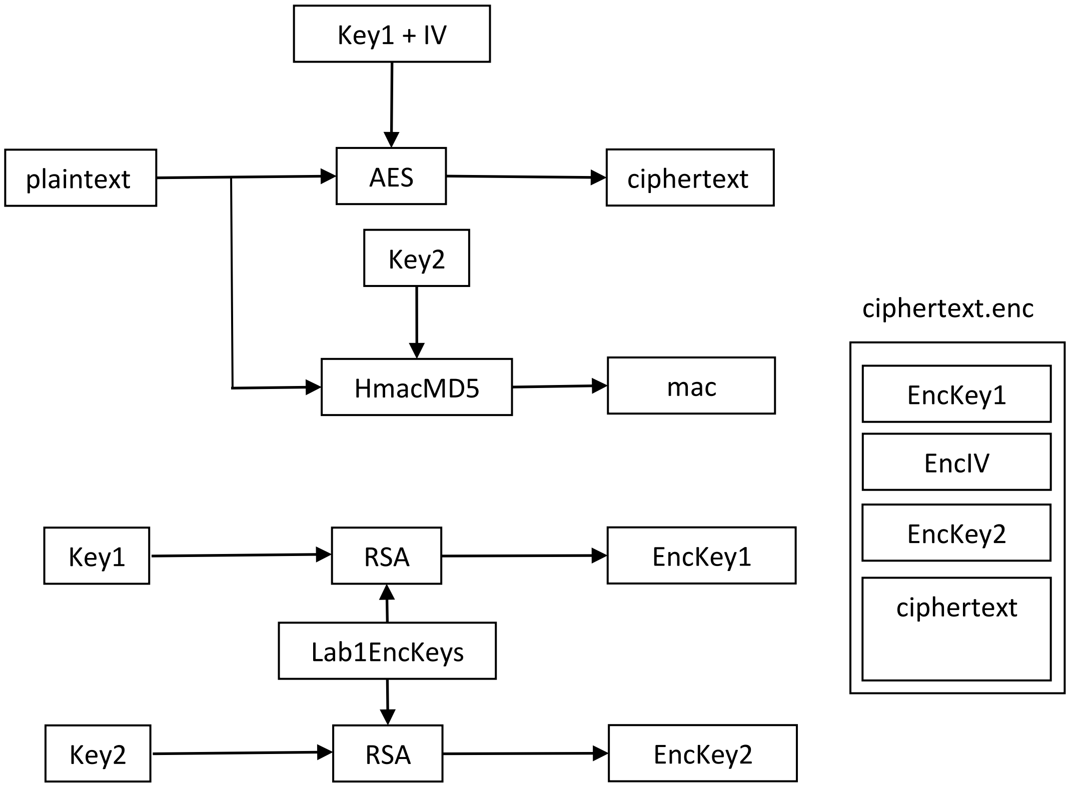
Laboratory exercise 1, Computer Security

A text message has been encrypted according to the figure below. In this lab you should do the following: Recover the plaintext, and verify the integrity of the message. To verify, both message authentication code and digital signature will be used. Before you proceed you need to familiar yourself with the examples on how to use encryption in Java that were given in the lectures and also on the exercises. Furthermore, you need to know the concepts of symmetric encryption, and public key systems for encryption of symmetric keys and digital signatures.



On ‘Itslearning’ you can find a file lab1Sec.zip. In the zip-file you have the following files:

Lab1Store – A keystore with a private key for decryption

Lab1Sign.cert – Certificate with public key for verification of digital signatures

Ciphertext.enc – Encrypted file

Ciphertext.enc.sig1 and Ciphertext.enc.sig2 – two digital signatures

Ciphertext.mac1.txt and ciphertext.mac2.txt – two message authentication codes

Task 1: Get Key1, IV and Key 2

1. Split the content of ciphertext.enc into its different parts. The structure of the encrypted file is the following: First we have 128-bytes (1024 bits) of a symmetric key encrypted with RSA using the public key "lab1EncKeys" then we have 128-byte of encrypted IV-value to be used in the decryption of the data. Before the actual encrypted data, there is another block of 128-byte containing encrypted key for a HmacMD5. Finally we have the encrypted data. The encryption is made with AES in CBC mode, using PKCS5 padding.
   (The reason that the encrypted symmetric keys have 128 bytes instead of 128 bits is because Java RSA algorithm pad the keys to the same size of the RSA key before encryption)
2. To get the keys and IV to be used to decrypt we need a private key. The key is stored in the keystore "lab1Store". The password to access lab1Store is "lab1StorePass". The alias for the private key is "lab1EncKeys" (bad name as already used to name its associated public key), and the password is "lab1KeyPass".

Task 2: Get the plaintext

When Key1 and IV are recovered you can decrypt and get the message.

Open question:

Why is it not a good idea to simply encrypt the plaintext with the receiver's public key? Why bother to generate Key1, IV, and encrypt them?