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:
Tiago Oliveira (talm.oliveira@campus.fct.unl.pt),
Tomás Pessanha(t.pessanha@campus.fct.unl.pt)

Summary

The objective of this assignment was the design and implementation of a Secure Peer-Group Oriented Chat System, supported by Secure IP Multicasting Communication Channels.

The work involved the design, analysis, development; and the preparation a demonstration.

The assignment was addressed with a provided Peer-Group Multicast Chat-System , 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				
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/tomasPs/SRSC-TP1-1920**				

1. **Introduction**

In this assignment only phase 1 was addressed as a consequence only the following things were addressed: The implementation and extention of the SCMP, the implementation of the SCMPCSockets, the implementation of the secureMChatClient and the support for different cryptographic parameterizations of the application.

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**

3. **Adversary model**

3.3.1 **Phase 1.**

The adversary model did not change from the original as such the protections remain as such:

-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.

Additionally the out-of-scope attacks mentioned were not taken into consideration.

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

The format of the SMCP messages is the same as in the initial specifications with the only change being when there is the need to use an IV for the encryption, then a new parameter is added with the IV in the message. This lets the other clients know what the IV used was without the need to send another message

4.1 **Configuration file for MchatClient applications protected by SCMP**

```
<config>
        <endpoint ip="224.5.6.7" port="9000">
                <SID>Chat of Secret Oriental Culinary</SID>
                <SEA>AES</SEA>
                <SEAKS>256</SEAKS>
```

```
             <MODE>CBC</MODE>
            <PADDING>PKCS5Padding</PADDING>
            <INTHASH>SHA256</INTHASH>
            <MAC>HMacSHA256</MAC>
            <MAKKS>256</MAKKS>
    </endpoint>

    <endpoint ip="252.10.20.30" port="12224">
            <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>
    </endpoint>

    <endpoint ip="230.100.100.100" port="6666">
            <SID>Secret Chat of the Long Night of Horrors</SID>
            <SEA>Blowfish</SEA>
            <SEAKS>448</SEAKS>
            <MODE>CBC</MODE>
            <PADDING>PKC5Padding</PADDING>
            <INTHASH>SHA1</INTHASH>
            <MAC>DES</MAC>
            <MAKKS>64</MAKKS>
    </endpoint>
</config>
```

4.2 Keystores

The keystore of type 'jecks' was used for the clients keys. The convention of the keys aliases is 'ip:port-type', so for an example: "224.5.6.7:9000-se" for the symmetric key used the types allowed are: -(se) for symetric keys -(mac) for mac keys. The keystore comes with 3 endpoints included, using the endpoints given by the intructor in the Annex 3 file.
The password for the keystore is 'g11srsc' And the password for each key is the endpoint 'ip:port'.

4.3 Running the Client as a Standalone Application

The included SecureMchatClient.jar file is not signed and can be run with the normal command of: java -jar SecureMchatClient.jar <nickusername> <grupo IPMulticast> <porto> { <ttl> } .
There is an included readme file next to the jar for convenicence.

4.4 Tested Cryptographic Parameterizations

We only used AES for our symmetric encrpytion.

We used the following modes: -CBC -EBC -CTR

And for hashing we tested with: -SHA256

For the Macs we tested with: -HMacSHA256

5 Conclusions and Final Remarks

We feel that during the develoment of this assignment we learned much about the nuances behind connection security, and the many things we mentioned in the introduction.