Eder Mazariegos (AKA Class Loner ☹)

11/12/2017

CIS3362

**Homework 6**

1. **What is the prime factorization of 589449600?**

589,449,600/2 = 294,724,800

294,724,800/2 = 147,362,400

147,362,400/2 = 73,681,200

73,681,200/2 = 36,840,600

36,840,600/2 = 18,420,300

18,420,300 = 9,210,150

9,210,150/2 = 4,605,075

4,605,075,3 = 1,535,025

1,535,025/3 = 511,675

511,675/5 = 102,335

102,335/5 = 20,467

20,467/97 = 211

211/211 = 1

2^ + 3^2 + 5^2 + 97 + 211

1. **What is φ(589449600)?**

Phi(n) = n Pi(1-(1/p))

Phi(589,449,600) = 589,449,600 π (1-(1/2))(1-(1/3))(1-(1/5))(1-(1/97))(1-(1/211))

= 154,828,800

1. **Using Fermat’s Theorem, determine 345625190 mod 2099.**

3456^25190 MOD 2099

a^n-1 = 1 MOD n

a^2099-1 = 1 MOD 2099

a^2098\*k 🡪 25190/2098 = 12.0067 🡪25190 = 2098 \* 14

= 3456^2098\*12(1 by Fermat’s theorem) 3456^14 MOD 2099

= (1) 3456^14 MOD 2099

= 3456 = 1357 (MOD 2099)

1357^2 = 626 (MOD 2099)

1357^4 = 626^2 = 1462 (MOD 2099)

1357^8 = 1462^2 = 662 (MOD 2099)

1357^14 = 662 \* 1462 \* 626 = 291 (MOD 2099)

1. **Using Euler’s Theorem, determine 266051 mod 2664.**

13^6051 MOD 2664

n = 2664/2 = 1332

1332/2 = 666

666/2 = 333

333/3 = 111

111/3 = 37

37/37 = 1

p = 2,3,37

n = 2664

Phi(n) = n π (1-(1/p))

=2664 (1-(1/2))(1-(1/3))(1-(1/37))

= 864

.

.

6051 MOD Phi(n)

= 6051 MOD 864

= 3 MOD 864

.

.

13^3 MOD 864

= 297 MOD 864 = 297

1. **In an RSA scheme, p = 13, q = 31 and e = 127. What is d?**

Phi(n) = (P-1)(Q-1)

D = e^-1 MOD Phi(n)

Phi(n) = (12)(30)

= 360

d = 127^-1 MOD 360

360 = 127 \* 2 + 106

127 = 106 \*1 +21

106 = 21 \* 5 + 1

1 = 106 – (21 \* 5)

1 = 106 – [127 - 106]5

1 = -127(5) + [360 – 127(2)](6)

1 = (-17)(127) + 360(6)

Therefore, 1 MOD 360 = (-17)(127) 🡪360-17 = 343

1. **One of the primitive roots (also called generators) mod 29 is 2. There are 11 other primitive roots mod 29. One way to list these is 2a1 mod 29, 2a2 mod 29, ... 2a12 mod 29, where 0 < a1 < a2 < ... < a12. (Note: it’s fairly easy to see that a1 = 1, since 2 is a primitive root.) Find the values of a10, a11 and a12 and the corresponding values 2a10 mod 29, 2a11 mod 29, and 2a12 mod 29.**

P MOD 29 = 2

2^1 = 2^0 \* 2 = 1 \* 2 = 2 = 2 (MOD 29)

2^2 = 2^1 \* 2 = 2 \* 2 = 4 = 4 (MOD 29)

2^3 = 2^2 \* 2 = 4 \* 2 = 8 = 8 (MOD 29)

2^4 = 2^3 \* 2 = 8 \* 2 = 16 = 16 (MOD 29)

2^5 = 2^4 \* 2 = 16 \* 2 = 32 = 3 (MOD 29)

2^6 = 2^5 \* 2 = 3 \* 2 = 6 = 6 (MOD 29)

