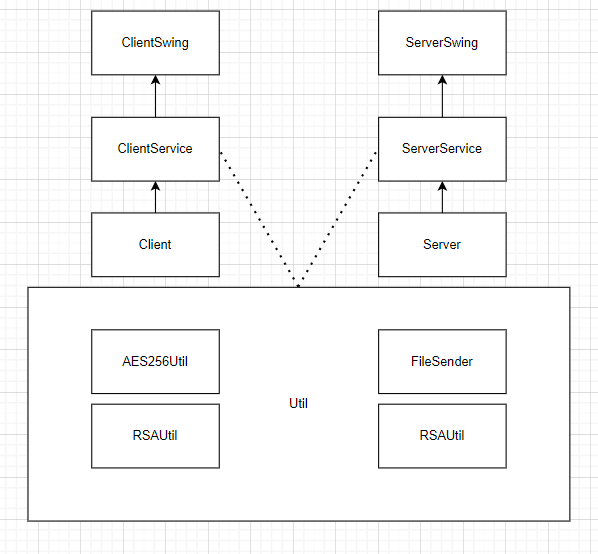
Information Security

Programming Assignment #1

**17102062 Lee Yong Hoon**

1. Class Relation



1. RSAUtil.java

Util class related to RSA

**genRSAKeyPair() method**

* make new keypair that having public key and private key

**encryptRSA(String plainText, PublicKey publicKey) method**

* encrypt plaintext using publicKey

**decryptRSA(String encrypted, PrivateKey privateKey)**

* decrypt ciphertext using privateKey

1. AESUtil

Util class related to AES256

**encrypt(String text,String key)**

* Encrypt text using AESKey

**decrypt(String cipherText, String key)**

* Decrypt ciphertext using AESKey

1. Server.java

Treat all field related to server

텍스트이(가) 표시된 사진

자동 생성된 설명

* Port number is 5050
* IP address and connection status
* Having key information and client publickey information
* Having chatHistory

1. ServerService.java

ServerService.java has all methods related to Server.

But sending and receiving file method are in FileSender.java and FileReceiver.java

1. Client.java

텍스트이(가) 표시된 사진

자동 생성된 설명

* Port number is 5050
* IP address and connection status
* Having key information and client publickey information
* Having chatHistory

1. ClientService

ClientService.java has all methods related to Client.

But sending and receiving file method are in FileSender.java and FileReceiver.java

1. ServerSwing, ClientSwing and scenario

# **Open project**

1. decompress security.zip file

2. make security folder

텍스트이(가) 표시된 사진

자동 생성된 설명3. put all items in security folder

텍스트이(가) 표시된 사진

자동 생성된 설명

4. Open eclipse

5. File -> Import -> General -> Existing Projects into Workspace

텍스트이(가) 표시된 사진

자동 생성된 설명

6. select root directory -> click Browse… -> select directory where ‘security’ folder exists

