CRYPTOGRAPHY

Programmmieren eines einfachen Verschlüsselungsprogrammes

Janeczek, Mair TGM 5AHITT

Inhaltsverzeichnis

Aufgabenstellung

DezSys06 - Verschlüsselung

Kommunikation [12Pkt]

Programmieren Sie eine Kommunikationsschnittstelle zwischen zwei Programmen (Sockets; Übertragung von Strings). Implementieren Sie dabei eine unsichere (plainText) und eine sichere (secure-connection) Übertragung.

Bei der secure-connection sollen Sie eine hybride Übertragung nachbilden. D.h. generieren Sie auf einer Seite einen privaten sowie einen öffentlichen Schlüssel, die zur Sessionkey Generierung verwendet werden. Übertragen Sie den öffentlichen Schlüssel auf die andere Seite, wo ein gemeinsamer Schlüssel für eine synchrone Verschlüsselung erzeugt wird. Der gemeinsame Schlüssel wird mit dem öffentlichen Schlüssel verschlüsselt und übertragen. Die andere Seite kann mit Hilfe des privaten Schlüssels die Nachricht entschlüsseln und erhält den gemeinsamen Schlüssel.

Sniffer [4Pkt]

Schreiben Sie ein Sniffer-Programm (Bsp. mithilfe der jpcap-Library http://jpcap.sourceforge.net oder jNetPcap-Library http://jnetpcap.com/), welches die plainText-Übertragung abfangen und in einer Datei speichern kann. Versuchen Sie mit diesem Sniffer ebenfalls die secure-connection anzuzeigen.

Info

Gruppengröße: 2 Mitglieder
Punkte: 16

Erzeugen von Schlüsseln: 4 Punkte

Verschlüsselte Übertragung: 4 Punkte

Entschlüsseln der Nachricht: 4 Punkte

Sniffer: 4 Punkte

Anforderungsanalyse

- **Es werden zwei Akteure (Client-Server) für die sinnvolle Kommunikation benötigt.**

- **Die Kommunikation muss mittels Ports ermöglicht werden.**

- **Der Kommunikationsfluss muss mittels eines Sniffers bewiesen werden können.**

