ICS 254 Semester Project Report: Diffie Hellman Algorithm

---

by
Faisal Sattar
201584210
Mujtaba Al-Mohsin
201480180

King Fahd University of Petroleum and Minerals

Prepared for

Dr. Wasfi Khatib

25th of April 2018

1- (15 points) Explain the details of the Diffie-Hellman key agreement protocol. In doing so, you must explain the following:

a. The objective(s) of the protocol.

The main objective of this protocol is to securely send cryptographic keys over an open and an unsecure channel.

b. How and under which restrictions/environment does it work?

This can be explained using an example. Suppose Faisal, Mujtaba and Dr. Wasfi belong to a single Whatsapp group and Faisal wants to send a message to Mujtaba in the group without Dr. Wasfi knowing what it is. This can be done by assigning two values; A Prime modulo P and a base B such that both these values are public. For this protocol to work, the base B much be a primitive root of P.

Mujtaba and Faisal generate a private number each which is known only to them. These private numbers can be known as P_F and P_M.

Now, Mujtaba computes the Base to the power of his private number and mods it with P and sends it to Faisal. Let us call this number $Public_M$.
Then Faisal also computes the Base to the power of his private number and mods it with P and sends it to Mujtaba. Let us call this number $Public_F$.

Faisal and Mujtaba now compute $Public_M$ to the power of (P_F mod P) and $Public_F$ to the power of (P_M mod P) respectively and they get the shared secret key.

The reason this method works is because Dr. Wasfi will need at least one of the private numbers to get the shared key. Finding the private number in this case is extremely hard since because there can be an infinite set of Discrete Logarithms to that specific P.

c. Determine which information can be made public and which information must be made private.

Public information: The base B and the prime modulo P. The exchanged information between two parties is also made public.
Private information: The random private numbers of Faisal and Mujtaba.

2- (60 points) Implement the Diffie-Hellman key agreement protocol using Java, with the ability to use very large numbers. In this implementation, you should verify that the user input consisting of a prime number p and a primitive root a are valid. If not, the program should reject the current input and ask the user to enter correct input. The keys $k1$ and $k2$ are randomly generated.

Please check the attached Java File.

3- (70 points) BONUS: Try to break the security of this protocol by writing code that will generate the shared key as output.

Please check the attached Java File.

4- (15 points) Run your program with different values of p and a and show snapshots of the running of the program with different input values (invalid and valid).

Case 1: Negative Modulo Number

```
Enter Modulo Prime Number p:
-80
Enter a positive number only.

Enter Modulo Prime Number p:
```

Case 2: Not a Prime Modulo Number

```
Enter Modulo Prime Number p:
99
99 is not a Prime Number
        Enter again

Enter Modulo Prime Number p:
```

Case 3: Positive and a Prime Modulo Number but a negative Primitive Root

```
Enter Modulo Prime Number p:
8011

Enter Primitive Root of p:
-23

 ...PLEASE WAIT...

Enter a postive Number only

Enter Primitive Root of p:
```

Case 4: Positive and a Prime Modulo Number but not a Primitive Root

```
Enter Modulo Prime Number p:
8011

Enter Primitive Root of p:
23
 ...PLEASE WAIT...

23 is not a Primitive Root of 8011
Enter Again

Enter Primitive Root of p:
```

Case 5: Positive and a Prime Modulo Number and a Positive Primitive Root

```
Enter Modulo Prime Number p:
13463

Enter Primitive Root of p:
910
 ...PLEASE WAIT...

Random Private Number of A: 56
Random Private Number of B: 57
Public Key of A: 12095
Public key of B: 7179

Shared secret number of A and B: 6169

(BONUS) Find Shared Secret With Only Public Keys: 6169
{CHECK WORKING IN CODE}
```

5- (10 points) Include a small report that contains the following information:

a. How to compile and run the code.

To run the code, simply import the attached .rar file and run the code on any IDE.
Enter the values of the required Prime Modulo and the Primitive Roots. Depending on the size of the numbers, it will give a result within a fraction of a second or within a couple of minutes.

b. Any specific details and/or algorithms that you have implemented, especially about the BONUS part.

TO CHECK IF A NUMBER IS PRIME:

This was done by simply using a boolean value for BigInteger.isPrime(int certainty) method after it was given the go-ahead by our instructor.

TO CHECK FOR THE PRIMITIVE ROOT:

This was done by using two ArrayLists of type BigInteger and storing each number from 1 to the (Prime Modulus minus 1) numbers in Array1 and then using modPow method to get the power of the entered number modded by the Prime Modulo and storing it in Array2. After this was done, we checked to see if each number in Array2 contained each number in Array1. If it did, then it was indeed the Primitive Root of that Prime Modulo Number.

TO FIND THE SECRET KEY:

We used modPow again on the Public keys with the power as the Private Keys and the mod value as the Prime Modulo.

BONUS – TO FIND THE SHARED KEY WITHOUT THE PRIVATE VALUES:

This was done by using a method called getPow which basically calculated each power and tested if it was equal to the public key by a brute-force method. After getting the power, it was put into the trusty modPow method to get the secret key.

c. The sample runs of your program that are described in Point "4".

```
Enter Modulo Prime Number p:
47

Enter Primitive Root of p:
11
 ...PLEASE WAIT...

Random Private Number of A: 68
Random Private Number of B: 11
Public Key of A: 17
Public key of B: 39

Shared secret number of A and B: 6

(BONUS) Find Shared Secret With Only Public Keys: 6
{CHECK WORKING IN CODE}
```

```
Enter Modulo Prime Number p:
22303

Enter Primitive Root of p:
12370

 ...PLEASE WAIT...

Random Private Number of A: 22
Random Private Number of B: 58
Public Key of A: 18147
Public key of B: 7442

Shared secret number of A and B: 691

(BONUS) Find Shared Secret With Only Public Keys: 691
{CHECK WORKING IN CODE}
```

```
Enter Modulo Prime Number p:
45737

Enter Primitive Root of p:
38297

 ...PLEASE WAIT...

Random Private Number of A: 70
Random Private Number of B: 58
Public Key of A: 2034
Public key of B: 40954

Shared secret number of A and B: 11086

(BONUS) Find Shared Secret With Only Public Keys: 11086
{CHECK WORKING IN CODE}
```

d. Who did what in the programming assignment.

Initially, all the programming was done by Mujtaba, but the logic of his implementation was in Integer and Big Integers couldn't be accommodated in his code. After he sent the code to Faisal, Faisal made several changes to the core logic in the working of the program and put a cleaner implementation of his code. In the end, both joined up and did the bonus part of the assignment, and later since Mujtaba was the busier of the two, Faisal solely made the report on his own, with complete formatting and the layout done by Faisal.