Title: CSCI 55500 Homework #3 report
Name: Zhihao Cao
Data: 10/20/2016
Instructor: Xukai Zou
Programming language: Java
Development software: Eclipse

1. **[10 points]** Without using a computer, calculate $24^{66,000,000,023}$ (mod 77). *Hint*: Using Chinese Remainder Theorem.

$$77 = 11 \times 7$$

We find a way to apply Fermat's Little Theorem to simplify the problem:

$$24^{66,000,000,023} = 24^{(10)\times 6,600,000,002} \times 24^3$$

$$24^{66,000,000,023} = 24^{(6)\times 11,000,000,003} \times 24^5$$

By applying Fermat's Little Theorem:

$$24^{10}\, mod\ 11 = 1$$

$$24^6\, mod\ 7 = 1$$

Therefore:

$$24^{(10)\times 6,600,000,002} \times 24^3 = 24^3 \equiv 8\, mod\ 11$$

$$24^{(6)\times 11,000,000,003} \times 24^5 = 24^5 \equiv 5\, mod\ 7$$

Then we use Chinese Remainder Theorem:

$$M = 77, m_1 = 11, m_2 = 7$$

$$M_1 = 7, M_2 = 11$$

$$a_1 = 8, a_2 = 5$$

By Extended Euclidean Algorithm:

$$y_1 = {M_1}^{-1}\, mod\ m_1 = 8$$

$$y_2 = {M_2}^{-1}\, mod\ m_2 = 2$$

$$\sum (a_i M_i y_i)\, mod\ M$$

$$= (8 \times 7 \times 8) + (5 \times 11 \times 2)\, mod\ 77$$

$$= 558\, mod\ 77 = 19$$

2. ElGamal

First, load input file into arraylist.

Then, do x = y2(y1^a)^-1 mod p for each pair of y.

Then, the decrypted text should be decoded back to alphabet letters.

```
//decode
BigInteger ts = new BigInteger("26");
BigInteger a = x_.divide(ts.pow(2)).mod(ts);
BigInteger b = x_.mod(ts.pow(2)).divide(ts);
BigInteger c = x_.mod(ts.pow(2)).mod(ts);
```

Then, use a, b, and c index values to lookup English letter array to get the plaintext.

3. (a)

According to Algorithm 5.16, we have:

$$c_1 = {b_1}^{-1} \; mod \; b_2$$

$$c_2 = \frac{c_1 b_1 - 1}{b_2}$$

$$x_1 = {y_1}^{c_1} ({y_2}^{c_2})^{-1} \; mod \; n$$

Since $y_1 = x^{b_1} \; mod \; n, and \; y_2 = x^{b_2} \; mod \; n$

We combine these equations:

$$x_1 = x^{b_1 c_1} (x^{b_2 c_2})^{-1} \; mod \; n$$

$$x_1 = x^{b_1 c_1 - b_2 c_2} \; mod \; n$$

$$x_1 = x^{(b_1({b_1}^{-1}) mod \; b_2) - (b_2 \frac{({b_1}^{-1} \; mod \; b_2) b_1 - 1}{b_2})} \; mod \; n$$

$$x_1 = x^{(1) - (({b_1}^{-1} \; mod \; b_2) b_1 - 1)} \; mod \; n$$

$$x_1 = x^{({b_1}^{-1} \; mod \; b_2) b_1} \; mod \; n$$

$$x_1 = x$$

(b)

n = 18721

b_1 = 43

b_2 = 7717

y_1 = 12677

y_2 = 14702

According to Algorithm 5.16, we have:

$$x_1 = {y_1}^{c_1} ({y_2}^{c_2})^{-1} \; mod \; n$$

$$x_1 = y_1^{(b_1^{-1}\ mod\ b_2)}(y_2^{\frac{(b_1^{-1}\ mod\ b_2)b_1-1}{b_2}})^{-1}\ mod\ n$$
$$x_1 = 15001$$

The solution is calculated by program.

4.

a) How many points are over E.
1008

b) What is the lexically largest point over E. Here lexically larger point means to order the points by the first coordination first and then the second coordination.
(1038,1037)

c) Does point (1014, 291) belong to E?
No, the closest one is (1014, 290)

d) Suppose alpha=(799,790) is a generator and beta=(385,749). (E, alpha, beta) is the ElGamal public key. Given the plaintext value (575,419) and random K=100, what is the ciphertext value? Given the ciphertext value ((873,233), (234,14)), what is the plaintext value.
Ciphertext:
y1 (873,233)
y2 (963,817)
Plaintext:
(319,784)

e) Suppose E and a generator alpha=(818,121) are public. Alice and Bob achieve a shared secret by doing Diffie-hellman key exchange. Alice sends Bob a value (199,72), and Bob sends Alice a value (815,519), what is the secret they achieve?
share_key = (191,568)

5.

a) None.

XOR operation cannot guarantee the confidentiality.

No hash of M to guarantee integrity.

No signature or mutual known keys to guarantee authentication.

No signature to guarantee Non-Repudiation.

b) C, I

Encryption using k_1 guarantee confidentiality.

Hash guarantee the integrity.

K_2 inside the hash and with k_1, mutual keys guarantee authentication.

No signature to guarantee Non-Repudiation.

c) C, I, A, NR

Encryption using R_{pub} key guarantee confidentiality.

Signature guarantee the authentication and Non-Repudiation.

Hash guarantee the integrity

d) C, A, NR

Encryption using R_{pub} and S_k guarantee confidentiality.

Signature guarantee the authentication and non-repudiation.

No hash to guarantee the integrity.

6.

$$\delta = k^{-1}(m - a\gamma)\ mod\ (p-1)$$
$$\delta k = m - a\gamma\ mod\ (p-1)$$
$$a\gamma = m - \delta k\ mod\ (p-1)$$
$$a = (m - \delta k)\gamma^{-1}\ mod\ (p-1)$$
$$a = (m - 0)\gamma^{-1}\ mod\ (p-1)$$
$$a = m\gamma^{-1}\ mod\ (p-1)$$

We know γ and p.

Therefore, “a” is no longer a Discrete Logarithm problem. It would be easy for attacker to compute “a”.