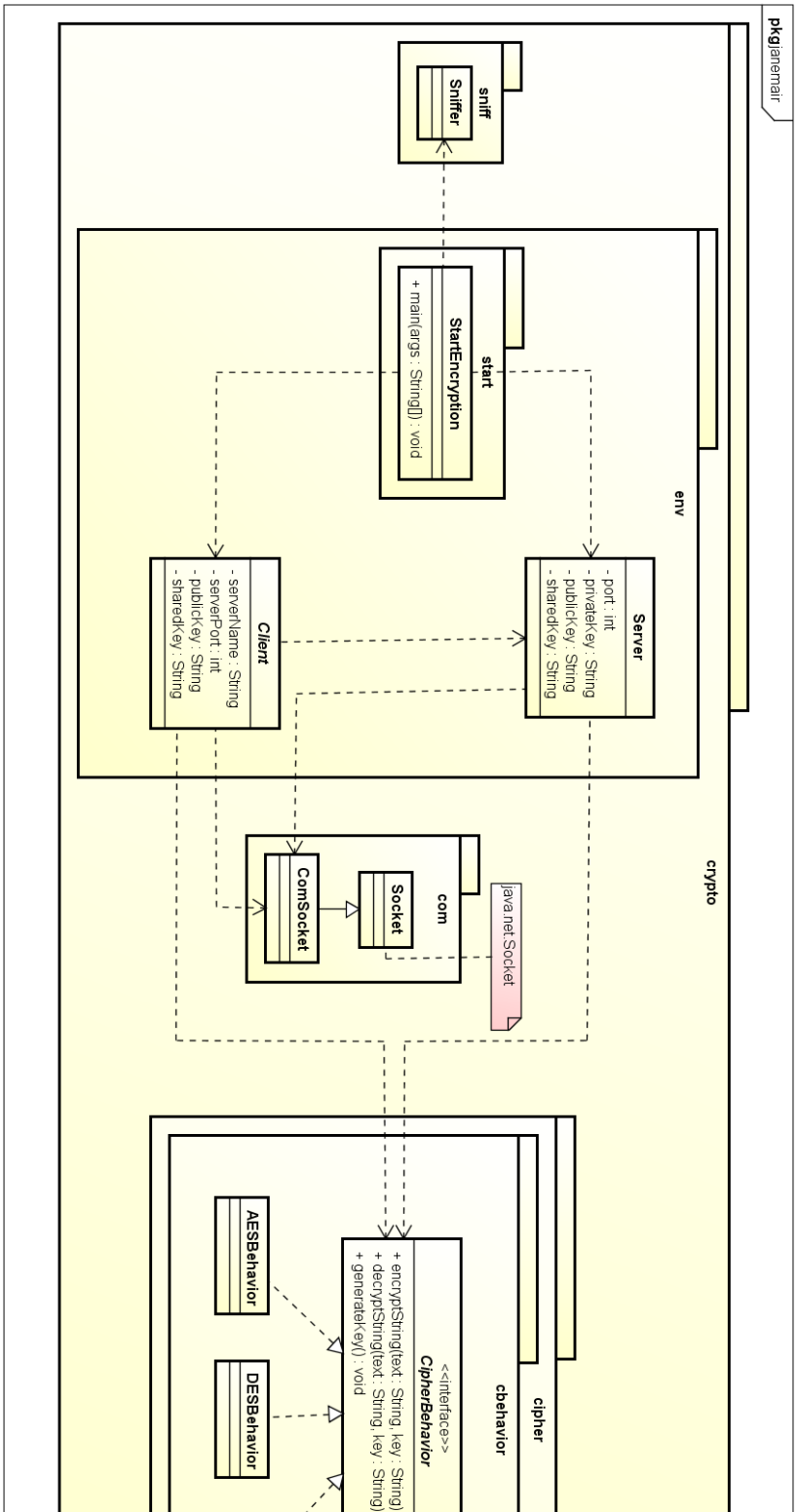
- **Der übertragene Text muss mittels einer Verschlüsselungsmethode (AES), vor dem übertragen, verschlüsselt worden sein.**

Designüberlegung

Wir haben 3 Akteure:

- Client
- Server
- Sniffer

Diese Akteure werden alle Statisch von einer Hauptklasse aufgerufen. Diese Akteure öffnen einen Socket und Kommunizieren über diesen. Der Server lässt einen privateKey und einen publicKey generieren und schickt seinen publicKey zu den Client. Dieser erstellt einen sharedKey und verschlüsselt jenen mit dem publicKey, um ihn danach wieder zurückzuschicken, wo er wieder entschlüsselt wird. Dabei ist die Art und Wiese wie der Key verschlüsselt und erstellt wird mittels eines Strategy-Patterns dynamisch definiert. Der User kann selbst entscheiden, ob er die Nachricht verschlüsselt oder unverschlüsselt versenden möchte.


pkg janemair
sniff
Sniffer
start
StartEncryption
+ main(args : String[]) : void
env
Server
- port : int
- privatekey : String
- publickey : String
- sharedkey : String
Client
- serverName : String
- serverPort : int
- publicKey : String
- sharedkey : String
com
ComSocket
Socket
java.net.Socket
crypto
cbehavior
cipher
<<Interface>>
CipherBehavior
+ encryptString(text : String, key : String)
+ decryptString(text : String, key : String)
+ generatekey() : void
AESBehavior
DESBehavior

Technologiebeschreibung

Sockets

Definition:

A *socket* is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

An endpoint is a combination of an IP address and a port number. Every TCP connection can be uniquely identified by its two endpoints. That way you can have multiple connections between your host and the server. **[2]**

Dieser sogenannte Socket wird uns im Bereich der Kommunikation zwischen der Client und der Serverklasse behilflich sein. Beim Starten dieser wird nämlich ein Port angegeben, der für die Socket-Communication relevant ist.