텍스트이(가) 표시된 사진

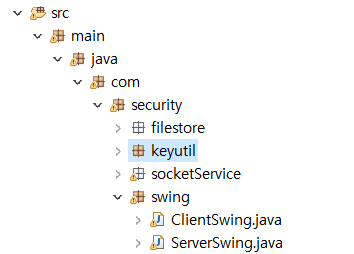
자동 생성된 설명

7. Below picture are structure of my projects

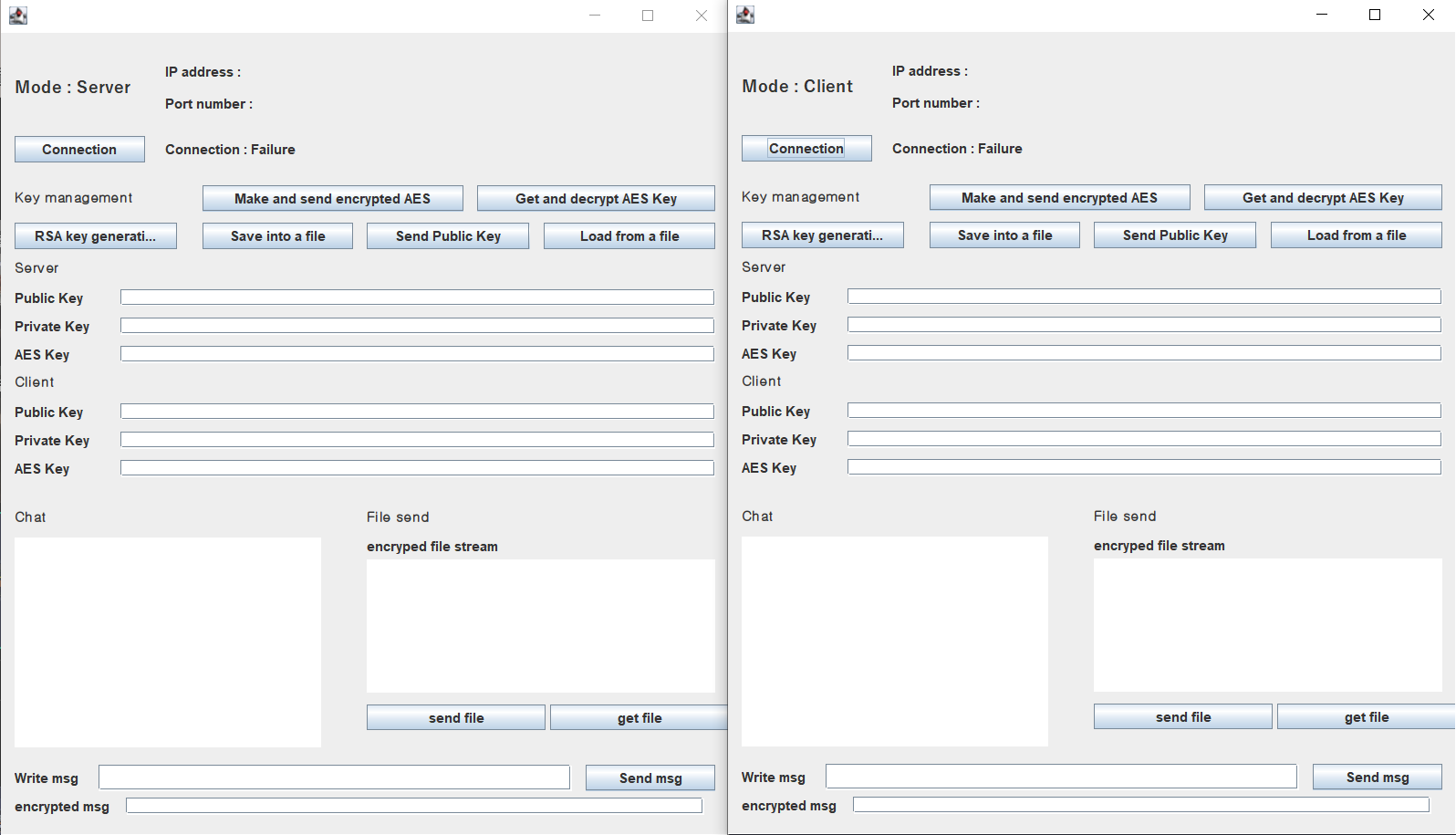
You can click ClientSwing and ServerSwing and run each file

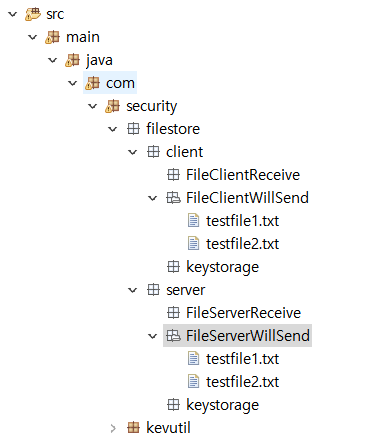
텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

8. You can use 2 Swing program one is for server and other is for client



9. before start, there are client local directory and server local directory.

