|  |  |  |  |
| --- | --- | --- | --- |
| **Ex. No. 05** | **Diffie Hellman Key Exchange** | | |
| Date of Exercise | 17 – 02 - 2015 | Date of Output Verification | 17 – 02 - 2015 |

**Question**

Alice wanted to exchange a secrete key to Bob using Diffie-Hellman Key Exchange who is at a different terminal.

**Procedure**

1. Alice and Bob agree to use a prime number *p* = 23 and base *g* = 5 (which is a [primitive root modulo](http://en.wikipedia.org/wiki/Primitive_root_modulo_n) 23).
2. Alice chooses a secret integer ***a*** = **6**, then sends Bob *A* = *g****a*** mod *p*
   * *A* = 5**6** mod 23 = 8
3. Bob chooses a secret integer ***b*** = **15**, then sends Alice *B* = *g****b*** mod *p*
   * *B* = 5**15** mod 23 = 19
4. Alice computes ***s*** = *B****a*** mod *p*
   * ***s*** = 19**6** mod 23 = **2**
5. Bob computes ***s*** = *A****b*** mod *p*
   * ***s*** = 8**15** mod 23 = **2**
6. Alice and Bob now share a secret (the number **2**).

**Program**

User A:

import java.io.\*;

import java.math.\*;

import java.net.\*;

import java.util.\*;

public class NetworkSecurityLabExp5DiffieHellmanKeyExchageA {

public static Scanner in = new Scanner(System.in);

public static BigInteger q, alpha, xa, xb, ya, yb, key;

public static long tl;

public static ArrayList al = new ArrayList();

public static void main(String[] args) throws IOException {

System.out.println("------User A------");

ServerSocket ss = new ServerSocket(3000);

System.out.println("Waiting for User B to Connect");

Socket s = ss.accept();

System.out.println("\nUser B Connected!");

InputStream is = s.getInputStream();

OutputStream os = s.getOutputStream();

PrintWriter pw = new PrintWriter(os, true);

BufferedReader br = new BufferedReader(new InputStreamReader(is));

do {

System.out.println("\nEnter Prime q");

q = in.nextBigInteger();

} while (!checkprime(q));

pw.println(q);

System.out.println("Sent q to User B");

primitiveroot(q);

pw.println(alpha);

System.out.println("Sent alpha to User B");

tl = 2 + (int) (Math.random() \* (q.longValue() - 2));

xa = BigInteger.valueOf(tl);

System.out.println("\nxa: " + xa);

ya = alpha.modPow(xa, q);

System.out.println("ya: " + ya);

pw.println(ya);

System.out.println("Sent ya to User B");

pw.println(q);

System.out.println("Waiting for User B to send yb");

yb = BigInteger.valueOf(Integer.valueOf(br.readLine()));

key = yb.modPow(xa, q);

System.out.println("Key: " + key);

}

public static void primitiveroot(BigInteger q) {

Boolean k = false;

int i = 2;

alpha = BigInteger.valueOf(i);

BigInteger bj, bk;

do {

k = false;

alpha = BigInteger.valueOf(i);

System.out.println("\n\*i: " + alpha);

for (int j = 1; j < q.intValue(); j++) {

bj = BigInteger.valueOf(j);

bk = alpha.modPow(bj, q);

System.out.println(bk);

if (al.contains(bk)) {

k = true;

break;

} else {

al.add(bk);

}

}

if (k == true) {

i++;

al.clear();

}

} while (k);

System.out.println("Alpha Selected: " + alpha);

}

public static boolean checkprime(BigInteger t) {

int r = 0;

int a =t.intValue();

if (a <= 1) {

return false;

}

for (int i = 2; i < a; i++) {

if (a % i == 0) {

return false;

}

}

return true;

}

}

User B:

import java.io.\*;

import java.math.BigInteger;

import java.net.\*;

import java.util.Scanner;

public class NetworkSecurityLabExp5DiffieHellmanKeyExchageB {

public static Scanner in = new Scanner(System.in);

public static BigInteger q, alpha, ya, yb, xb, key;

public static long tl;

public static void main(String[] args) throws IOException {

System.out.println("------User B------");

Socket s = new Socket("127.0.0.1", 3000);

System.out.println("\nConnected to User A\n");

InputStream is = s.getInputStream();

OutputStream os = s.getOutputStream();

PrintWriter pw = new PrintWriter(os, true);

BufferedReader brs = new BufferedReader(new InputStreamReader(is));

System.out.println("\nWaiting for User A to Send q");

q = BigInteger.valueOf(Integer.valueOf(brs.readLine()));

System.out.println("q Received: " + q);

System.out.println("\nWaiting for User A to Send alpha");

alpha = BigInteger.valueOf(Integer.valueOf(brs.readLine()));

System.out.println("alpha Received: " + alpha);

System.out.println("\nWaiting for User A to Send ya");

ya = BigInteger.valueOf(Integer.valueOf(brs.readLine()));

System.out.println("ya Received: " + ya);

tl = 2 + (int) (Math.random() \* (q.longValue() - 2));

xb = BigInteger.valueOf(tl);

System.out.println("\nxb: " + xb);

yb = alpha.modPow(xb, q);

System.out.println("yb: " + yb);

pw.println(yb);

System.out.println("Sent yb to User A");

key = ya.modPow(xb, q);

System.out.println("Key: " + key);

}

}

**Input**

Q:31

**Output**

User A:

------User A------

Waiting for User B to Connect

User B Connected!

Enter Prime q

31

Sent q to User B

\*i: 2

2

4

8

16

1

2

\*i: 3

3

9

27

19

26

16

17

20

29

25

13

8

24

10

30

28

22

4

12

5

15

14

11

2

6

18

23

7

21

1

Alpha Selected: 3

Sent alpha to User B

xa: 29

ya: 21

Sent ya to User B

Waiting for User B to send yb

Key: 11

BUILD SUCCESSFUL (total time: 5 seconds)

User B:

run:

------User B------

Connected to User A

Waiting for User A to Send q

q Received: 31

Waiting for User A to Send alpha

alpha Received: 3

Waiting for User A to Send ya

ya Received: 21

xb: 7

yb: 17

Sent yb to User A

Key: 11

BUILD SUCCESSFUL (total time: 3 seconds)

**Result**

The implementation of Diffie Hellman Key Exchange using a network communication is done successfully.

[Signature of the Staff In-charge]

Name of the Staff In – charge: Mr. Manoj Kumar

Date: 17 – 02 - 2015