Verschlüsselungsmethoden

AES

The Advanced Encryption Standard or AES for short is a symmetric block cipher used to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data. **[3]**

Der AES Algorithmus ist der Nachfolger des Data Encryption Standards, da dieser gegen Brute-Force-Attacks verwundbar war. Bisher konnte keine effektive Methodik zum Brechen dieses symetrischen Blockciphers entwickelt werden.

DES

The Data Encryption Standard (DES) is an outdated symmetric-key method of data encryption.

DES is known for using the same key to encrypt and decrypt a message, so both the sender and receiver share one private key. **[4]**

Der Data Encryption Standard wurde aus diesem Grund von dem sichereren Algorithmus AES ersetzt.

RSA

RSA is a cryptosystem for public-key encryption, and widely used for securing sensitive data, particularly when being sent over an insecure network such as the internet. **[5]**

Public-key Algorithmen bestehen grundlegend aus zwei mathematisch zusammengesetzten Schlüsseln, einem öffentlichen und einem privaten Schlüssel. Der öffentliche Schlüssel, wie der Name schon sagt, kann jedem beliebigen gegeben werden, jedoch sollte der private Schlüssel geheim gehalten werden. Dieser wird nämlich sowohl für die Entschlüsselung verwendet.

Sniffer

A *network sniffers* monitors data flowing over computer network links. It can be a self-contained software program or a hardware device with the appropriate software or firmware programming. Also sometimes called "network probes" or "snoops," sniffers examine network traffic, making a copy of the data but without redirecting or altering it. Some sniffers work only with TCP/IP packets, but the more sophisticated tools can work with many other protocols and at lower levels including Ethernet frames. **[6]**

Aufwandsabschätzung und Arbeitszeitaufzeichnung

Janeczek:

Working Hours

DATE	PHASE	TASK	ESTIMATION	ACTUAL	COMMENT
26.01.2015	Dokumentation	Initialisierung des Projekts + Schreiben wesentlicher Dokumentation	1:00:00	0:29:00	Initial Step
26.01.2015	Design	UML-Design Second Draft	2:00:00	0:55:00	Second Design Draft
26.01.2015	Implementation	Implementing the functionality of the program	5:00:00	0:30:00	STILL ONGOING....
12.02.2015	Implementation	Learning from examples	0:30:00	1:00:00	trying to get back into the code
13.02.2015	Implementation	Making Code Executeable	1:00:00	1:30:00	
		SUM	9:30:00	4:24:00	

Mair:

Working Hours

DATE	PHASE	TASK	ESTIMATION	ACTUAL	COMMENT
23.01.2015	Design	Erstellen eines UML-Designs	1:00:00	0:45:00	First Design Draft
26.01.2015	Design	UML-Design Second Draft	2:00:00	0:55:00	Second Design Draft
26.01.2015	Documentation	Requirementsanalyse, Designüberlegung	0:50:00	0:30:00	
27.01.2015	Implementation	Erstellung erster Verschlüsselungsmethoden	1:30:00	2:30:00	
12.02.2015	Implementation	Learning from examples	0:30:00	1:00:00	trying to get back into the code
		SUM	5:50:00	5:40:00	

Gesamt:

NAME	ESTIMATION	ACTUAL
Janeczek	9:30:00	4:24:00
Mair	5:50:00	5:40:00
SUM	15:20:00	10:04:00