**Client local directory : Filestore.client**

**FileClientReceive** : when server send file in FileServerWillSend, receiver file is saved in this folder

**FileClientWillSend** : files that client will send to server

**Keystorage** : save client’s public key and private key. Also save server’s public key

**Server local directory : filestore.server**

**FileServerReceive** : when server send file in FileClientWillSend, receiver file is saved in this folder

**FileServerWillSend** : files that client will send to server

**Keystorage** : save client’s public key and private key. Also save server’s public key

# **Make connection**

You can click connection in server mode

You can see that waiting client connection

텍스트이(가) 표시된 사진

자동 생성된 설명

After that click connection in client mode

You can see that client connection is successful

텍스트이(가) 표시된 사진

자동 생성된 설명

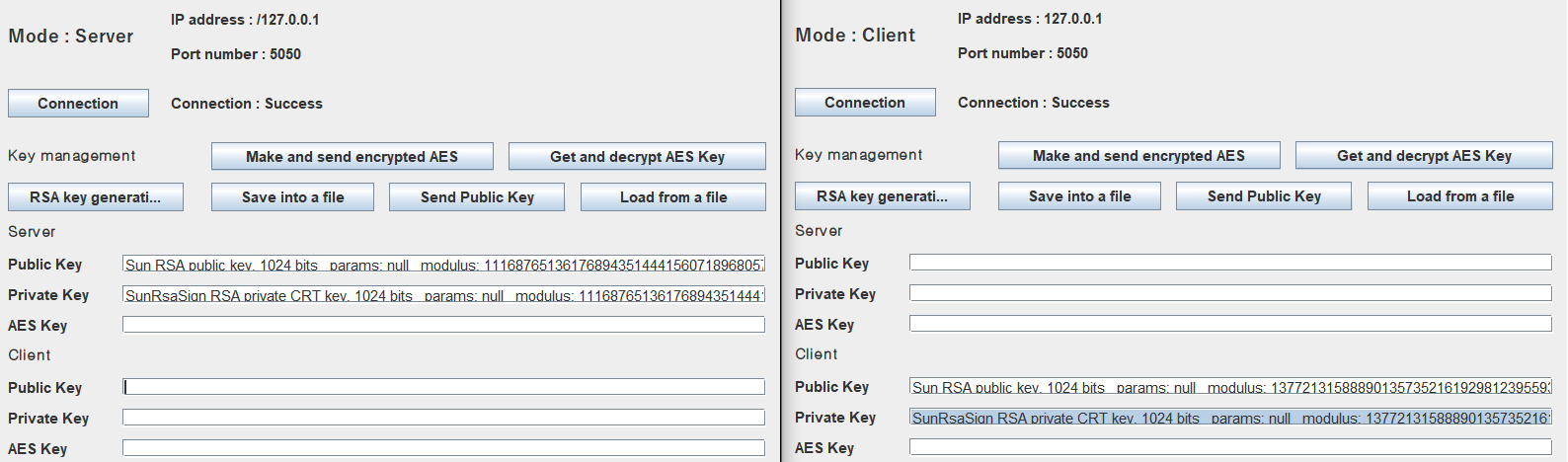
After that, you click connection in server again.

You can see that server connection is also successful.

