## **CS3009-NETWORK SECURITY & CRYPTOGRAPHY**

**CODING ASSIGNMENT**

### **NAME:***SUPRATIM SAHOO*

**ROLL.NO:**2019158

**README**

In the ‘Assignment folder’ ,all the code files are present.Code has been implemented using **Java** programming language.

They include:

1. Util.java
2. AddRoundKey.java
3. ShiftRows.java
4. SubNibbles.java
5. MixColumns.java
6. KeyGenerator.java
7. Server.java
8. Client.java

**\* All inputs provided by the user will be in the form of whole numbers.This makes implementation of the algorithms simpler as I was facing difficulty while using other techniques**.

**DESCRIPTION OF CODE FILES:**

**1.Util.java**

This program file provides utility services for handling nibbles like creating empty nibbles,printing nibbles,etc.

Functions/Methods used in the program file:

* byte[][] createEmptyNibbles() : This method creates and returns empty nibbles(2X2 byte matrix) when invoked.
* void printNibbles(byte[][]) : This method takes nibble as argument and displays it in 2x2 matrix format of 4 bit binary string.
* byte[] nibblesToArray(byte[][]) : This method takes nibble as an argument and converts it into a byte array of 4 elements and returns it.
* byte[][] arrayToNibbles(byte[] ) : This method takes byte array as argument and converts it to nibble and returns it.
* byte bitsToByte(String) : This method takes binary bit string as input and converts it into byte value and returns it.
* short bitsToShort(String) : This method takes binary bit string as input and converts it into short value and returns it.
* void main() : Main method used for testing.

**2.AddRoundKey.java**

This program file provides Add Round Key functionality of S-AES.

Functions/Methods used in the program file:

* byte[][] add(byte[][] , byte[][]) : This method takes two nibbles as arguments and adds(XORs) them and returns the resultant nibble.
* void main() : Main method used for testing.

**3.ShiftRows.java**

This program file provides Shift Rows and Inverse Shift Rows functionality of S-AES.

Functions/Methods used in the program file:

* byte[][] shift(byte[][]) : This method takes a nibble as an argument and right shifts the second row by 1 step.
* void main() : Main method used for testing.

**4.SubNibbles.java**

This program file provides Substitute Nibbles and Inverse Substitute Nibbles functionality of S-AES.Substitution and Inverse Substitution is done from ‘lookupTable’ and ‘inverseLookupTable‘ respectively.

Functions/Methods used in the program file:

* byte[][] substitute(byte[][]) : This method takes a nibble as an argument and returns the substituted nibble by looking in ‘lookupTable’.
* byte[][] invertSubstitute(byte[][]) : This method takes a nibble as an argument and returns the inverse substituted nibble by looking in ‘inverseLookupTable’.
* byte subNib(byte) : This function takes byte as input and returns the substituted byte by looking in ‘lookupTable’.
* byte invertSub(byte) : This function takes byte as input and returns the substituted byte by looking in ‘inverseLookupTable’.
* void main() : Main method used for testing.

**5.MixColumns.java**

This program file provides Mix Columns and Inverse Mix Columns functionality of S-AES.Mixing and Inverse Mixing is done with the help of the filled multiplication table ‘multTable’ which contains results of multiplication in GF(16) modulo (x^4+x+1).

Functions/Methods used in the program file:

* byte add(byte , byte ) : This method takes two bytes as arguments and adds(XORs) them and returns the resultant byte.
* byte multiply(byte , byte ) : This method takes two bytes as arguments and multiplies them by looking into ‘multTable’.
* byte[][] mix(byte[][]) : This method takes nibble as an argument and applies MixColumns operation to it and returns the resultant nibble.
* byte[][] invertMix(byte[][]) : This method takes nibble as an argument and applies Inverse MixColumns operation to it and returns the resultant nibble.
* void main() : Main method used for testing.

**6.KeyGenerator.java**

This program file is used for the Key Generation process in S-AES.

