**Client Server Messaging System**

Design Objective

To allows for 2 parties to exchange message securely.

Demonstration

Launch each application on both Server and Client site.

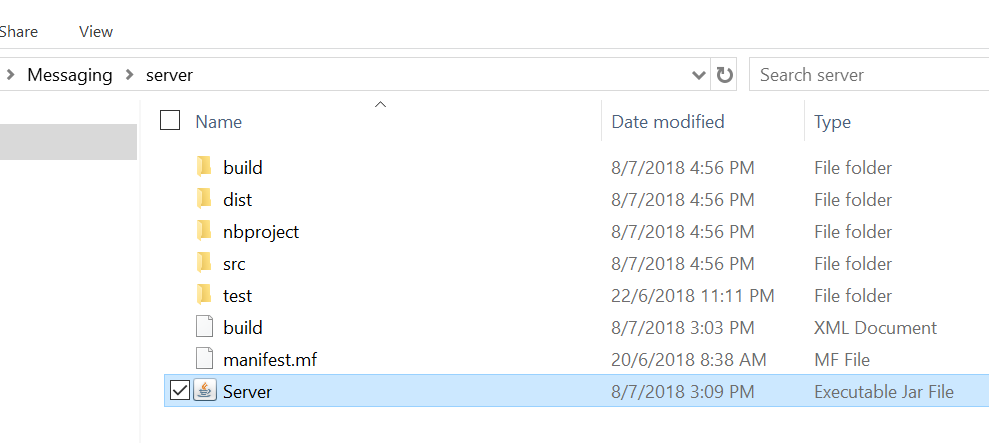


Figure 1 : Launching Server Application

1. Double click the Server jar file as shown in Figure 1. The server application will be appeared as shown in Figure 2.

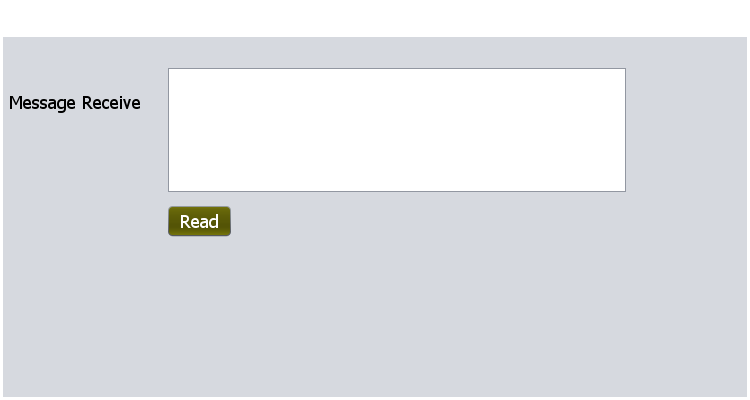


Figure 2 : Server Listening Message from Client

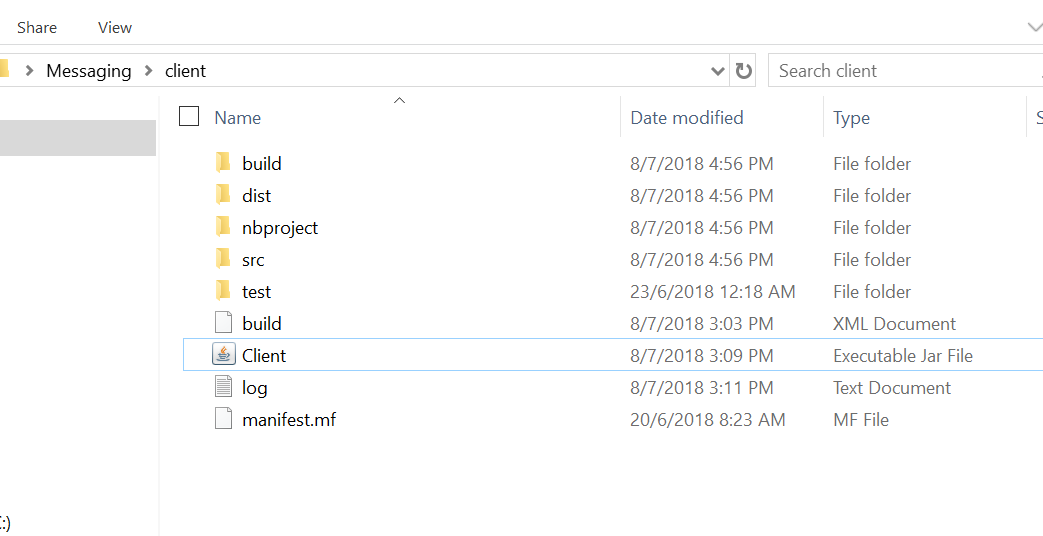


Figure 3 : Launching Client Application

1. Double click the Client jar file as shown in Figure 3. The client application will be appeared as shown in Figure 3.

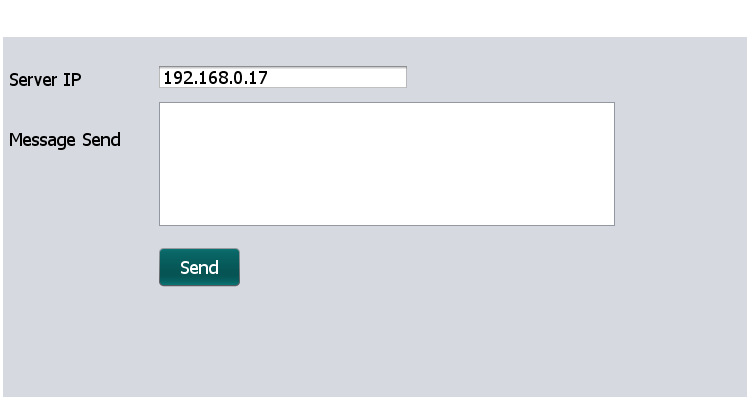


Figure 4 : Client Application

1. Input message and send to Server site.
2. The sender need to give the key for encryption message as shown in Figure 5.

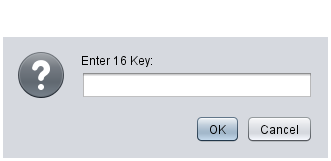


Figure 5 : Enter Key for Encryption

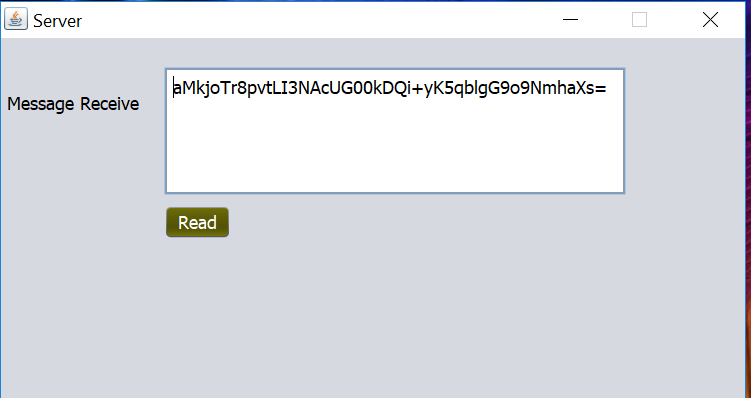


Figure 6 : Encrypted Message Arrived at Server