텍스트이(가) 표시된 사진

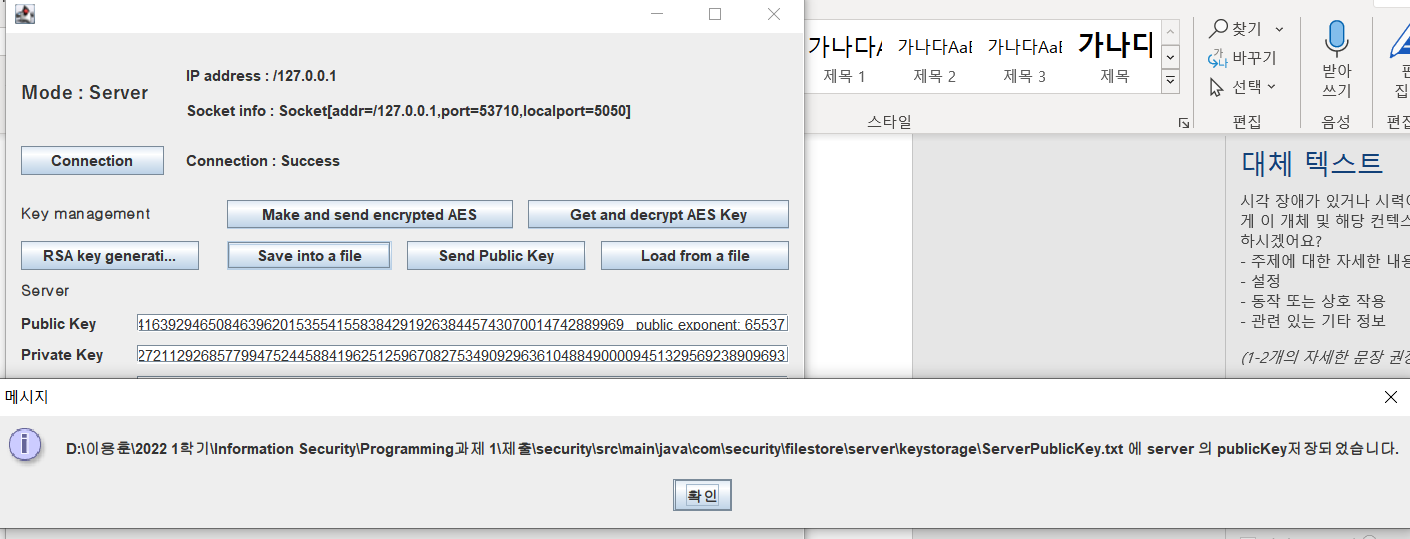
자동 생성된 설명

# **Make RSA key pair**



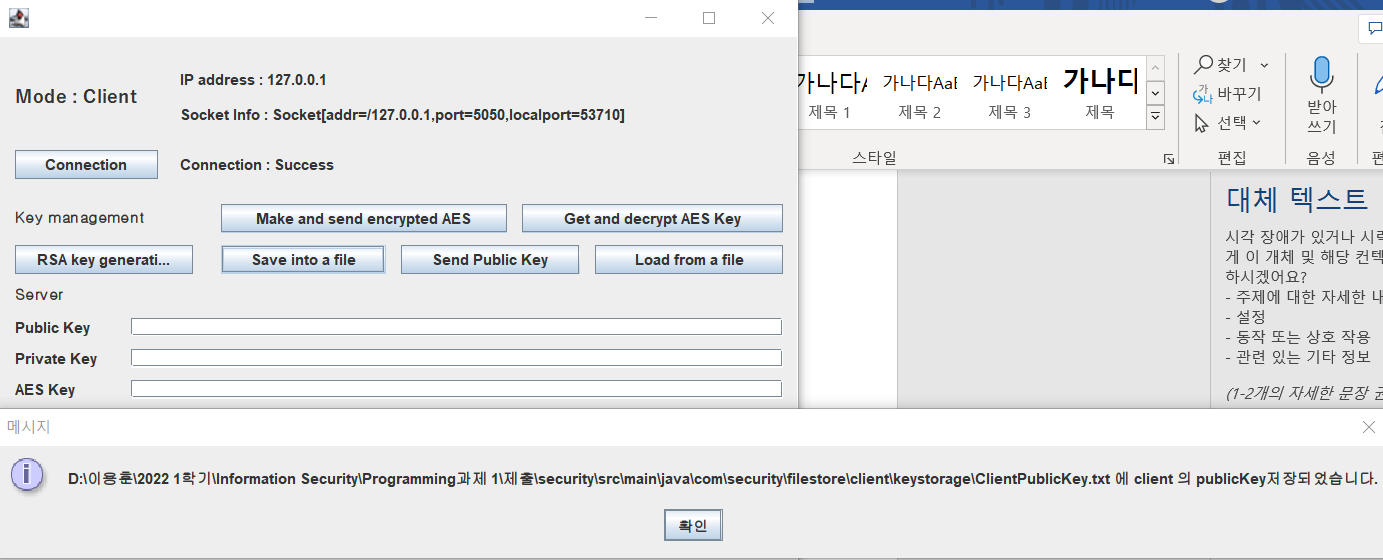
If you click the RSA key generation button in server mode and client mode, you can get different key pair per each.

