Cibersegurança - Segurança em Software (Módulo 2)

Trabalho Prático - Parte 2

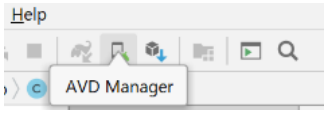
Objetivos:

- Compreender a estrutura de um APK
- Saber fazer o *repackaging* de uma aplicação Android
- Saber obter um certificado digital a partir de um *keystore* Java
- Saber utilizar algoritmos simétricos de uma biblioteca criptográfica

❖ Preparação

Pré-requisitos: Ambiente integrado de desenvolvimento Android Studio com ferramentas de linha de comando - https://developer.android.com/studio

Para evitar o uso de dispositivos reais, as questões seguintes assumem a existência de um Android Virtual Device (AVD). Existindo o AVD "cslab", o mesmo pode ser lançado com o emulador Android usando o comando:




```
emulator @cslab
```

Mais informações em:

https://developer.android.com/studio/run/emulator-commandline

https://developer.android.com/studio/command-line/sdkmanager

https://developer.android.com/studio/command-line/avdmanager

1. Considere o APK `RepackagingLab.apk` em anexo:
    a. Instale o APK no AVD e veja a imagem que aparece no ecrã inicial;
    b. Realize o repackaging do APK, mudando a imagem do ecrã inicial. Note que a imagem é um recurso presente em `[apk]\res\drawable`;
    c. Assine e instale o novo APK.

    Entrega**:**

    - evidências dos passos executados;
    - novo APK;
    - certificado associado à chave privada que assina a nova versão do APK.

2. O `Lab2_2.apk` em anexo é uma aplicação que pede uma frase ao utilizador e verifica se é ou não a frase correta. No entanto, a frase correta foi cifrada e guardada juntamente com a respetiva chave, no código fonte da aplicação.
    a. Instale o APK e teste a aplicação;
    b. Descompile o APK e analise o código. Na classe `MainActivity`, o método `void verify(View v)` é chamado para verificar se a *string* introduzida na caixa de texto é a correta. A partir deste método determine onde está:
        i. a frase correta cifrada
        ii. a chave usada para cifrar
        iii. o algoritmo de decifra e comparação;
    c. Realize um programa em Java que decifra a frase correta e teste a frase na aplicação Android.

    Entrega**:**

    - Breve descrição dos passos realizados;
    - Código fonte do programa que decifra a frase.

Notas sobre a biblioteca criptográfica da plataforma Java (JCA):

- O ficheiro `SymCipher.java` tem um exemplo de uma aplicação que usa a JCA para cifrar e decifrar uma mensagem com AES em modo CBC e *padding* PKCS#5;
- Na JCA a classe `Cipher` cifra e decifra usando o algoritmo (simétrico ou assimétrico), o modo de operação e o tipo de *padding*, indicados no método `getInstance`. Se a combinação não for suportada é lançada exceção em tempo de execução;
- As instâncias da classe `SecretKeySpec` são representações de chaves simétricas para determinado algoritmo simétrico (e.g. DES, AES), sendo possível afetar e ler diretamente o valor da chave. Estas representações são designadas de transparentes, em oposição à representação opaca. (https://docs.oracle.com/en/java/javase/11/docs/api/java.base/javax/crypto/spec/SecretKeySpec.html).