

# Configuration Manual

MSc Research Project

Cyber Security

Noorullah Mohammed Sakhib

Student ID: X21225273

School of Computing

National College of Ireland

Supervisor: Niall Heffernan

**MSc Project Submission Sheet**

**School of Computing**

**Student Name:** Noorullah Mohammed Sakhib

**Student ID:** X21225273

**Programme:** MSc in Cybersecurity

**Year:** 2022-2023

**Module:** MSc Internship

**Lecturer:** Niall Heffernan

**Submission Due Date:** 14/07/2023

**Project Title:** PROTECTING USER DATA STORED LOCALLY ON SMARTWATCHES USING ASCON ENCRYPTION: A LIGHTWEIGHT CRYPTOGRAPHY APPROACH

**Word Count:** 906

**Page Count: 7**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Noorullah Mohammed Sakhib

**Date:** 14/07/2023

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

Mohammed Sakhil Noorullah

Student ID: x21225273

## 1. Introduction

The configuration document offers comprehensive details regarding the hardware, software, and implements used for the research project's implementation, testing, and training. Additionally, it gives specifics on how to execute the program's code and defines the implementation process and how results are obtained. The project consists of technique to encrypt and decrypt the data stored on a smartwatch emulator and analyse the obtained results.

## 2. Environment Setup

### 2.1 Configuration Setup

To reproduce this study, the pre-requisites indicated below has been used, which entail various installations as listed below, are necessary.

The experiment can be performed on an isolated virtual environment of a stand-alone machine. The use of a virtual lab environment is not compulsory; hence we have performed this experiment on the local machine and the details can be seen from Table 1.

1. Table: Host System Configuration

<b>Host System configuration</b>	
<b>Operating System type</b>	<b>Windows 10 (64-bit)</b>
<b>Memory</b>	<b>16 GB RAM</b>
<b>Processor</b>	<b>Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz</b>
<b>GPU</b>	<b>NVIDIA GeForce 840M (2 GB DDR3 dedicated)</b>

## 3. Installation of Software and tools

For this research we require an integrated development environment (IDE) which we form a base for all our operation that needs to be performed. Android studio is an official IDE for Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. For this research we will utilize this application to perform our analysis on smartwatch encryption. The programming languages used in this application are mainly Kotlin and Java.

The Android studio version and the programming languages used in this application are mentioned in Fig 1 and Table 2.

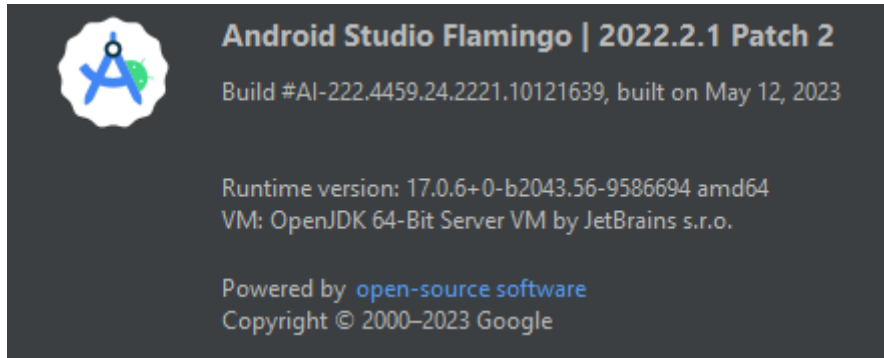


Fig 1: Android studio version

## 2. Table: Lab Configuration

Lab Configuration	
<b>Android Studio version</b>	<b>Android Studio Flamingo   2022.2.1 Patch 2</b> <a href="https://androidstudio.googleblog.com/2023/05/android-studio-flamingo-patch-2-now.html">https://androidstudio.googleblog.com/2023/05/android-studio-flamingo-patch-2-now.html</a>
<b>Programming languages</b>	<b>Kotlin and Java</b>

### 3.1 Extraction and Environment setup

The steps to extract code from a GitHub repository and run it in Android Studio as follows.

- a. *Clone the repository:*
  - Open a web browser and go to the GitHub repository [https://github.com/Noor17892/Ascon\\_Encryption\\_Project](https://github.com/Noor17892/Ascon_Encryption_Project)
  - Click on the "Code" button and select "Clone" to copy the repository URL.
- b. *Launch Android Studio:*
  - Launch Android Studio on your machine.
- c. *Open Project from Version Control:*
  - In Android Studio, select "Check out project from Version Control" on the welcome screen or from the "File" menu.
  - Choose "Git" from the dropdown list.
- d. *Paste Repository URL:*
  - Paste the repository URL you copied from GitHub into the "URL" field.
  - Choose a directory where you want to save the project on your local machine.
  - Click "Clone" to download the repository.
- e. *Build and Sync Project:*
  - Android Studio will open the project and begin syncing Gradle.
  - Wait for the Gradle build to complete. This might take a moment as Android Studio fetches dependencies.

f. *Run the App:*

- Once the project is synced, open the main app module (named "app", "common", "wearable").
- Click on the green "Run" button (a play icon) to build and run the app on an emulator or connected device.

**Note:** If there is any error while importing the code from GitHub repository then extract the .rar files from the zip folder (Ascon\_Encryption\_Project code) and paste all the files in the Android Studio folder.