# **Save RSA key file**



If you click save into a file button, you can save your public key into your local server directory.

(src\main\java\com\security\filestore\server\keystorage\ServerPublicKey.txt)



If you click save into a file button, you can save your public key into your local client directory.

(src\main\java\com\security\filestore\client\keystorage\ClientPublicKey.txt)

# **Send RSA key from server to client**

텍스트이(가) 표시된 사진

자동 생성된 설명

If you click send public key button on server mode, the public key text will send to client

텍스트이(가) 표시된 사진

자동 생성된 설명

In client mode, if you click load from a file button, key text is save in client local from server.

Server’s Public key text is save in client local directory

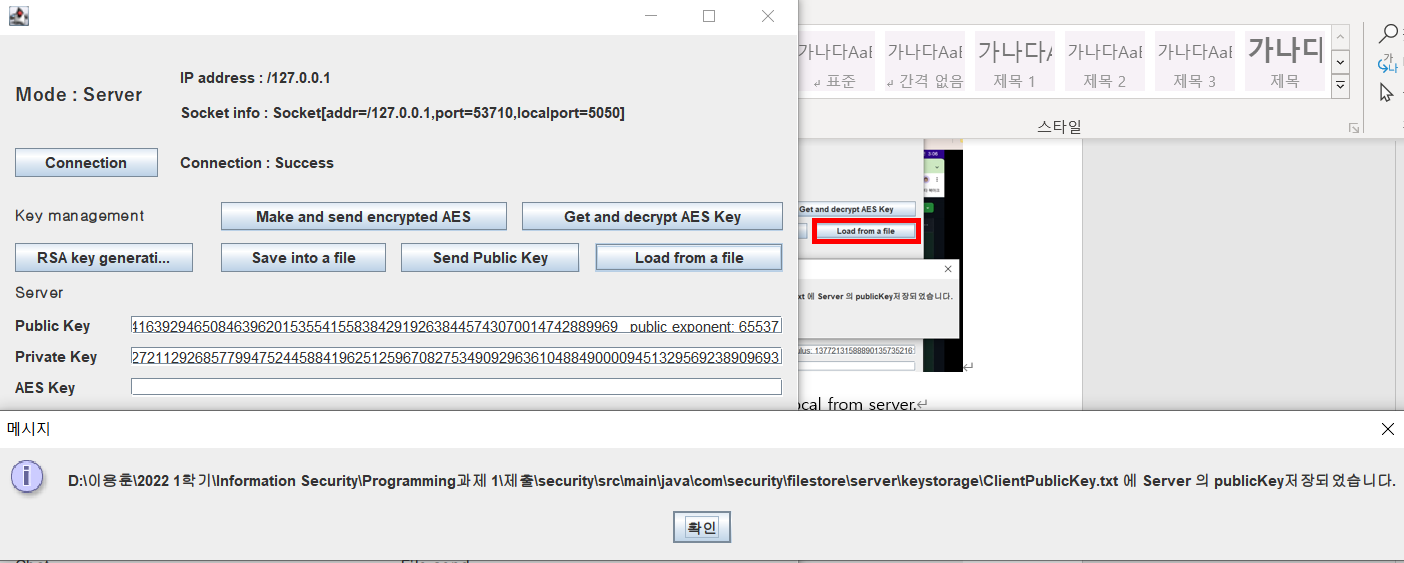
(src\main\java\com\security\filestore\client\keystorage\ServerPublicKey.txt)

