Client-Server Message Passing.

A client-server message passing application to provide authentication, integrity and key sharing among both the client and server with the help of RSA and AES algorithm.

It consists of two applications one is for the server side(i.e. Server) and other to the client side(i.e. Client). Both the applications contains Main class and AES class. AES class for both the applications is same.

AES.java Class

It contains the functions to encrypt and decrypt the message and ciphertext respectively.

- mixColumnEncrypt(String s) :- function to mix the column for encryption, returns binary string.
- mixColumnDecrypt(String s) :- function to mix the column for decryption, returns binary string.
- dec2bin(int n, int b) :- function to convert an integer into b bits binary string.
- EncryptionNibbleSubstitute(String b) :- returns substituted nibbles for in encryption purpose.
- DecryptionNibbleSubstitute(String b) :- returns substituted nibbles for in decryption purpose.
- xoring(String a, String b) :- returns the xor of a and b.
- subKeys(String secretKey) :- returns the subKeys of a secretKey.
- rowShift(String str) :- returns shiftedRow string.
- encrypt(int message, int key) :- returns the encrypted message using secretKey.
- decrypt(int ciphertext, int key) :- returns the decrypted ciphertext using secretKey.

Digest

To create a digital signature first we need to find the digest of given message.

- digest(String Message) :- returns the digest of a message using MD5 hash algorithm.

RSA

RSA is a algorithm used to encryption and decryption.

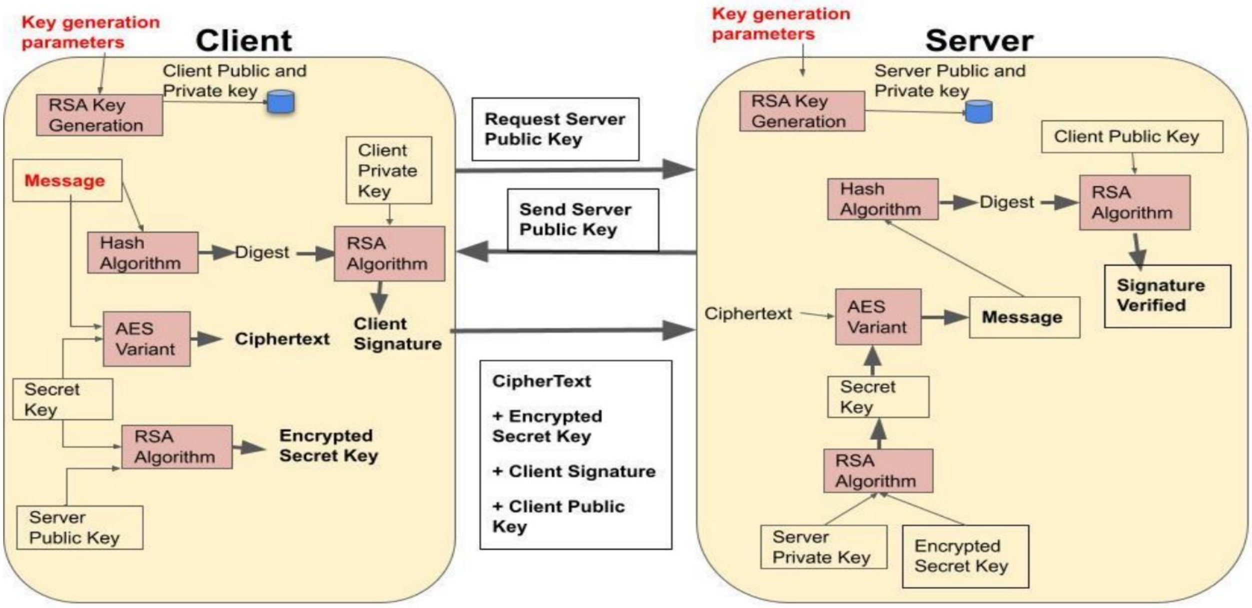
- RSA(BigInteger base, int exponent, int modulus) :- returns the encrypted or decrypted value of the base.

How to Run

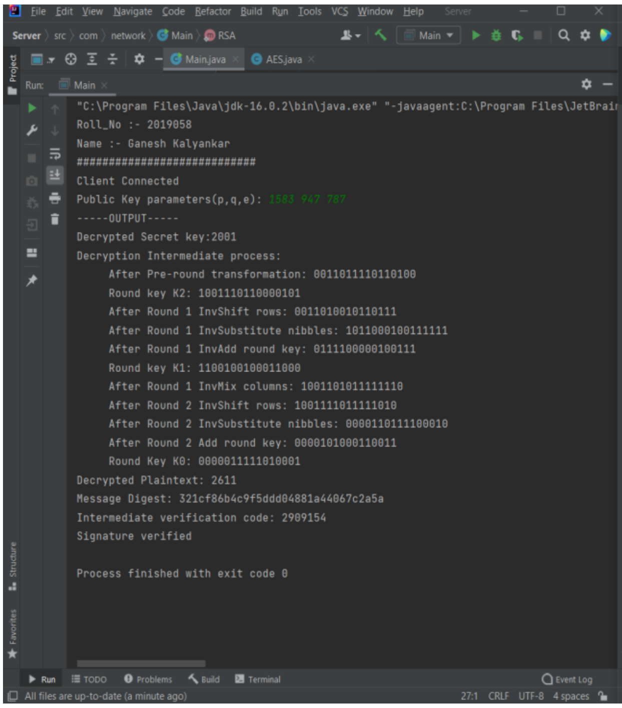
1. Open Server and Client applications in java IDE(i.e. Intellij, eclipse etc).
2. First run the Server.main and then Client .main.
3. Wait till ‘client connected’ message shown in the server side.
4. Now give the input to the client side i.e. 1)message 2)secretKey 3)p, q and e.
5. Now give the input to the server side i.e. 1)p, q and e.

