Computer Security

Lab 1 and 2

Some extra info

Sender

Key Pool:
- Public/private key pair Pu_s and Pr_s,
- Receiver's public key Pu_r,
- AES Key: Key1
- MAC key: Key 2
- Initial Vector: IV

1. Get a plaintext
2. Generate one-time keys and random number Key1, Key2 and IV
3. Get a ciphertext by applying AES encyption (with Key1 + IV) on the plaintext
4. Get a MAC by applying HmacMD5 algorithm (with Key2) on the plaintext
5. Encrypt Key1, Key2 and IV using Receiver's public key Pu_r
6. Sign the plaintext with the Sender's private key Pr_s

Receiver/YOU

Key Pool:
- Receiver's private key Pr_r: stored in Lab1Store
- Sender's public key Pu_s: in the certificate Lab1Sign.cert
- Encrypted versions of Key1, Key2 and IV

1. Get ciphertext

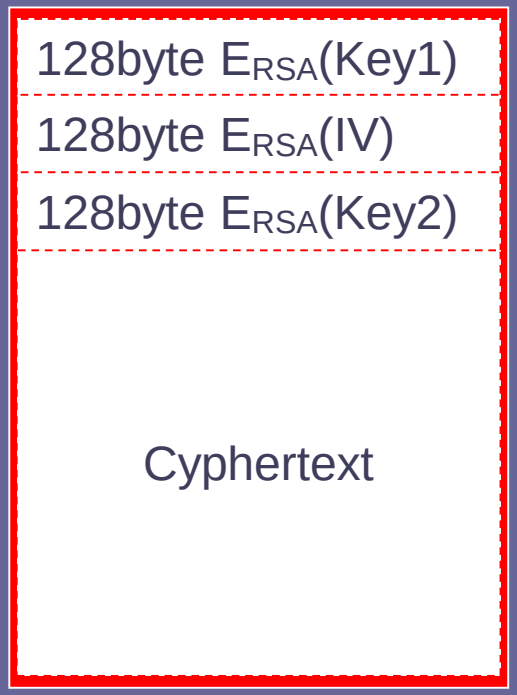
2. Get ciphertext.mac.txt

3. Get encrypted keys EncKey1, EncIV and EncKey2

4. Get dignal signatures ciphertext.enc.sig

Step 1

Ciphertext.enc split into 4 parts




```
byte[] encKey1=new byte[128];
```

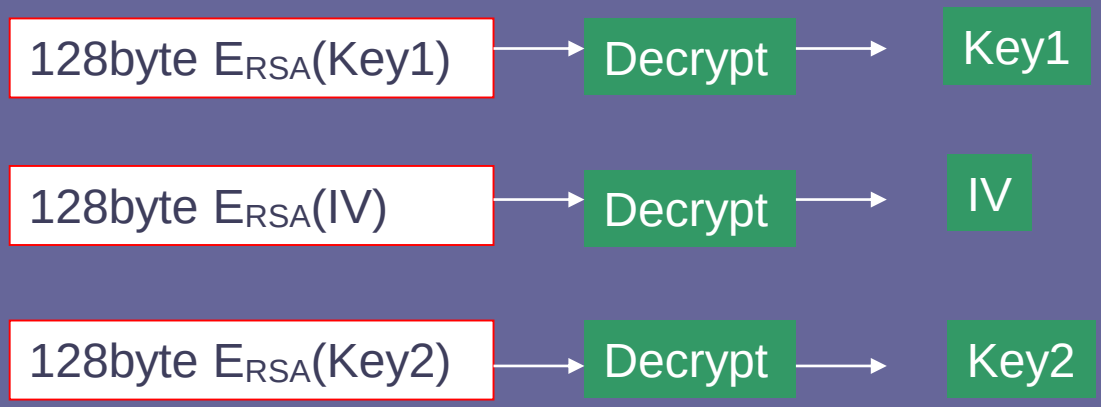
```
File f = new File("ciphertext.enc");
FileInputStream fis = new FileInputStream(f);
fis.read(encKey1);
```

