Programming Practice on Basic Crypto Zhanghao

How to run the program:

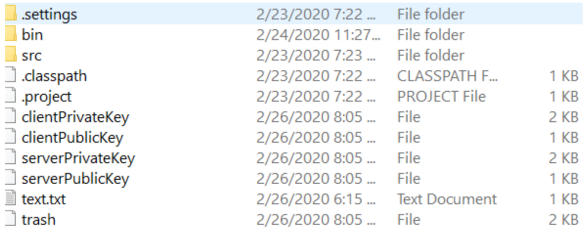
If run on eclipse: run Server.java first and then run Client.java

If run on pyrite: compile java file: javac Server.java Client.java

Run java: java Server first then java Client

Note:

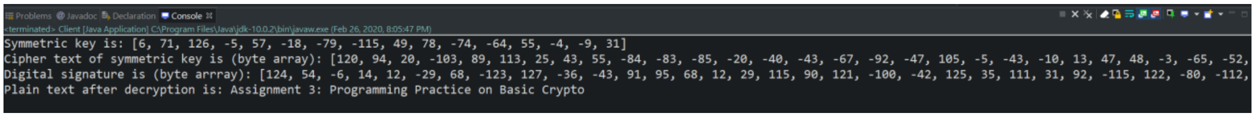
- AsymmetricKeyProducer take two arguments from server and client side instead of three auguments.
- One client with one server so no thread is implemented.
- Size of RSA keys are 2048 bits, size of symmetric key is 16 bytes (128 bits).
- In my implementation, client and server create RSA keys inside project folder and called clientPublicKey, clientPrivateKey, serverPublicKey, serverPrivateKey.
- In the real scenario, we assume that client can access server's public key and server can access client's public key. Although in this assignment we put them together.
- In the read scenario client and server should have fixed keys. In our implementation we assume that each server and client are new, so we create each RSA pairs of keys every time we run the program.
- For the purpose of easy demo, I use server to create both server key and client key instead of making another Test.java file and redo my process again.
- Use SHA256 with RSA to sign the key.
- Key files and text.txt file in under project folder. If user want to change text file source, then user need to go into Server.java file to change the source path.



.settings	2/23/2020 7:22 ...	File folder	
bin	2/24/2020 11:27...	File folder	
src	2/23/2020 7:23 ...	File folder	
.classpath	2/23/2020 7:22 ...	CLASSPATH F...	1 KB
.project	2/23/2020 7:22 ...	PROJECT File	1 KB
clientPrivateKey	2/26/2020 8:05 ...	File	2 KB
clientPublicKey	2/26/2020 8:05 ...	File	1 KB
serverPrivateKey	2/26/2020 8:05 ...	File	2 KB
serverPublicKey	2/26/2020 8:05 ...	File	1 KB
text.txt	2/26/2020 6:15 ...	Text Document	1 KB
trash	2/26/2020 8:05 ...	File	2 KB

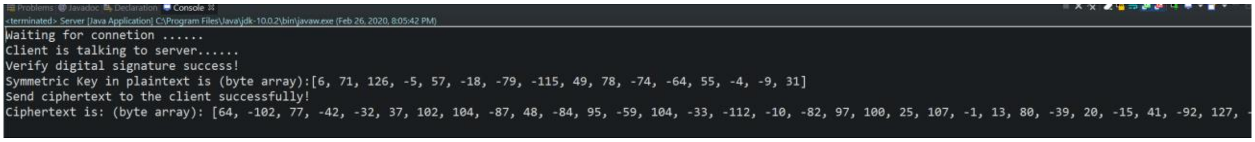
Screenshots:

Output from client console:



```
<terminated> Client [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (Feb 26, 2020, 8:05:47 PM)
Symmetric key is: [6, 71, 126, -5, 57, -18, -79, -115, 49, 78, -74, -64, 55, -4, -9, 31]
Cipher text of symmetric key is (byte array): [120, 94, 20, -103, 89, 113, 25, 43, 55, -84, -83, -85, -20, -40, -43, -67, -92, -47, 105, -5, -43, -10, 13, 47, 48, -3, -65, -52,
Digital signature is (byte arrray): [124, 54, -6, 14, 12, -29, 68, -123, 127, -36, -43, 91, 95, 68, 12, 29, 115, 90, 121, -100, -42, 125, 35, 111, 31, 92, -115, 122, -80, -112,
Plain text after decryption is: Assignment 3: Programming Practice on Basic Crypto
```

Output from server console:



```
<terminated> Server [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (Feb 26, 2020, 8:05:42 PM)
Waiting for connetion ......
Client is talking to server......
Verify digital signature success!
Symmetric Key in plaintext is (byte array):[6, 71, 126, -5, 57, -18, -79, -115, 49, 78, -74, -64, 55, -4, -9, 31]
Send ciphertext to the client successfully!
Ciphertext is: (byte array): [64, -102, 77, -42, -32, 37, 102, 104, -87, 48, -84, 95, -59, 104, -33, -112, -10, -82, 97, 100, 25, 107, -1, 13, 80, -39, 20, -15, 41, -92, 127, -
```