 49185229
213930082 159557439
73624006 229408211
292736574 18644176
237123292 168250610
38995570 306955959
199390530 176530325
226189829 196581913
195038651 170658203
```

Good Luck!

Solution:

To solve this we essentially need to solve the Discrete Log problem, which can be hard for very large numbers. Luckily the ‘Baby Step, Giant Step’ algorithm will help us solve this problem and find X_A

X_A=87543455

From here we can solve for the plain text.

Remember that just like the last problem, we still need to convert the decrypted plain text from base 26 to English letter.

Running ‘elgamaldec.py’ we get on the given information:

THEREISAHISTORYINSTALLATIONINTHELOBBYOFHECONEOFTHEPANELSDISCUSSESACRYPTOGRAPHERFROMWWIIGOTHEREANDBEHINDHISPANELFINDANOTEIHIDBRINGITTOMEFORSOMEMONEYXXX

Cleaned up:

THERE IS A HISTORY INSTALLATION IN THE LOBBY OF HEC ONE OF THE PANELS DISCUSSES A CRYPTOGRAPHER FROM WWII GO THERE AND BEHIND HIS PANEL FIND A NOTE I HID BRING IT TO ME FOR SOME MONEY

Refer to ‘elgamaldec.py’