**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:**

**Name Surname (**[**bv.santos@campus.fct.unl.pt**](mailto:bv.santos@campus.fct.unl.pt)**), Name Surname (**[**fmpi.santos@campus.fct.unl.pt**](mailto:vvv@yyy.kk)**)**

Report

***Summary***

You can write this short report in English or in portuguese. First you can address here your own abstract, as you want. Something like this can be used as an initial reference for you final abstract: 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/bvsantos/44935__45411.git> | | | | | |

1. **Introduction**

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.

Our team implemented phase 1with AES and we present the design and specification of the SAAHP protocol for the Phase 2.

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

Nothing is different from the initial statement.

1. **Adversary model**

**3.3.1 Phase 1.** Our team implemented security for the following attacks:

* 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);
* Message replay protection;
* Protection against masquerading of endpoints or identity spoofing of communicating peers (given the peer names or authenticated digital identifiers)

**3.3.2 Phase 2.** If your implementation addresses the phase 2, describe what you consider is different from the adversary model for the phase 2. You must only consider this section of you implemented phase 2 specifications or specification and implementation. If you consider that the adversary model conditions are exactly the same, you must only say it.

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

Beside the initial specifications of the SMPC message format our team added:

* sIDSize – sIDSize
* SAtributesSize – hash length
* Inside the SecuredPayload we added userNameLength and messageLength
  1. **Configuration file for MchatClient applications protected by SCMP**

<?xml version="1.0" encoding="UTF-8"?>

<arr>

<A>

<SID>Chat of Secret Oriental Culinary</SID>

<SEA>AES</SEA>

<SEAKS>256</SEAKS>

<MODE>GCM</MODE>

<PADDING>NoPadding</PADDING>

<INTHASH>SHA-256</INTHASH>

<MAC>HMacSHA256</MAC>

<MAKKS>256</MAKKS>

</A>

<B>

<SID>Secret Chat of the Long Night of Horrors</SID>

<SEA>RC6</SEA>

<SEAKS>256</SEAKS>

<MODE>CTR</MODE>

<PADDING>NoPadding</PADDING>

<INTHASH>SHA512</INTHASH>

<MAC>HMacSHA512</MAC>

<MAKKS>512</MAKKS>

</B>

<C>

<SID>Secret Chat of the Long Night of Horrors</SID>

<SEA>Blowfish</SEA>

<SEAKS>448</SEAKS>

<MODE>CBC</MODE>

<PADDING>PKCS5Padding</PADDING>

<INTHASH>SHA1</INTHASH>

<MAC>DES</MAC>

<MAKKS>64</MAKKS>

</C>

</arr>

* 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.

* 1. **Running the Client as a Standalone Application**

To run our application:

Install Bouncy Castle(<http://tomee.apache.org/bouncy-castle.html>)

Run: java -jar <File-location>\44935-45411MCHAT\target\44935-45411MCHAT-1.0-SNAPSHOT.jar

Or

Open as a maven project on a IDE and run

* 1. **Tested Cryptographic Parameterizations**

Include here the list of cryptographic parameterizations you checked in the experimental evaluation of your development in phase 1.

If you didn’t address

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.