Arbeitsdurchführung

Wir haben begonnen indem wir ein Designkonzept in Form eines UML-Diagrammes erstellt haben. Allerdings mit der Intention es nicht zu dynamisch zu gestalten, um es nicht unnötig groß zu gestalten (Wir wollen immerhin rechtzeitig fertig werden). Dabei haben wir beim Design mit dem nötigsten angefangen und dann ein paar dynamische Erweiterungen hinzugefügt. Nachdem wir den ersten Ansatz unseres Designs erstellt haben, haben wir die Aufgaben fürs erste geteilt. Einer schrieb die Dokumentation während der andere sich mit dem Source Code beschäftigt. Über den Ferien ist die Arbeit nicht wirklich vorangegangen weswegen wir uns jetzt entschieden haben das Design-Konzept minimalisiert wird, um so Zeit zu sparen. Desweiteren haben wir uns Beispiele angeschaut, um herauszufinden wie genau die Verschlüsselung in Java funktioniert.

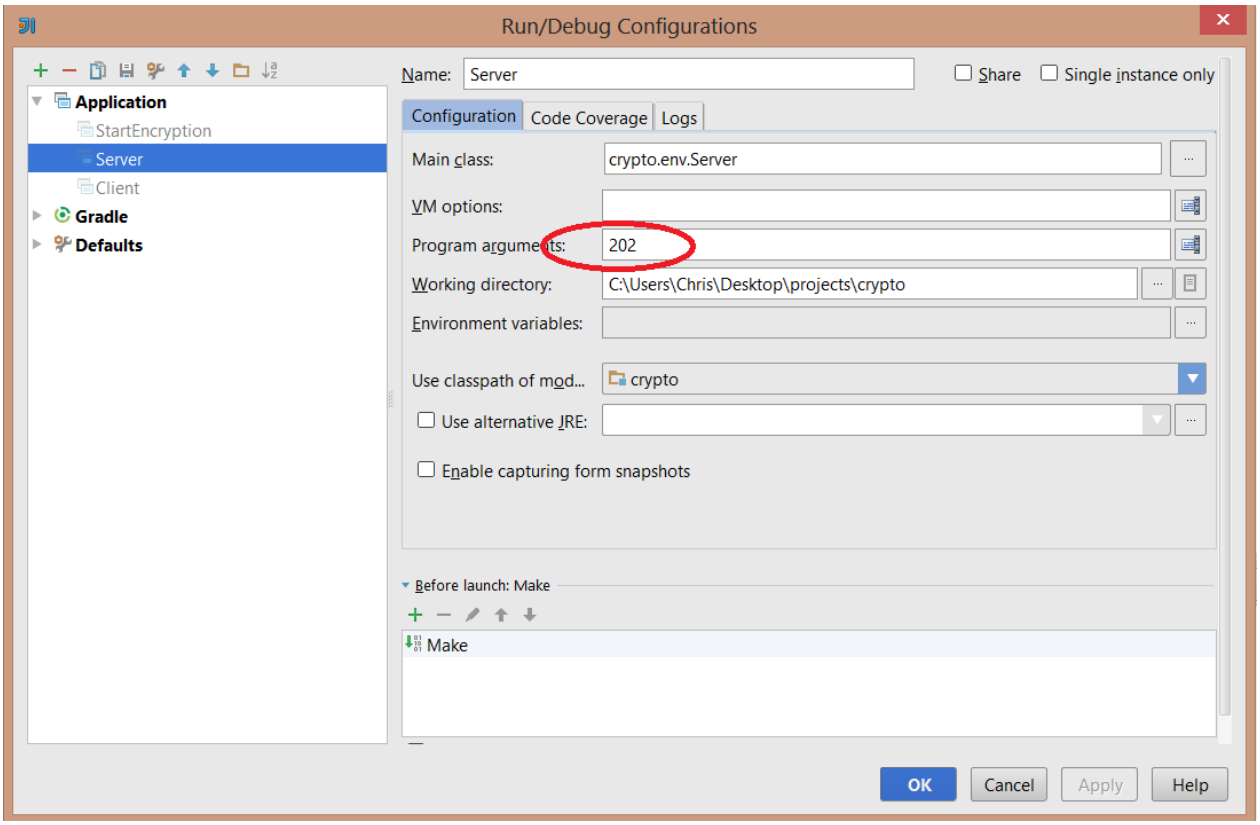
Testbericht

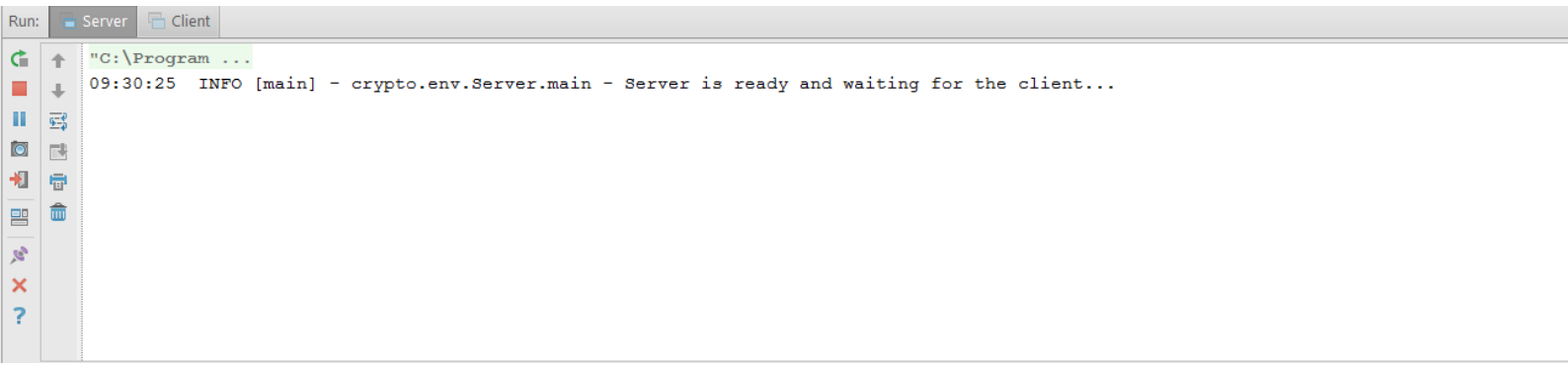
Secure Transmission:

Building artifacts in IntelliJ seemed quite problematic, so we decided to test our application in the IDE itself.

First of all the Server has to be started, with the following arguments:

Port … 202 (it can take any free port there is)