2^7 = 2^6 \* 2 = 6 \* 2 = 12 = 12 (MOD 29)

2^8 = 2^7 \* 2 = 12 \* 2 = 24 = 24 (MOD 29)

2^9 = 2^8 \* 2 = 24 \* 2 = 48 = 19 (MOD 29)

2^10 = 2^9 \* 2 = 19 \* 2 = 38 = 9 (MOD 29)

2^11 = 2^10 \* 2 = 9 \* 2 = 18 = 18 (MOD 29)

2^12 = 2^11 \* 2 = 18 \* 2 = 36 = 7 (MOD 29)

1. **In the Diffie-Hellman Key Exchange, let the public keys be p = 29, g = 19, and the secret keys be a = 11 and b = 13, 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?**

P = 29, g = 19 Secret Keys: a = 11, b =13

g^a MOD P = A M = A^b MOD P

g^b MOD P = B M = g^ab MOD P 🡨Common Shared Keys

Bob:

B = g^b MOD P

= 19^13 MOD 29

= ((19^7 MOD 29)(19^6 MOD 29)) MOD 29

= ((12)(22)) MOD 29

B = 3

Alice:

A = g^a MOD P

= 19^11 MOD P

= ((19^6 MOD 29)(19^5 MOD 29)) MOD 29

= ((22)(21)) MOD 29

= 462 MOD 29

A = 27

M = B^a MOD P

= g^ab MOD P

= 19^(13)(11) MOD 29

= 19^143 MOD 29 (Split separately into exponents that could be added and calculated)

= 15

M = A^b MOD P

= g^ab MOD P

= 19^(11)(13) MOD P

= 19^143 MOD 29 (Split separately into exponents that could be added and calculated)

= 15

1. **In El Gamal, Alice chooses YA = αXA mod q. Bob, who is sending a message, calculates a value K = YAk, where k is randomly chosen with 0 < k < q. Is it possible that for different choices of k, Bob will calculate the same value K, or will each unique value of k be guaranteed to produce a different value for K? Give a brief rationale for your answer.**

Alice chooses YA = a^XA MOD q and Bob calculates the value K = YA^k

* It Is possible for both Bob and Alice to have calculated the same value for K because is the Prime of the modulus were chosen such that A were a primitive root of q, you could receive identical values… for example

q = 11

A = 2

**Private Keys:**

Alice ‘s: XA = 5

Bob’s: XB = 8

Alice sends to Bob after choosing the secret key:

X = A^XA MOD q

= 2^5 MOD 11

= 32 MOD 11

= 10

Then, Bob chooses his secret key:

X2 = A^XB MOD q

= 2^8 MOD 11

= 256 MOD 11

= 3

Alice Calculates (Decrypt by raising to the power of Bobs Private key):

S = X2^XA MOD q

= 3^5 MOD 11

= 243 MOD 11

= 1

Bob Calculates (Decrypt by raising to the power of Alice’s Private key):

S = X^XB MOD q

= 10^8 MOD 11

= 1000000 MOD 11

= 1

Both are equal to 1 hence the value K or in this case S where Bob and Alice are sharing their public keys together can be the same when given the prime number and primitive root calculated.

1. **Program is Attached.**
2. **Program is Attached.**
3. **A primitive root, α, of a prime, p, is a value such that when you calculate the remainders of α, α2, α3, α4 , ... , αp-1, when divided by p, each number from the set {1, 2, 3, ..., p-1} shows up exactly once. Prove that a prime p has exactly φ(p-1) primitive roots. In writing your proof, you may assume that at least one primitive root of p exists. (Normally, this is the first part of the proof.) (Note: This question is difficult, so don't feel bad if you can't figure it out.)**

