Assignment 7: Security

- This assignment will be introduced in a class meeting on December 6th, 2016 at 11:30 in seminar room 0.124.
- The groups will present the solution of this assignment on December 13th, 2016 at 11:30. We will meet in seminar room 0.124.
- You may work on the assignments in groups of up to three people (if possible, keep the same groups as in the previous assignments).
- All members of a group have to show up together for the grading of the assignment. For the grading, each group will need to have the source code ready for the discussion as well as the running implementation to be presented by the group. Moreover, each group member might be asked questions about the solution.
- If you have questions, send an email to adnan.tariq@ipvs.uni-stuttgart.de.

Task 1 – Secure File Transfer

Task 1.1 – Encrypted File Transfer

Assume you want to send files via the Internet from a sender *A* to a receiver *B* using a TCP connection. In order to ensure confidentiality, you decide to encrypt the bytes that are transferred over the TCP connection.

Implement the sender and receiver using the Java programming language and a suitable combination of symmetric and asymmetric cryptography. Make sure that your implementation supports files of arbitrary size (possibly much larger than the main memory of the sender or receiver machine). The encryption should be done “on-the-fly”. That is, encrypt the bytes just before they are sent over the network rather than creating an encrypted copy of the file on hard disk first. Moreover, pay attention to use efficient encryption algorithms to encrypt the (large) content of the file, and asymmetric cryptography to facilitate the exchange of (public) keys.

In the directory `/home/tariqan/assignment7/` on the server *netappsvm.informatik.uni-stuttgart.de* you will find the public and private keys of a sender *A* and a receiver *B*. You can assume that public keys are well-known and accessible by everyone (A and B), while private keys are only accessible by the owner (i.e. the private key of *A* is only accessible by *A*, and the private key of *B* only by *B*).

(Continued on next page →)

You can read the keys from hard disk using the following Java code:

```
final String PUBLIC_KEY_PATH = "key.public";
PublicKey key = null;

try {
      FileInputStream fis;
      fis = new FileInputStream(PUBLIC_KEY_PATH);
      ObjectInputStream in = new ObjectInputStream(fis);
      key = (PublicKey) in.readObject();
      in.close();
} catch (FileNotFoundException e) {
      e.printStackTrace();
} catch (IOException e) {
      e.printStackTrace();
} catch (ClassNotFoundException e) {
      e.printStackTrace();
}
```

Test your implementation by transferring the file `/home/tariqan/assignment7/video.mp4` from *A* to *B*. Compare the original file with the received (decrypted) file using the `diff` command. After decryption, the received file should be the same as the original file.

Task 1.2 – File Integrity, Authenticity, and Non-repudiation

Describe how the integrity and authenticity of the transferred file can be assured. You do not have to implement these concepts. Just describe the necessary extensions of the protocol.

Could the sender deny that he has sent the file (plausible deniability)?