1. The Server site will receive the encrypted message as shown in Figure 6.
2. To read the message,
   1. Server site need to know the sender given key.
   2. Without correct key and correct encrypted message, the correct original message will not be shown.

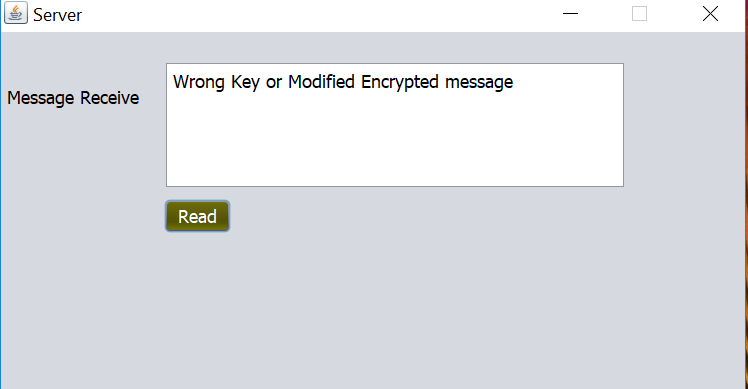


Figure 7 : Wrong Message Reply by System

**Confidentiality**

The middle man between client and server cannot get the original message. Therefore, message exchanging meet confidentiality.

**Integrity**

Assuming that the encrypted message is amended by the middle man, at the time try to read with correct key. The system can detect the modified encrypted message. The system will reply message as shown in Figure 7. Therefore, the message exchanging meet Integrity.

Using Technology

Implementing "AES/CBC/PKCS5PADDING" algorithm by using the following library of Java Platform.

javax.crypto.Cipher;

javax.crypto.spec.IvParameterSpec;

javax.crypto.spec.SecretKeySpec;

java.util.Base64;

Requirement

JDK 8

Netbeans 8.2