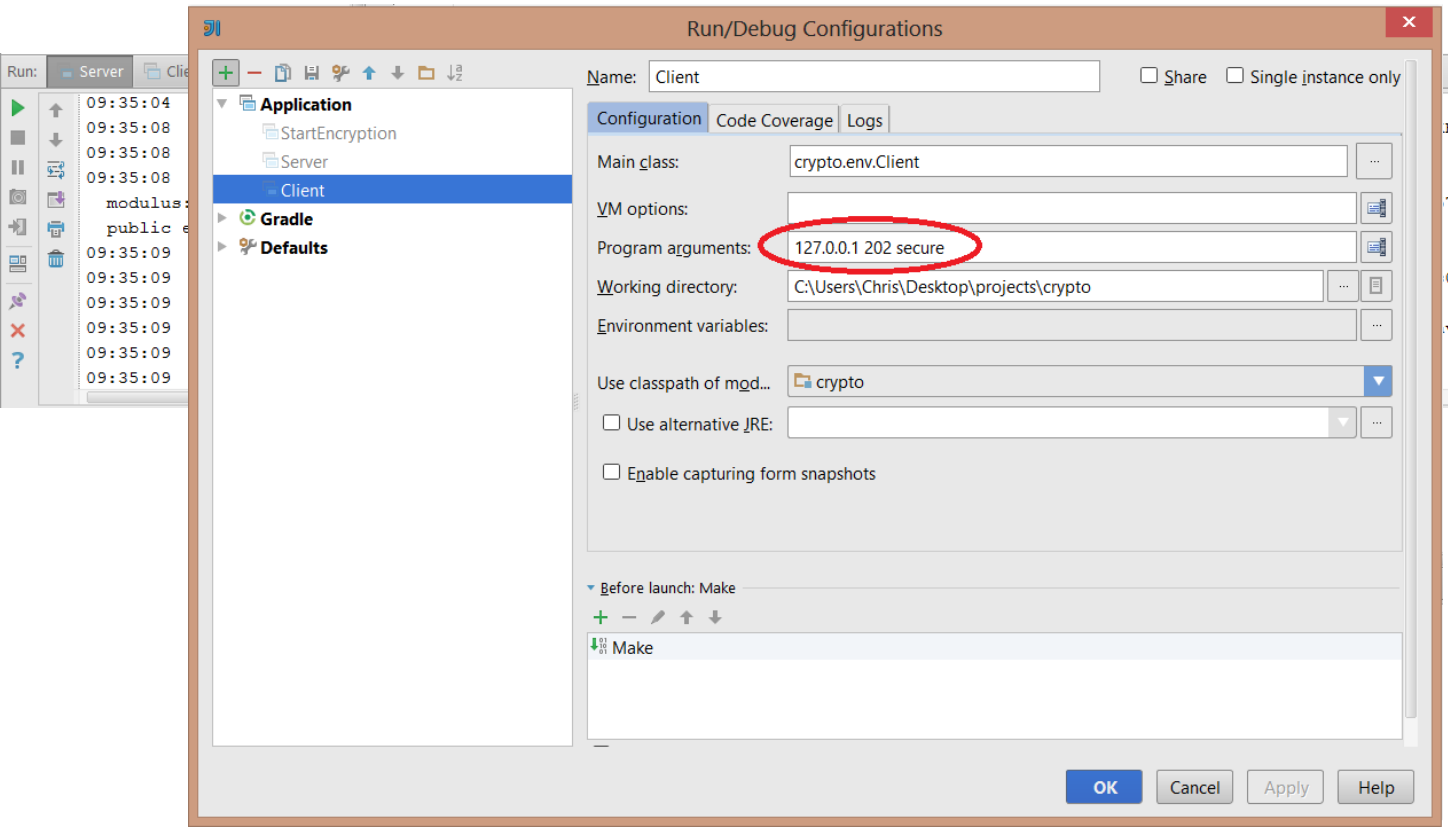


Secondly the Client has to be started with the following arguments:

