Segurança de Redes e Sistemas de Computadores 2019/2020
1º Semestre Aula Prática (LAB 4 – LAB 4.2)

Guidelines

Objetivo: In this Lab we can observe how to program with Asymmetric Cryptography (Java/JCE) to support Digital Signatures.

Hands-On Activities

1) 1-RSA-DAS Signatures

You can see how to build and how to verify (or validate) standard digital signatures using RSA with relevant padding patterns. In the provided examples we use PKCS#1 Padding.

You can see how to build and how to verify (or validate) DSA (or DSS) Digital Signatures

You must observe the sizes for signed objects, comparing with the signature construction and to understand why you obtain the observed sizes (rational from Lecture Classes).

2) 2-ECCDSA-Signatures

In the provided code we can see how to program with ECDSA Digital Signatures (using Elliptic Curves (ECC – Elliptic Curve Cryptography) to build and to verify those signatures.

Again, you must observe the sizes for signed objects, comparing with the signature construction and to understand why you obtain the observed sizes (rational from Lecture Classes).

3) 3-KeyExchangeRef

In the provided examples we will discuss the technique for materializing Secure Envelopes for secure distribution and establishment of secret parameters between principals. In this case we apply the technique for the distribution of symmetric Keys (Symmetric Cryptographic Algorithms) and then using the protected keys that can be exchanged to be used for message or data encryption. The examples are focused in the use of RSA to process the secure envelopes. However, the technique can be applied to any other Asymmetrci Algorithm providing support for encryption/decryption.

4) 4-Wrapping-RSA-Keys

Here we show a technique (called Key-Wrapping): the objective is to protect a Private Key (from a Key Pair – in the example we use RSA keys) encrypted with a symmetric cryptographic method (in the case we use AES). Remembering previous Labs, we can also use Password-Based Encryption constructions for the same purpose.

5) KEYTOOLS-KEYSTORES

Here you can find different demos, for the manipulation of Java keystores when using asymmetric cryptography and related keypairs. You have examples on the use of the Keytool Java tool to generate <private,public keys> and how to store and manage such keys in keystores (for storage of the keypairs) and how to export the public keys to trusted stores or to generate public key certificates.