COMP 232: Cybersecurity
Lab session 2

Lecturer: Alexei Lisitsa

For this session, I should like you to try some simple Java programs implementing DES encryption/decryption with JCA/JCE libraries.
Notice: no specific IDE or other development tools are assumed. All exercises can be done using simple editors and command line for compiling and executing programs.

1 DES encryption

Download, compile and run the following simple programs available at the web page `www.csc.liv.ac.uk/~alexei/COMP232/index.html` of COMP232 course:

- DES encryption in ECB mode;
- DES encryption in CBC mode with an inline IV;
- DES encryption in CBC mode with IV generated by Cipher object.

Look into the code and see how the encryption is implemented.

Task 1 Using the programs above (and their modifications) demonstrate the advantage of CBC mode over ECB mode.

2 Password-Based Encryption (PBE)

Notice that in the examples of programs given in the previous section the keys used in encryption were generated automatically without any user input. In many cases we would like to implement password-based encryption.
Password-based encryption mechanisms in JCE are based on cryptographic *haching* mechanisms. In short, a password and a *salt*, which is a random data is fed into a mixing function (a kind of hash function) which is applied iteratively the number of times defined by *iteration count*. All this happens in the Secret Key Factory object. The result is used then to create a key in the Cipher object. To decrypt one has to know a password (secret piece of data), the salt and iteration count (these two are usually not considered as secret and may be known to the attacker).

For this week exercise, try the following

Task 2

- Download the `Password-based encryption program, PBEs.java` from `http://www.csc.liv.ac.uk/~alexei/COMP232/index.html` and axiliary utility program `Util.java`.
- Compile both programs (e.g. `> javac *.java`), execute PBEs and explore the code to find out how to do password-based encryption.
- Try then to change iteration count to a smaller value and see how this affects the execution time.
- Modify the program, so it can decrypt the encrypted message.

Implementation of decryption is very similar to the case of encryption. You need to do the same preparatory steps as for encryption (including salt and iteration count initialization, creation PBE parameter set, initialization of password, etc) The only difference is that initialization of PBE Cipher should be done differently, as shown in the fragment below:

```
              //  Initialize PBE Cipher with key and parameters
pbeCipher.init(Cipher.DECRYPT_MODE, pbeKey, pbeParamSpec);

// decrypt the ciphertext
byte[] plaintext = pbeCipher.doFinal(aCyphertext);
String StringPlaintext = new String (plaintext);
```

3 After the Lab: from DES to AES

Once you have tried all the above, try to repeat Task 1 above with AES rather than DES encryption algorithm. Essentially you will need to replace "DES" with "AES" everywhere. In the case of difficulties please consult the Reference Manual of JCA/JCE, which contains fragments of code illustrating the use of AES. Please notice that password-based encryption for AES implemented in JCA/JCE is limited to the key size 128 bits. You can further explore the sample programs for password-based AES encryption available at the web page of module.

References

JCA/JCE Reference manual can be found at
`http://java.sun.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html`