Let’s assume a is our Primitive Root…

Then, {1,2,3,4,5….. P-1} == {a,a^2,a^3,a^4,……a^P-1}

Where a^P-1 = 1 MOD P |a^k| can also be our Primitive Root…

So, |a^k| = P-1

And |a^k| = |a|/gcd(k, P-1) or |a^k| =|a^k|/gcd(k, P-1),

Which yields, |a^k| = P-1/gcd(k, P-1) or |a| = P-1,

If |a^k| = P-1 then,

P-1/gcd(k, P-1) = P-1,

And gcd(k, P-1) = 1 (Euler’s of Phi gcd = 1?) 🡪

The prime has exactly Phi(P-1) Primitive root such that, a^k : 1<= K <= P-1, gcd(k, P-1) = 1 for a^k and for K… which would be Phi(P-1) or K: 1<= K <= P-1, gcd(K, P-1) = 1.

1. **Alice and Bob are using Diffie-Hellman to exchange a secret key. They are using the prime number p = 1234577 and the generator g = 1225529. Alice picks a secret value a and sends ga = 654127 to Bob. Bob picks a secret value b and sends gb = 221505 to Alice. What is the secret key they share?**

P = 12345 g^a = 654127

g = 1225529 g^b = 221505

**Public Key Values**:

A = g^a MOD P

= 1225529^221505 MOD 1234577

= 399759

B = g^b MOD P

= 1225529^221505 MOD 1234577

B = 188068

Alice does…

M = B^a MOD P

= g^ba MOD P

= 1225529^(221505)(654127) MOD 1234577

= 685920 (Secret Key)

and Bob does…

M = A^b MOD P

= g^ab MOD P = 122529^(654127)(221505) MOD 1234577

= 685920 (Secret Key)

1. **Decrypt the following Message:**

20429835450828679741350 =x1

26022799626812591980567 =x2

30572114224921561344399 =x3

14180424833673414562055 =x4

19539282983393676142312 =x5

These 5 blocks of cipher text were created with a set of RSA public keys that follow:

n = 43767782750765499923141

e = 986321785648512635467

When you decrypt, you'll initially get numbers, but those numbers can be converted into blocks of 16 letters each.

* Couldn’t get my code to run correctly on this one because I had a hard time figuring out the implementation of getting the Private keys and then converting the numbers received to actual text… the Phi(n) came out to be something like 43767782750765499923141 and I knew the equation was something like x1….x5^some number given MOD n = Given value = Text, but I just couldn’t figure out how to get that by myself. So, here is the code, some of it I found on the website and other bits from other places.

import java.math.\*;

public class Q6 {

public static void main(String[] args) {

BigInteger n = new BigInteger("22564125445");

// Number is not even

BigInteger div = new BigInteger("3");

BigInteger one = new BigInteger("1");

// Dummy values

BigInteger p = new BigInteger("1");

BigInteger q = new BigInteger("1");

// Factor n.

while (true) {

// Found the divisor!

if (n.mod(div).equals(new BigInteger("0"))) {

p = div;

q = n.divide(div);

break;

}

div = div.add(new BigInteger("2"));

}

// Factorization n and phi.

BigInteger phi = (p.subtract(one)).multiply(q.subtract(one));

System.out.println("p = "+p+" and q = "+q+" phi(n)= "+phi);

// Find d.

BigInteger e = new BigInteger("923453487449");

BigInteger d = e.modInverse(phi);

System.out.println("Decryption exponent: "+d);

}

}

Scanner fin = new Scanner(new File("message.txt"));

while (fin.hasNext()) {

// Read in this block and create the BigInteger.

BigInteger cipher = new BigInteger(fin.next());

BigInteger plain = cipher.modPow(d,n);

String ans = new String;

ans = convertBack(plain, 10);

System.out.println(plain);

}

fin.close();

public static String convertBack(BigInteger m, int blocksize) {

String message = "";

BigInteger ans = zero;

for (int i=0; i<blocksize; i++) {

BigInteger leftover = m.mod(twentysix);

int leftint = leftover.intValue();

char[] tmp = new char[1];

tmp[0] = (char)('A'+leftint);

message = (new String(tmp)) + message;

m = m.divide(twentysix);

}

return message;

}