# **Send RSA key from client to server**

텍스트이(가) 표시된 사진

자동 생성된 설명

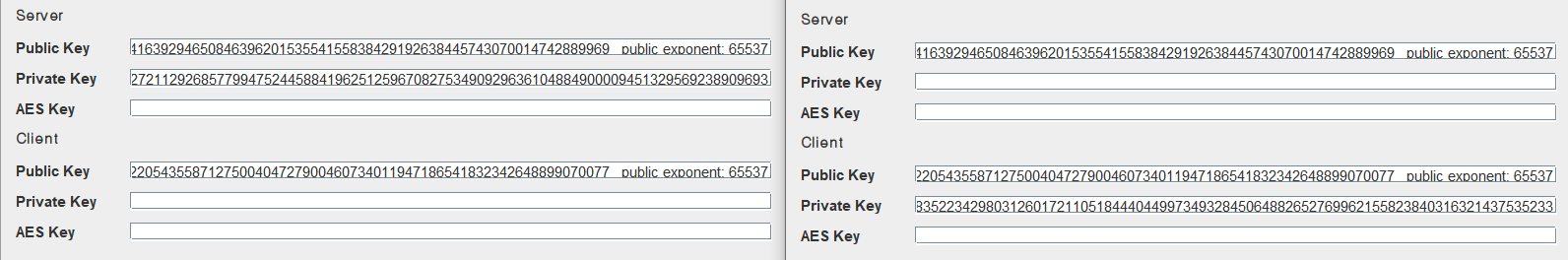
If you click send public key button on client mode, the public key text will send to client



In server mode, if you click load from a file button, key text is save in server local from server.

Client’s Public key text is save in client local directory

(src\main\java\com\security\filestore\server\keystorage\ClientPublicKey.txt)



RSA exchange is completed.

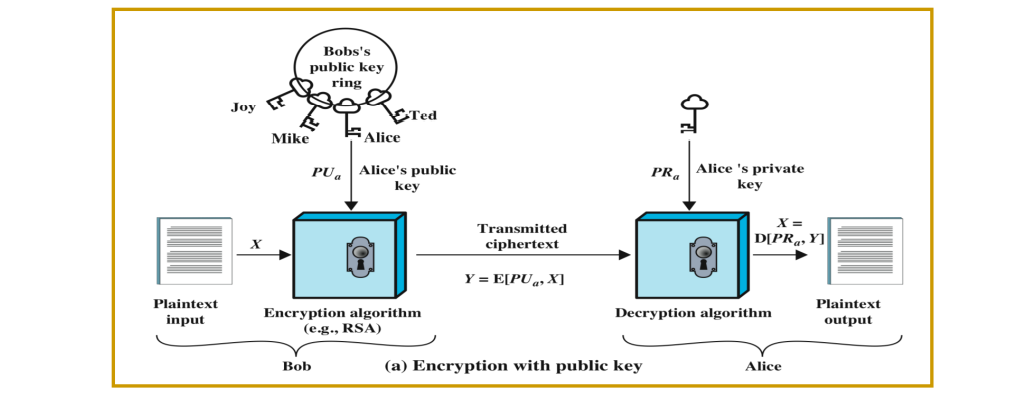
You can see that client and server have own public key and private key and other side’s public key

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

# **Make AES Key and encrypt by other side’s public and other side will decrypt by using own private key**



We use this way.

텍스트이(가) 표시된 사진

자동 생성된 설명

In server click make and send encrypted AES button. In server AES Key is made and AES key is encrypted by client’s public key and send to client.

텍스트이(가) 표시된 사진

자동 생성된 설명

If you click get and decrypt AES Key in client, you can get encrypted key and decrypt it by using client’s private key

You can see that AES key of both side are same.