Note :- secretKey strictly less than the modulus(p*q) of server side.

FLOW DAGRAM

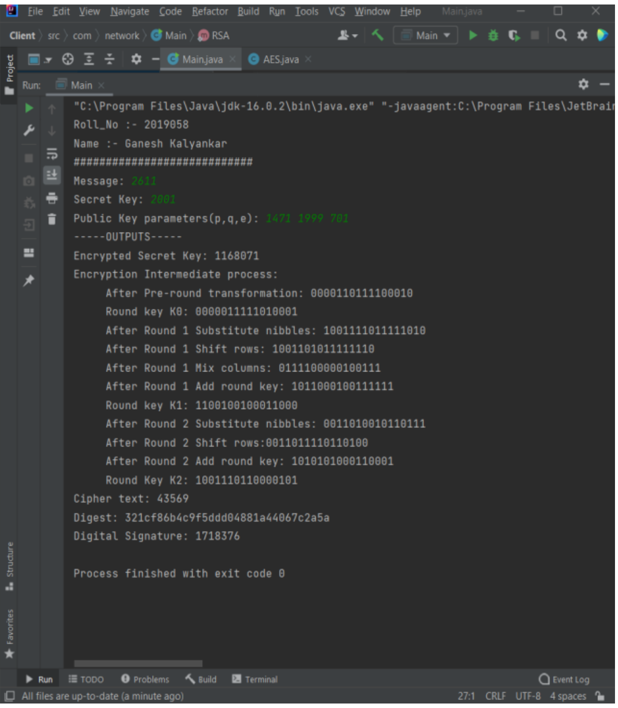


OUTPUT


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help Server
Server src com network Main RSA
Main
Main.java
AES.java
Run: Main
Run TODO Problems Build Terminal Event Log
All files are up-to-date (a minute ago) 27:1 CRLF UTF-8 4 spaces

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrai
Roll_No :- 2019058
Name :- Ganesh Kalyankar
############################
Client Connected
Public Key parameters(p,q,e): 1583 947 787
-----OUTPUT-----
Decrypted Secret key:2001
Decryption Intermediate process:
     After Pre-round transformation: 0011011110110100
     Round key K2: 1001110110000101
     After Round 1 InvShift rows: 0011010010110111
     After Round 1 InvSubstitute nibbles: 1011000100111111
     After Round 1 InvAdd round key: 0111100000100111
     Round key K1: 1100100100011000
     After Round 1 InvMix columns: 1001101011111110
     After Round 2 InvShift rows: 1001111011111010
     After Round 2 InvSubstitute nibbles: 0000110111100010
     After Round 2 Add round key: 0000101000110011
     Round Key K0: 0000011111010001
Decrypted Plaintext: 2611
Message Digest: 321cf86b4c9f5ddd04881a44067c2a5a
Intermediate verification code: 2909154
Signature verified

Process finished with exit code 0
```


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help Main.java
Client src com network Main RSA
Main
Main.java
AES.java
Run: Main
Run TODO Problems Build Terminal Event Log
All files are up-to-date (a minute ago) 27:1 CRLF UTF-8 4 spaces

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrai
Roll_No :- 2019058
Name :- Ganesh Kalyankar
############################
Message: 2611
Secret Key: 2001
Public Key parameters(p,q,e): 1471 1999 701
-----OUTPUTS-----
Encrypted Secret Key: 1168071
Encryption Intermediate process:
     After Pre-round transformation: 0000110111100010
     Round key K0: 0000011111010001
     After Round 1 Substitute nibbles: 1001111011111010
     After Round 1 Shift rows: 1001101011111110
     After Round 1 Mix columns: 0111100000100111
     After Round 1 Add round key: 1011000100111111
     Round key K1: 1100100100011000
     After Round 2 Substitute nibbles: 0011010010110111
     After Round 2 Shift rows:0011011110110100
     After Round 2 Add round key: 1010101000110001
     Round Key K2: 1001110110000101
Cipher text: 43569
Digest: 321cf86b4c9f5ddd04881a44067c2a5a
Digital Signature: 1718376

Process finished with exit code 0
```