Functions/Methods used in the program file:

* KeyGenerator(byte[]):Constructor used for initializing keyBits with secret key entered by the user.
* short genW0():This method generates the w0 keyword used in round key K0 and returns it.
* short genW1():This method generates the w1 keyword used in round key K0 and returns it.
* short genW2():This method generates the w2 keyword used in round key K1 and returns it.
* short genW3():This method generates the w3 keyword used in round key K1 and returns it.
* short genW4():This method generates the w4 keyword used in round key K2 and returns it.
* short genW5():This method generates the w5 keyword used in round key K2 and returns it.
* String fill8Bits(String) : This method fills 8 bits of a binary string less than 8 bits with leading zeroes and returns it.
* String fill4Bits(String) : This method fills 4 bits of a binary string less than 4 bits with leading zeroes and returns it.
* short[] splitWord(short) : This method splits a 8 bit word into two 4 bit nibbles and returns an array containing both the nibbles.
* byte[][] keyNibbles(int , int ) : This method is used for generating round key nibbles based on round numbers and return it.
* short[] rotateWordNibbles(short[] ) : This method is used for rotating word nibbles and returning the rotated word nibbles.
* short g(short, int ) : This method is a helper function used for generating round keys which uses the above methods and returns a specific value.
* void generate() : This method invokes the methods genW0() to genW5() for creating round keys.
* void main() : Main method used for testing.

**7.Server.java**

This program file provides all the functionalities required on the server side.

Functions/Methods used in the program file:

* byte[][] decrypt(byte[][] ):This method is used for the S-AES decryption process , displays steps of the process and returns the plaintext nibble.
* byte[] getSHA(String) : This method takes string value as an argument and applies SHA-256 algorithm to it and returns a byte array of the computed hash.
* int toInt\_Value(byte[] ) : This method byte array as input and converts it into Integer value and returns it.
* long modInverse(long, long) : This method takes two long values(number,modulus) as argument and returns the modular inverse of the number w.r.t modulus.
* long exponentmod(long a,long b, long c) :This method takes three long values as arguments and returns the modular exponentiation of the arguments,i.e.((a^b)%c).
* long RSA\_key(long p,long q,long e) : This method takes the public key parameters as input and computes the private key and returns it.
* long RSA\_decrypt(long,long,long) : This method takes three arguments(ciphertext,key,modulus) and performs RSA decryption process and returns plaintext.
* long gcd(long , long ): This method takes two long arguments and computes their gcd and returns it.
* void main() : Main method used for coordinating different functionalities of the Server.It takes input of public key parameters from the user and sends public key to the Client.It receives Ciphertext,Encrypted Secret Key,Digital Signature,Client public key and performs necessary computations on them.

**8.Client.java**

This program file provides all the functionalities required on the client side.

Functions/Methods used in the program file:

* byte[][] encrypt(byte[][] ):This method is used for the S-AES encryption process , displays steps of the process and returns the ciphertext nibble.
* byte[] getSHA(String) : This method takes string value as an argument and applies SHA-256 algorithm to it and returns a byte array of the computed hash.
* int toInt\_Value(byte[] ) : This method byte array as input and converts it into Integer value and returns it.
* long modInverse(long, long) : This method takes two long values(number,modulus) as argument and returns the modular inverse of the number w.r.t modulus.
* long exponentmod(long a,long b, long c) :This method takes three long values as arguments and returns the modular exponentiation of the arguments,i.e.((a^b)%c).
* long RSA\_key(long p,long q,long e) : This method takes the public key parameters as input and computes the private key and returns it.
* long RSA\_encrypt(long,long,long) : This method takes three arguments(plaintext,key,modulus) and performs the RSA encryption process and returns ciphertext.
* long gcd(long , long ): This method takes two long arguments and computes their gcd and returns it.
* void main() : Main method used for coordinating different functionalities of the Client.It establishes connection with server (requests for public key).It receivesServer public Key from Server.It takes input of Message,Secret key,public key parameters from the user and sends public key Ciphertext,Encrypted Secret Key,Digital Signature,Client public key to the Server.