**Integrated MSc Course on Informatics Engineering, DI/FCT/UNL**

**Computer Networks and Systems Security / Semester 1, 2019-2020**

**WORK-ASSIGNMENT #1 REPORT for Evaluation**

**A Peer-Group Oriented Chat using Secure Multicast Communication Channels**

**Authors:**

**Francisco Cardoso (**[**fj.cardoso@campus.fct.unl.pt**](mailto:fj.cardoso@campus.fct.unl.pt.z)**)**

Report

***Summary***

The objective of the project assignment is the design and implementation of a Secure Peer-Group Oriented Chat System, supported by Secure IP Multicasting Communication Channels. The work involves: the design analysis; development; and the preparation of a deployment package for demonstration. The system will be addressed from a provided Peer-Group Multicast Chat-System (with an initial implementation in Java language), not supporting the intended secure guarantees. In this report we present our implementation and the achieved objectives.

**Summary table of the TP1 implementation submitted for evaluation (fill with X), according to the Google Submission Form**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Coverage of the performed work TP1 | | YES | NO | Tested it works well | Tested, doesn’t work well | Doesn’t work |
| The work only addressed the implementation of the PHASE 1 | | **X** |  |  | **X** |  |
| The work addressed the implementation of the PHASE 1 and in this report we present the design and specification of the SAAHP protocol for the Phase 2 | |  | **X** | N/A | N/A | N/A |
| The work addressed the implementation of Phase 1 and Phase 2, bit the phases were not integrated | |  | **X** |  |  |  |
| The work involved the development and the integration of Phase 1 and Phase 2 | |  | **X** |  |  |  |
| URL of the GitHub Repo Project shared with henriquejoaolopesdomingos: | **https://github.com/fjscardoso/SRSC.git** | | | | | |

1. **Introduction**
2. **System Model, Architecture and Components**

Use this section to characterize form the system model and components presented in the initial statement, presenting specifically what is different from what is initially presented in the initial statement. **Adversary Model Considerations**

1. **Adversary model**

**3.3.1 Phase 1.** The adversary model for the project follows the typology of communication attacks against the underlaying channels supporting the SCMP and SAAHP, respectively IP Multicast channels and TCP/IP channels, as defined in the OSI X.800 framework (studied in lectures). Then, SCMP and SAAHP will include protection for passive and active communication attacks, requiring the following protections:

* Message confidentiality avoiding release of message contents (based on connectionless confidentiality arguments for SCMP and connection-oriented confidentiality for SAAHP);
* Message integrity (based on connectionless integrity assumptions for SCMP and connection-oriented integrity for SAAHP), protecting from message tampering, as well as, from traffic-flow tampering, including disordering attacks on the respective message flows);
* Authentication control services for message authentication controls
* Message replay protection;
* Protection against masquerading of endpoints or identity spoofing of communicating peers (given the

peer names or authenticated digital identifiers)

* Non-repudiation guarantees: each peer will maintain a local log of messages observed in each session

(with time-stamping controls and integrity proofs of all observed messages, as well as, the ordering observed integrity guarantees.

The following attacks are out-of-scope of the adversary model definition for the project requirements.

* + Traffic analysis with reconnaissance of supported protocols;
  + Denial of service attacks causing message-suppressions in the communication channel, channel

unavailability by communication disruptions, or unavailability of communicating peer-endpoints and their client applications;

Being out-of-scope of mandatory requirements, students are invited in designing and implemented improved security mechanisms, thought to mitigate (or minimize) the impact of those attacks.

The adversary model assumptions are focused on the protection of multicast communication channels and endpoints processing the SMCP and SAAHP protocols. Intrusion attacks against the computers running the **SecureMChatClient** causing incorrect modifications on the runtime environment, or originated by the injection of malicious code changing the correct behavior of the software execution environment, are not considered in the adversary model definition.

**3.3.2 Phase 2.**

1. **Phase 1 – Secure Multicast Communication Protocol (SMCP)**

I added the size of the packet sent to the SMCP socket(magic number, message type, username and message) to the secure payload in order to be able to read the exact bytes when the socket receives the message.

* 1. **Configuration file for MchatClient applications protected by SCMP**

Didn’t address

* 1. **Keystores**

Describe here the context of the used keystores and used storetypes. You can put the context of the keystores using keytool with option –list

You must report also the password(s) you use to protect the keystore or keystore entries.

**4.3 Running the Client as a Standalone Application**

Report here ho to run your client application MchatClient, according to your final deployment, namely:

* executable jar signed application
* executable jar not-signed application
* executable java application (main class)

You must have here all the required setup or any relevant information allowing to run your final deploy-able standalone application, not necessary only how to run it from the Eclipse IDE (or other IDE) environment.

* 1. **Tested Cryptographic Parameterizations**

Didn’t address.

satisfied by the cryptographic constructions in your message format.

1. **Conclusions and Final Remarks**

Use this to summarize your main conclusions and to present your final remarks on the implementation of the TP#1.