Project 1

Let's consider an Android application called "Secure Browser " which opens a *WebView* on a specific website. In order to avoid confused deputy attacks and to preserve privacy, Secure Browser accepts encrypted request only from authorized apps called "Supplicants", on an *ACTION_SEND intent filter*.

The Supplicant establish a session key with the Secure Browser through a protocol which satisfies the following requirements:

- At the end of the protocol execution a session key is established between the Supplicant and the Secure Browser.
- At the end of the protocol the Supplicant believes that the Secure Browser has the session key.
- At the end of the protocol the Secure Browser believes that the Supplicant has the session key.

Design and analyze a key exchange protocol matching these requirements.

Develop the Secure Browser and Supplicant apps with the following specifics:

- Secure Browser and Supplicant communicate through intents or a shared file in the SD Card, different for each supplicant.
- Secure Browser accepts three types of messages:
    - Start: ID -> Begins the key exchange protocol with the supplicant ID.
    - Open: ID -> Reads a string in the file related to a specific supplicant, decrypts the string and opens a *WebWiew* at the resulting web page.
    - Close: ID -> Close the connection and deletes the session key for user ID.
- If the ID is unknown or the message is in an unknown format, no actions are performed.
- Supplicant uses intents with the ACTION_SEND parameter to send messages to Secure Browser.

A short report on the project should be presented, discussing the hypothesis in which the protocol works correctly and the implementation choices.

Hints:

- Concurrent access to files and modification checking is easily handled through the FileObserver object.
- Accessing to the SD Card requires permissions.
- On the emulator it is possible to create virtual SD Cards.

Project 2

On Android device it is possible to intercept SMS text message before they leave the phone or as soon as they are received. To avoid this privacy issue it is possible to implement an SMS encryption app to exchange SMS messages with other users sharing the same apps. The users have both a public and private key. When a user wants to start a private communication with a peer, a protocol to establish a session key is started. Design and analyze a protocol which satisfies the following requirements:

- At the end of the protocol, both peers share a session key for symmetric encryption of SMS messages.
- At the end of the protocol, both peers are sure that the other peer received the session key.
- Expiring PINs should be used to ensure freshness.

Develop a sample application that performs the protocol and shares encrypted messages. The key exchange protocol is performed through SMS text messages, while PINs are manually inserted by users.

A short report on the project should be presented, discussing the hypothesis in which the protocol works correctly and implementation choices.

Hints:

- Use the emulators for app testing. It is possible to run two different instances of the emulator that can communicate through messages and phone calls.
- The permission required to read incoming messages is the READ_SMS permission while to send text messages the SEND_SMS permission is required.

Project 3

Let's consider a Peer-to-Peer system of Android devices where each peer owns a public and a private key. Before starting the communication, two peers (A and B) establish a session key to encrypt their communications. Design and analyze a key exchange protocol which satisfies the following requirements:

- At the end of the protocol a session key is established.
- At the end of the protocol A knows that B has received the session key.
- At the end of the protocol B knows that A has received the session key.

Develop a simple application which implements the designed protocol. After the session key is established A and B exchange a couple of messages encrypted and decrypted through the session key. The text of the messages, both sent and received, should be visible in an Activity. The application should run either on real devices or on the Android Emulator, included in the Android SDK.

A short report on the project should be presented, discussing the hypothesis in which the protocol works correctly and implementation choices.

Hints:

- Each emulator instance has a virtual network card and a private IP address.

Project 4

Two Android users, Alice and Bob, have a private app for secure chat (via Internet) between them made of two activities (see Figure). Activity 1 contains a textEdit component to write a message which is sent at the pressure of a button send. A text view shows if at the current status, the communication is "encrypted" or "not-secure".

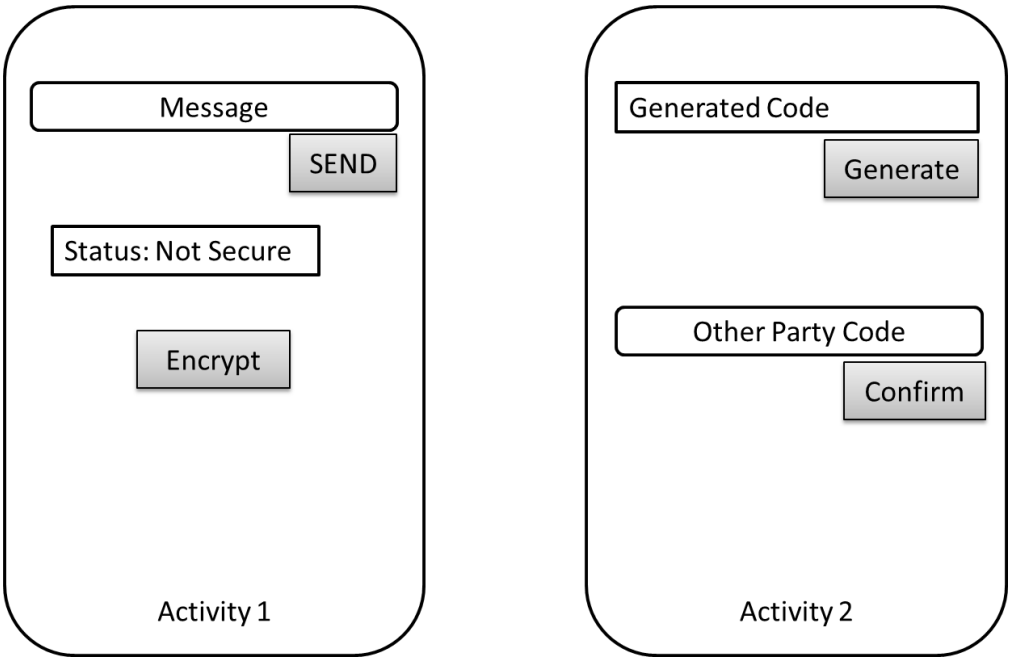
Pressing the "Encrypt" button, the Activity 2 comes in foreground. This activity has a text view to show a One Time Password code, generated by pressing the "Generate" button. The code has to be sent by Alice via SMS to Bob, who will insert the code in a textEdit component of Activity 2 and press the "confirm" button to go back to Activity 1. The code is used on both Alice and Bob device as Initialization Vector for a symmetric cipher of developer choose. The symmetric key will thus be used to encrypt all the messages sent by Activity 1 in both directions.

Implement the described application and write a short report explaining the application structure and demonstrating the validity of the envisioned protocol for data exchange, via BAN-Logic.

Optional: SMS message from Alice can be sent using the normal messaging app of the emulator, or directly from Activity 2 (more clean). In the last case, switch to Activity 1 immediately after sending the message.

Hints:

- Use the emulators for app testing. It is possible to run two different instances of the emulator that can communicate through SMS messages and phone calls.
- SMS has to be considered as a secure channel, i.e. if Bob receives an SMS from Alice he is sure that has been Alice the one sending that message.
- To send text messages the SEND_SMS permission is required.
- Each emulator instance has a virtual network card and a private IP address.



Java and Cryptography:
Some references to libraries for implementing cryptography in Java using standard cyphers.
https://docs.oracle.com/javase/7/docs/api/java/security/package-summary.html
https://docs.oracle.com/javase/7/docs/api/javax/crypto/package-summary.html
https://www.bouncycastle.org/java.html