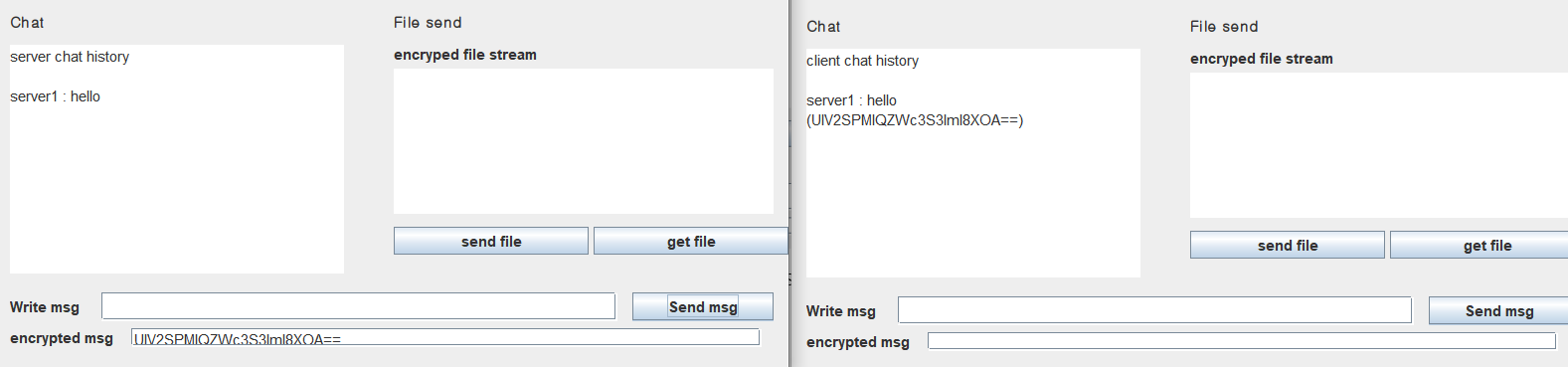
You can exchange AES key with adversary direction(make AES in client and send it to server)

