University of Cape Town
Department of Computer Science
CSC 4026Z
Network and Internetwork Security 2020
Practical

1 Introduction

The following tutorial is the practical component for this course and is to be completed in groups of four. Please send the names of your group members to Andre Lopes, who is acting as the TA for the course this year. His contact details are in Section 6.

The objective of the practical is get experience with crypto functions, and in particular to validate key authenticity and to simulate / replicate the message confidentiality and authentication aspects of PGP (see slides Part 5, Slide 5 "PGP Cryptographic Functions" as a guide). The Certification Authority role required is more X.509-like than PGP-like.

2 The task

Create two **Client** Applications that will initially exchange and validate each other's public keys, which have been issued by a Certification Authority which they both trust. They should then transmit messages to each other, using the shared key, private key, hashing and compression functions, in the same manner which PGP would do.

The Client Applications are expected to have:

- A private and public key pair of their own
- The public key of the Certification Authority
- A certificate (containing the client's own public key) signed by the Certification Authority

Take note that the trusted third-party interaction (getting the certificates from the Certification Authority) need not exist as an application in your final submission. You may instead pre-create the certificates needed for authentication of the public keys, and store them where they can be accessed by the clients.

The Client communication system (based on UDP or TCP) should be established so that at least one secured message is sent from one of the clients to the other. The manner in which communications are initiated can vary (you may find it easier to have one client listen while another client initiates communications).

Public / private key pairs should be created for use with RSA (for public key encryption) and a shared key generated for use with DES, AES etc. (for shared key encryption).

For testing purposes please include debugging statements. i.e. output the encrypted text messages along with the decrypted text to the console so that when it is run we can see what it is doing.

A short write-up (no more than 5 pages) is required to explain your implementation, choice of crypto algorithms, key management, communication connectivity model, testing procedure and assumptions made. Be sure also to document how to execute / run the program(s) submitted (this can be written as a separate document if needed).

3 Assessment

Overall system design and functionality to achieve stated goal [**40%**]

To be determined based on running the system and write-up

- Communications implementation [**20%**]
- Security implementation [**30%**]
- Evidence of testing [**10%**]

4 Cryptographic and Implementation Details

The prescribed programming language for this practical is Java. We do **not** expect you to code your own encryption libraries and thus encourage you to use encryption libraries such as Bouncy Castle for Java.

The Java API has some excellent documentation on the security libraries included in the SDK.

Cryptography details:

For asymmetric encryption, we encourage you to use the RSA algorithm in ECB mode with PKCS1 padding. The algorithm specification string in Java is "RSA/ECB/PKCS1Padding".

For symmetric encryption, we encourage you to use AES algorithm in CBC mode with PKCS5 padding. The algorithm specification string in Java is "AES/CBC/PKCS5Padding".

As mentioned previously, for evidence of testing and insight, you should ensure that sufficient 'debug' statements are included so that the ciphertext and message components can be viewed.

5 Due Date

The system along with the write up will be due on **Friday 12th June 2020** at **23:59**, and is to be submitted on Vula. Usual late hand in penalties will apply.

The practical component makes up 40% of the module assessment.

6 Queries

Any queries should be directed to Andre Lopes LPSAND004@myuct.ac.za. Make sure to include NIS2020 in the subject of your email.