Google ”FileInputStream java” and read about different read methods of that class

Step 2

Decrypt the keys and IV

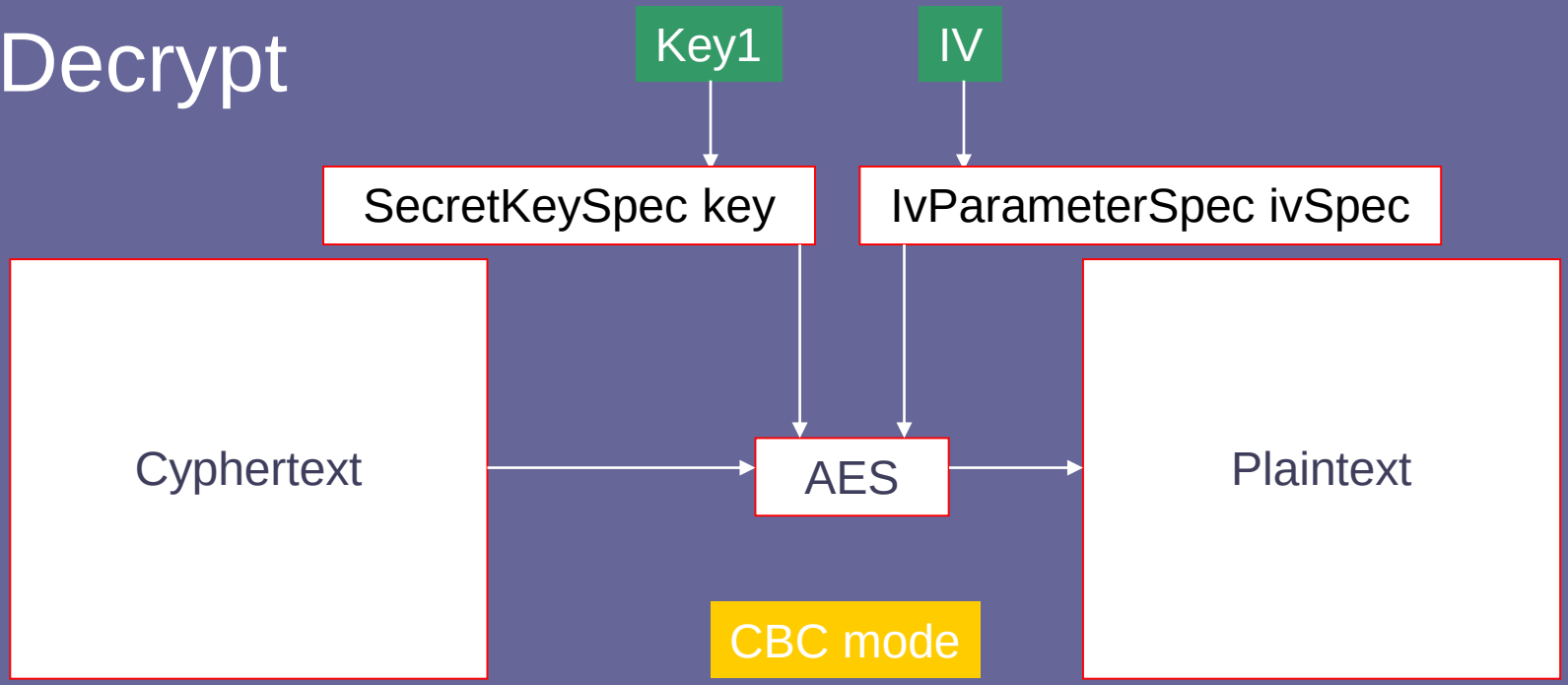
There is code to load keys from keystore file




```
byte key1[] = new byte[128];
Cipher rsaDec=Cipher.getInstance("RSA");
rsaDec.init(Cipher.DECRYPT_MODE,
Lab1EncKey);
key1=rsaDec.doFinal(encKey1)
```

