Testing Crypto 1.0

This document contains all the necessary information to execute the suite of tests for the packages javax.crypto and javax.crypto.spec

In order to be able to perform the tests it is necessary to have the JDK 1.5.0 (or superior) and Eclipse installed. Eclipse must be shaped with compliance for java 5.0

Later on, the projects TestingUltilities and CryptoJunitTesting of the cvs of testing must be obtained.

In the security file jre/lib/security/java.security the provider BouncyCastle and Cryptix must be registred (jar files are included), adding this line:

- security.provider.<corresponding number >=org.bouncycastle.jce.provider.BouncyCastleProvider
- security.provider.< corresponding number >=cryptix.jce.provider.CryptixCrypto

The tests can be executed individually by means of the plugin of JUnit of Eclicpse, the provided suite runner might also be used.

The tests are found in the package ar.org.fitc.test.math.main and they are divided as follows:

AllPerformanceTests: measures the execution time for each Junit method of the testcase, including Setup() and tearDown(), with a given iteration number.
LaunchAllCryptoTests: Run all unit tests for javax.crypto and javax.crypto.spec package (black and white box).
LaunchBouncyCastleProviderTest: Runs a test bag which includes all the ciphers from the given provider (BouncyCastle).
LaunchCryptixCryptoProviderTest: Runs a test bag which includes all the ciphers from the given provider (CryptixCrypto).
LaunchSunJCEProviderTest: Runs a test bag which includes all the ciphers from the given provider (SunJCE).
LaunchIntegrationTests: Contains three integration tests, the first one is one clientserver encrypted communication. This is done by making a key exchange before starting the communication. The second one, is a key exchange method between multiple parts. And the last one, shows the blocking of a stream when a block cipher is used.

Notes:

- DESKeySpec: Two weak keys were found, but Sun's does not consider them as weak keys. The two keys are: "1F1F 1F1F 0E0E 0E0E" and "E0E0 E0E0 F1F1 F1F1"
- When we try to instantiate a cipher object, by using the static method getInstance() and passing the following arguments: algorithm, mode and a not valid padding. According to the specs, in this case a NoSuchPaddingException should be thrown but Sun throws NoSuchAlgorithmException.
- Some of the tests may fail if the unlimited strength policy file is not installed.