# **Send message**

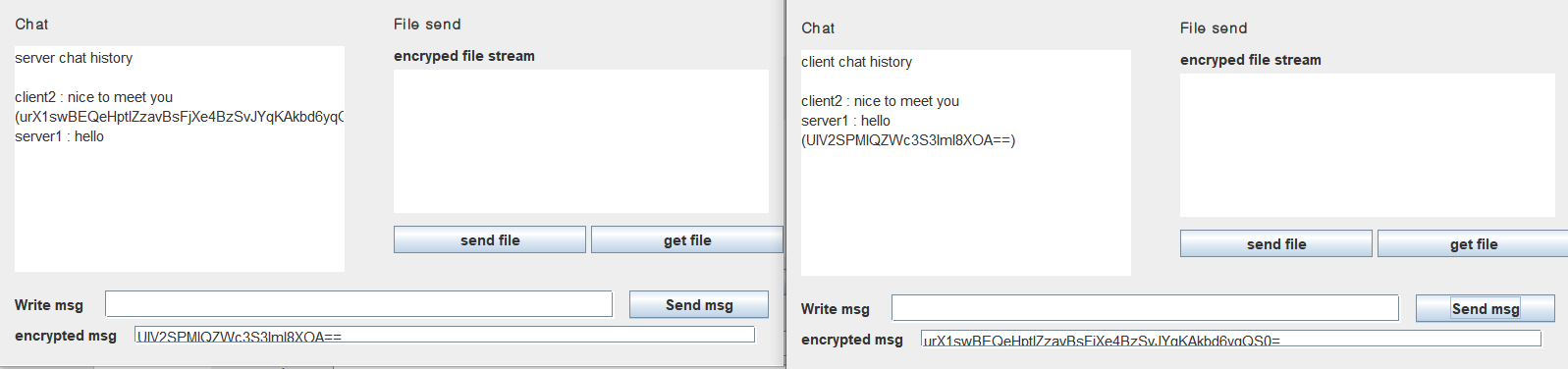
**(Server -> Client)**

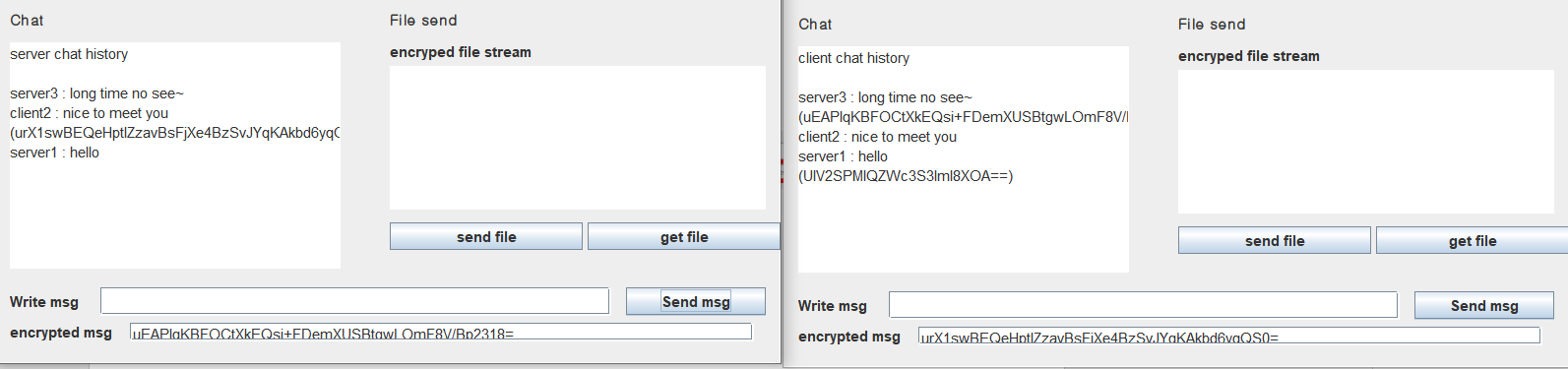


**If you write message and click send msg button, message send to client from server.**

**Message is encrypted by AES key in server and decrypted AES key in client.**

**You can see that both message is send successfully and check that encrypted message are also same in both side.**

**(Client -> Server)**

**(Server -> Client)**

You can write message and click send msg. the message is encrypted by AES key.

Then you can see your encrypted message with your hi message.

And the chatting box is updated.

In client mode, it is continuously check message stream from server.

If client mode check input of stream, server’s encrypted key is decrypted by AES Key. And chatting box in client mode is updated

# **Send File with Digital Signature**

We send file with AES key. In this case, to assure integrity, we attack digital signature when send file. Used digital signature is other side’s public key.

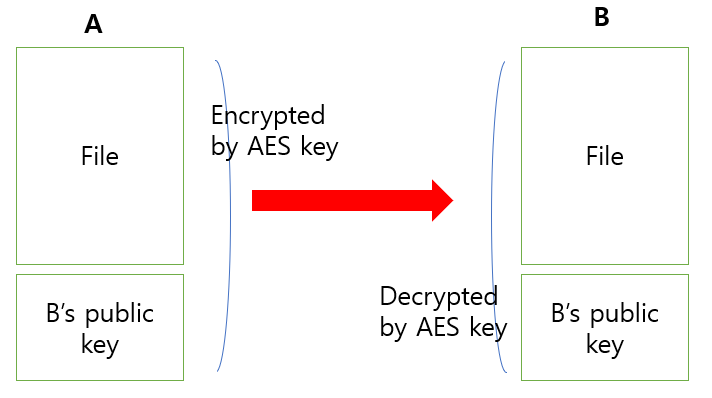
A side attached with B’s public key. The file is encrypted by AES key

B side get encrypted file

B decrypt cipher text by using AES key

B check integrity by using his own public key.

If there is same public key then there is no attack during sending on link



In server, click file button

텍스트이(가) 표시된 사진

자동 생성된 설명

Choose one of the test file

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

You can see that server path and client path and

Red box is digital signature that is client’s public key.

You can see ciphertext of file by AES.

In client, get file button

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

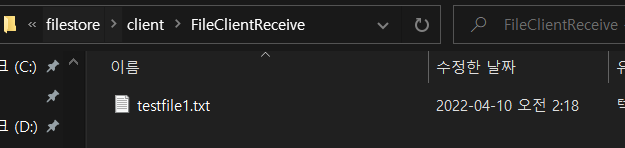
You can see that server path and client path and

Red box is digital signature that is client’s public key.

In this case, digital signature is same.

You cans see digital signature and ciphertext of file by AES

After send file, you can see that file is saved in that directory



텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

You can see that both file are same.

You can send file client to server with the same way.