Step 3

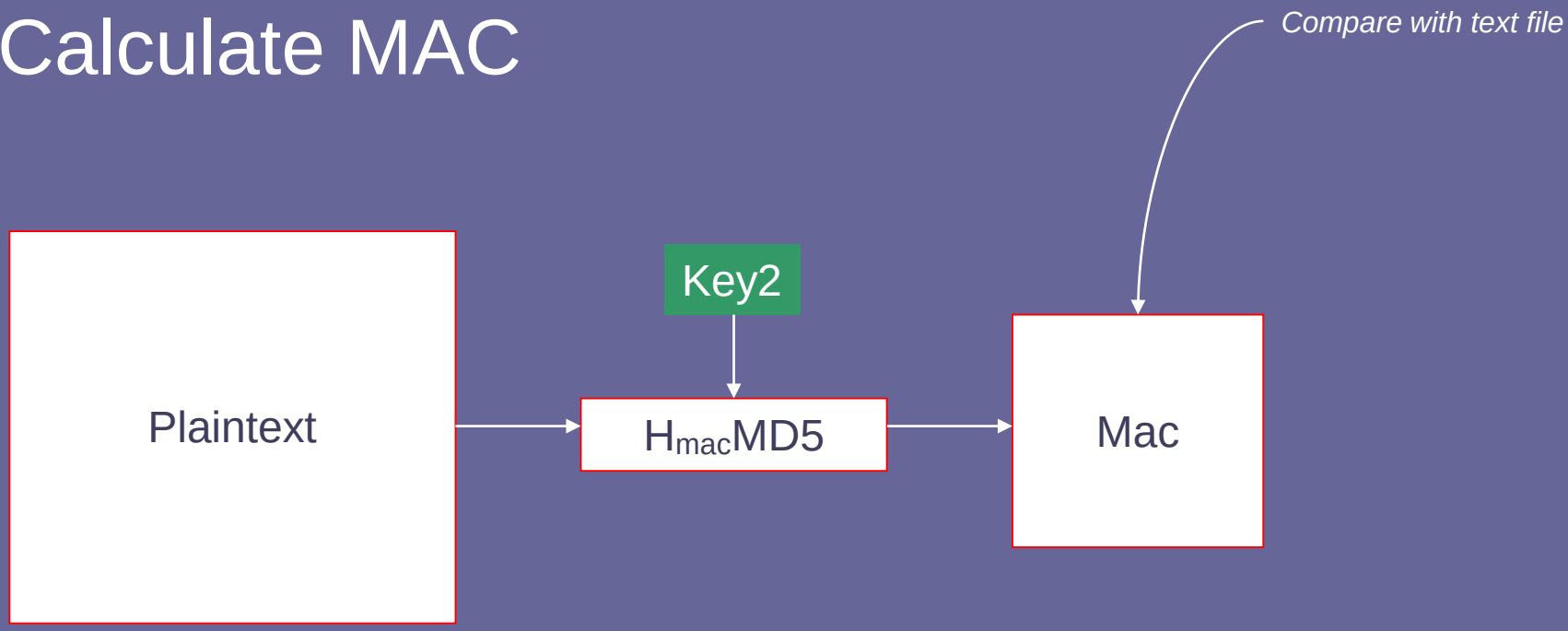
Decrypt



Cipher aesDec= ??
SecretKeySpec key = ??
IvParameterSpec ivSpec = ??

Step 4

Calculate MAC



Step 5
Read the Public Key

The Factory method:

```
FileInputStream readPuKey = new FileInputStream(“lab1Sign.cert”);
CertifiicateFactory cf = CertificateFactory.getInstance(“X.509”);
Certificate certificate = cf.generateCertificate(readPuKey);
PublicKey puKey = certificate.getPublicKey();
```

You can also use the keystore method,
but firstly load the certificate into a keystore.

Step 6

Digital Signature