STI 2017/2018

Practical Assignment #3

Programming and Security

1. Goals

- Add security to a Chat service
- Explore the usage of security/cryptography functions in Java

2. General description

The main goal of this assignment is to implement a secure Chat service[1]. A service of this type is very simple, and enables the various users of the service to send messages between each other, as Figure 1 illustrates. In a traditional Chat system, communications travel in the network without security applied. Also, such systems do not support mechanisms to authenticate users (clients), nor to verify the integrity and the authenticity of the messages received by the various communicating parties, among other security-related aspects, as we explore in the practical assignment.

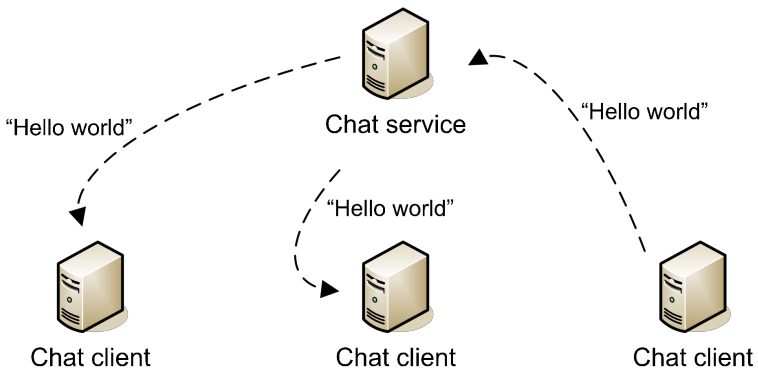

Figure 1 – The Chat service

3. A secure Chat service

Base components (client and server) of the Chat service

As a **base system on top of which security mechanisms will be applied**, the student may consider the source code (in Java) provided with the assignment. Considering the provided source code, the system is composed of a server and client programs, with the server accepting multiple connected clients and supporting the exchange of messages between the various

[1] Although traditionally this is not so, for the purpose of this assignment you should consider that this service requires a high level of security

clients. In the Chat system, the server broadcasts each message sent by a client to the other connected clients. A client may also send a message containing the string **".quit"** to the server, as a special code for the connection to be terminated. In this case, the remaining clients connected to the server are warned of the fact that a client is logging out of the Chat service. We must note that the main goal of the assignment is **not** to implement functionalities other than those required to secure the Chat service, but rather to use/implement security mechanisms/functionalities to protect the network communications and the various communication parties of the system, according to the requirements discussed next.

Security requirements to consider

The following security requirements and functionalities should be added to the base Chat system:

- **Confidentiality**

  Protect all communications (messages) exchanged between the server and clients of the Chat service. A symmetric cipher should be used for this purpose, in the usage mode and using the key size considered to be the most appropriate to the service at hand.

- **Authenticity**

  Guarantee that all messages exchanged in the system are authentic.

- **Integrity**

  Guarantee integrity for all communications in the system, with the main goal of identifying illegal modifications to messages.

- **Non-repudiation**

  Guarantee that all communications are unequivocally related to a particular entity (server or client of the service).

- Access control

  Refuse connections from particular clients, identified either by their IP address or by their identity in the service.

Other security-related functionalities

Other important aspects related with security are key management and the secure management of confidential information, as follows:

- **Key management**

  A system using symmetric cryptography is usually only secure as long as the cryptographic keys employed to support encryption and decryption are periodically changed/renewed. Other important aspect is how such keys are provisioned. Thus, you should design and implement a key management model with the following goals in mind:

  - ✓ Keys must be provisioned for the purpose of guaranteeing security in the service.
  - ✓ Keys employed to encrypt communications in the service should be periodically renewed.

- **Secure management of confidential information**

  Confidential or secret information (private keys, passwords, etc.) should be stored securely, whenever possible.

Security libraries

As previously discussed, the provided example code in Java may be used as the base Chat service on top of which security is implemented. For this purpose, the JCA (Java Cryptography Architecture) may be used. Among other documentation sources, you may consider the following:

- Java SE Security:
  http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136007.html
- Java Cryptography Architecture (JCA) Reference Guide:
  http://docs.oracle.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html

4. Delivery of the Practical Assignment

Important aspects:

- Although extra functionalities may be added to the Chat system (other than those required for the implementation of the required security-related mechanisms), they **will not be considered** while grading the assignment.
- Your code and report should properly describe and justify the security functionalities employed to protect communications in the system, as per the requirements here identified.

Your report should include the following information:

- The **security model** and **security procedures** implemented: how communicating entities are identified, how keys are distributed/managed, authentication and key management, quality of random seeds, etc.

- The cryptographic algorithms employed, as well as how they are used (usage mode, size of cryptographic keys employed, etc.)

- A description of the steps performed to verify the normal operation of the various security services implemented.

Delivery of the assignment:

- Please deliver an archive containing the report plus all source code files of your secure Chat implementation.

- Reports delivered without PGP will not be accepted.

- Limit date to deliver your assignment: 28/5/2018.

- Each day after the established final delivery date represents a 20% penalty on the final grade of the assignment.