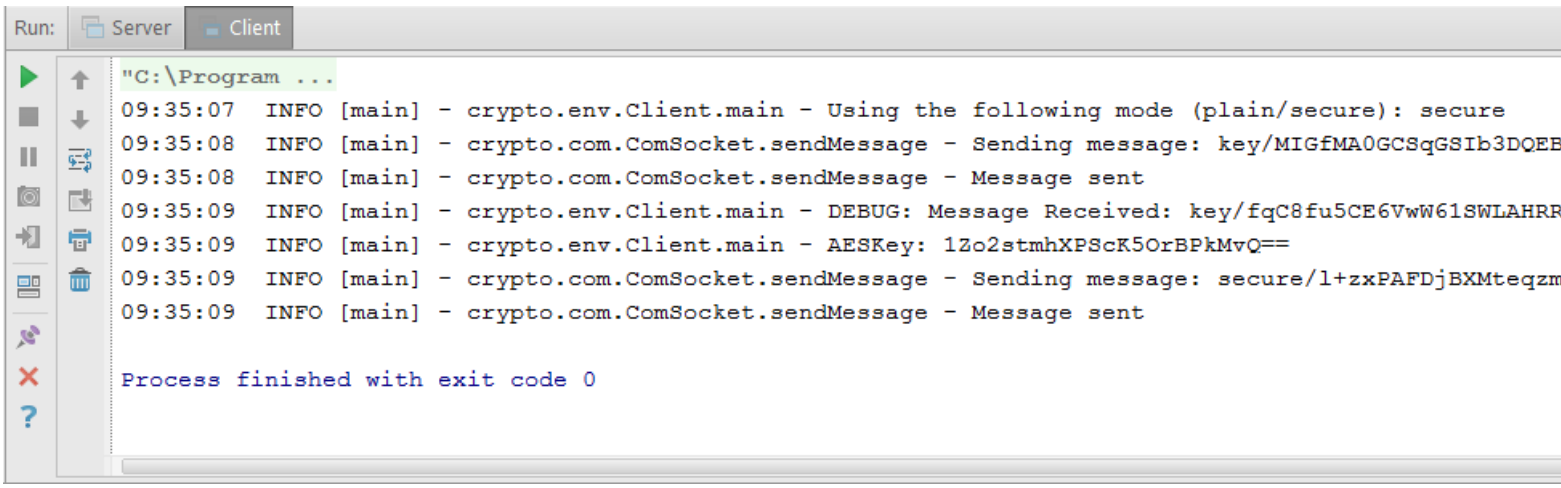
IP … 127.0.0.1

Port … 202

Mode … plain/secure

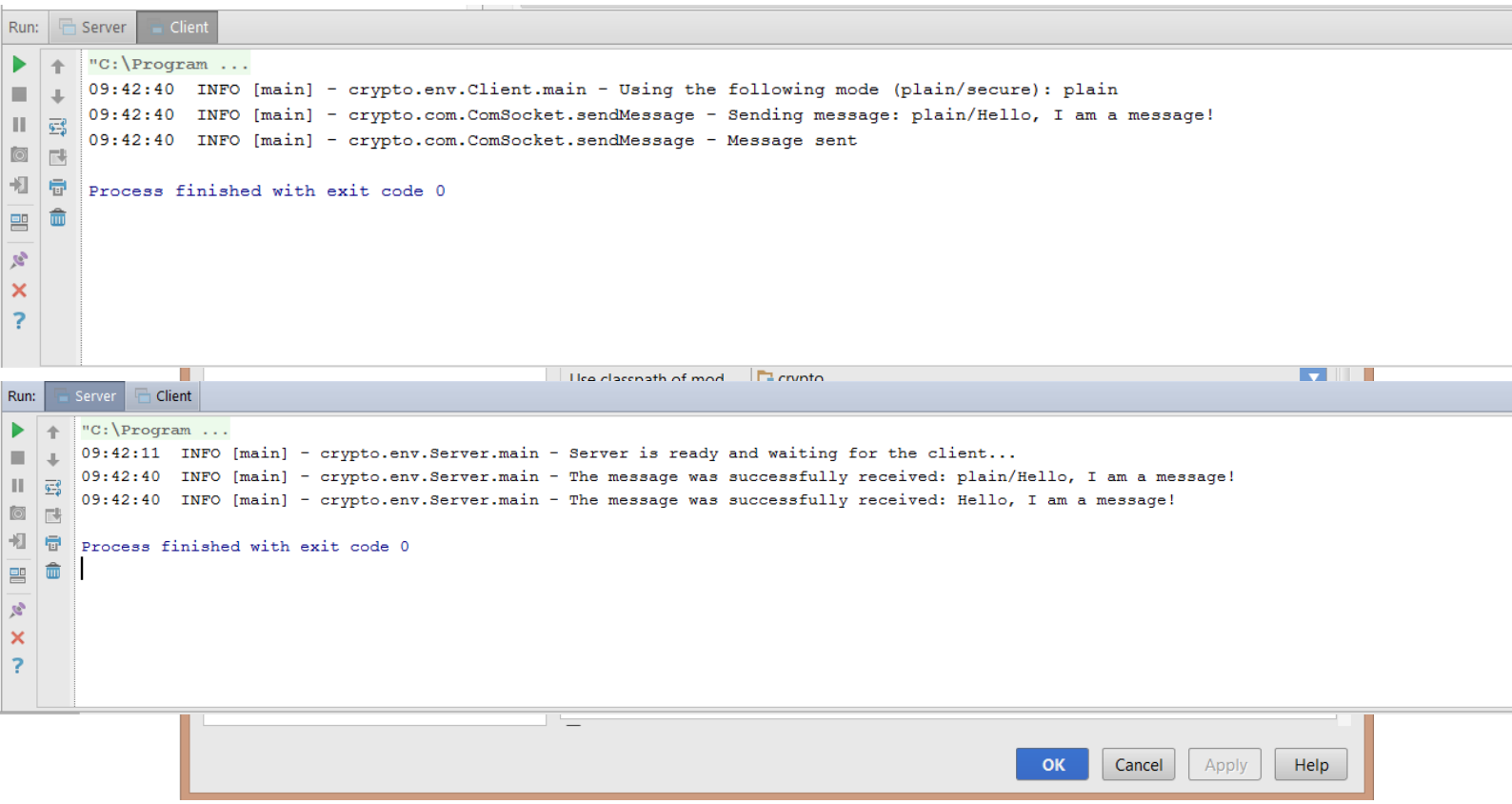


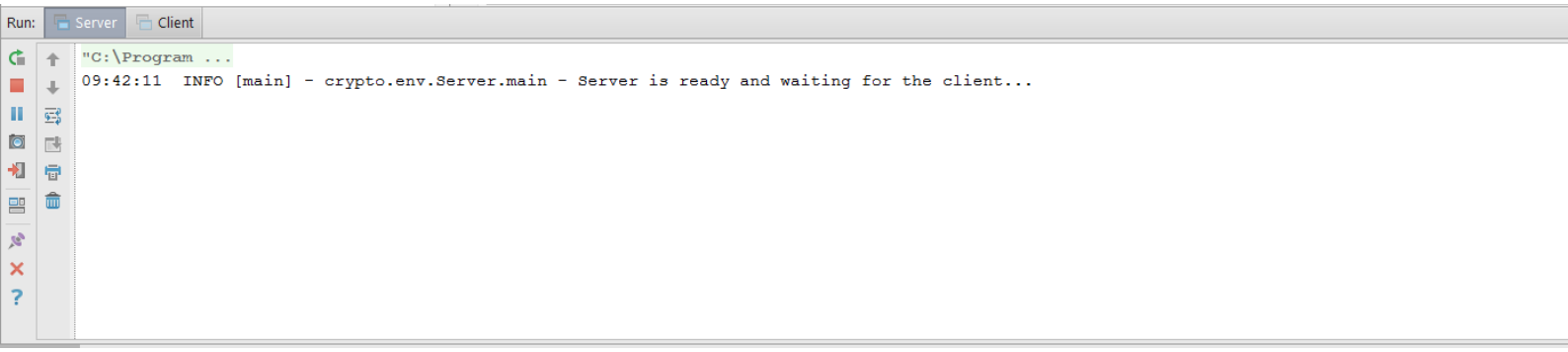
What's happening on the Server side?



Plain/Insecure Transmission:

The mode has to be changed from secure to plain in the Client Arguments:





Quellenangabe

[1] **Title,** Autor**,** Link

[2] **Sockets,** Java, http://docs.oracle.com/javase/tutorial/networking/sockets/definition.html

[3] **AES,** Margaret Rouse, http://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard

[4] **DES**, Margaret Rouse, http://searchsecurity.techtarget.com/definition/Data-Encryption-Standard

[5] **RSA,** Margaret Rouse, http://searchsecurity.techtarget.com/definition/RSA

[6] **Sniffer,** Bradley Mitchell, http://compnetworking.about.com/od/networksecurityprivacy/g/bldef_sniffer.htm