MATH3815 project3 --- Diffie-Hellman Key Exchange simulation

implementing Fast Exponentiation and Miller-Rabin Primality Test

Designer: Dai Junyan

Email: daiju@kean.edu

Instructor: Dr. Pinata Winoto

Date: 2018/6/24

Introduction:

This program purposes on simulating Diffie-Hellman Key Exchange algorithm with Fast Exponentiation Modulus and Miller-Rabin Primality Test. The language applied in this program is JAVA.

Methods:

This program can be clearly divided into three parts: Diffie-Hellman Key Exchange, Fast Exponentiation Modulus and Miller-Rabin Primality Test. I will explain these three implementations one by one:

Diffie-Hellman Key Exchange:

Set-up: 1. The program asks user enter a prime number p. (the program will test if the entered p is likely a prime or not using Miller-Rabin Primality Test, it will be explained later)

2. Randomly generate an integer g∈{2,3,…,p-2}

3. Publish p and g

Alice		**Bob**
choose $a = k_{pr,A} \in \{2, \ldots, p-2\}$		choose $b = k_{pr,B} \in \{2, \ldots, p-2\}$
compute $A = k_{pub,A} \equiv \alpha^a \bmod p$		compute $B = k_{pub,B} \equiv \alpha^b \bmod p$
	$\xrightarrow{k_{pub,A}=A}$	
	$\xleftarrow{k_{pub,B}=B}$	
$k_{AB} = k_{pub,B}^{k_{pr,A}} \equiv B^a \bmod p$		$k_{AB} = k_{pub,A}^{k_{pr,B}} \equiv A^b \bmod p$

Fast Exponentiation Modulus:

In the public and private computation and primality test, the Square-and-Multiply for Modular Exponentiation is applied to speed up the power modulus calculation. The rationale is explained in detail in textbook 7.4

Fast Exponentiation

Square-and-Multiply for Modular Exponentiation
Input:
base element x
exponent $H = \sum_{i=0}^{t} h_i 2^i$ with $h_i \in 0, 1$ and $h_t = 1$
and modulus n
Output: $x^H \mod n$
Initialization: $r = x$
Algorithm:

```
1      FOR i = t − 1 DOWNTO 0
1.1         r = r^2 mod n
            IF h_i = 1
1.2             r = r · x mod n
2      RETURN (r)
```

Miller-Rabin Primality Test:

Primality test can improve my program to randomly generate prime integers. But in this program, the user can directly enter a number and the program will tell user if the number entered is probably prime or certainly composite. There are two ways for primality test, Fermat Primality Test and Miller-Rabin Primality Test. Compare these two algorithm(textbook 7.6.2), Miller-Rabin Primality Test is relatively powerful method and is often used to generate RSA and DHKE primes.

Miller-Rabin Theorem:

Theorem 7.6.1 *Given the decomposition of an odd prime candidate* $\tilde{p}$

$$\tilde{p} - 1 = 2^u r$$

where r is odd. If we can find an integer a such that

$$a^r \not\equiv 1 \bmod \tilde{p} \quad \text{and} \quad a^{r2^j} \not\equiv \tilde{p} - 1 \bmod \tilde{p}$$

for all $j = \{0, 1, \ldots, u-1\}$, *then* $\tilde{p}$ *is composite. Otherwise, it is probably a prime.*

Miller-Rabin pseudo code:

Miller–Rabin Primality Test
Input: prime candidate $\tilde{p}$ with $\tilde{p} - 1 = 2^u r$ and security parameter s
Output: statement "$\tilde{p}$ is composite" or "$\tilde{p}$ is likely prime"
Algorithm:

```
1      FOR i = 1 TO s
            choose random a ∈ {2, 3, ..., p̃ − 2}
1.2         z ≡ a^r mod p̃
1.3         IF z ≢ 1 and z ≢ p̃ − 1
1.4             FOR j = 1 TO u − 1
                    z ≡ z^2 mod p̃
                    IF z = 1
                        RETURN ("p̃ is composite")
1.5             IF z ≠ p̃ − 1
                    RETURN ("p̃ is composite")
2      RETURN ("p̃ is likely prime")
```

Output Demo:

If I entered a composite integer at first, the error message would be prompted. And ask me to enter a new prime integer:

```
Enter a prime p: 99
ERROR! The number 99 is not a prime number!
Enter a prime p:
```

If I entered correctly(prime integer), the program starts Diffie-Hellman Key Exchange simulating:

```
Enter a prime p: 99
ERROR! The number 99 is not a prime number!
Enter a prime p: 101
g is 69
Alice chooses her private key(randomly): 85
Bob chooses his private key(randomly): 48
Alice sends to Bob her computed public key: 91
Bob sends to Alice his computed public key: 36
Alice got the common key: 1
Bob got the common key: 1
```