#### 4. Execution

Once the setup is complete and Gradle build is done, we can start with the replication of our experiment. The Android studio comprises of 3 main modules.

- a. **app:** The app module has been designed to perform operation of an android emulator environment which is illustrated in Fig 2. The MainActivity.kt comprises the code which shows the output on Android emulator as illustrated in Fig 3, that can be selected from device manager.

The file to be analysed is first added into the common module and then file name is added into the MainActivity.kt of this module and the run is initiated. For this experiment we have used a test file "projecttest" which is a .txt format and which contains data in text format, that would help in understanding on how the application is performing encryption and decryption.

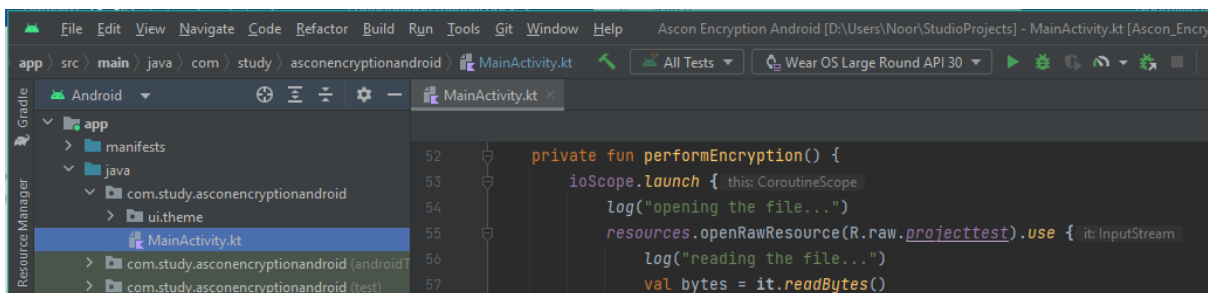


Fig 2: App module for smartphone emulator



Fig 3: Android emulator

- b. **Common:** The common module has been designed stored the Ascon encryption code and the data set is stored which will be processed for performing encryption and decryption operation and the performance is calculated as illustrated in Fig 4.

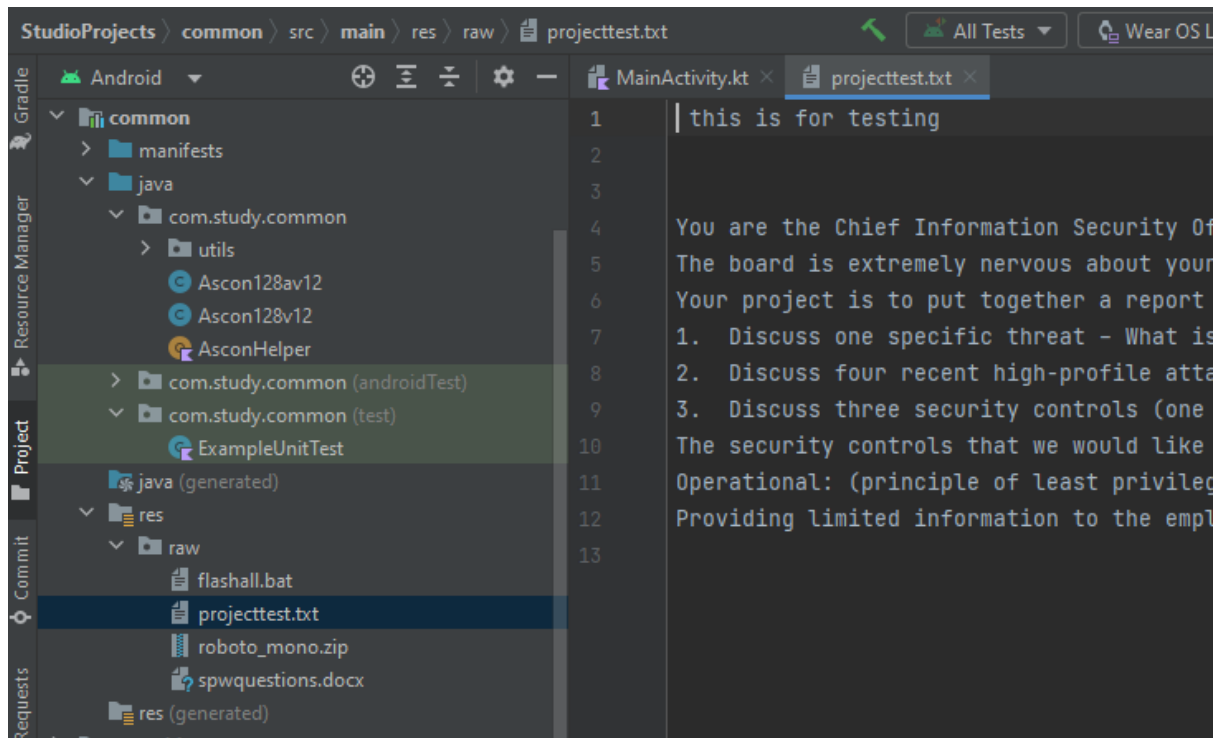


Fig 4: Common Module

- c. **Wearable:** The wearable module has been designed to perform operation of a smartwatch emulator environment which is illustrated in Fig 5. The MainActivity.kt comprises the code which shows the output on smartwatch emulator as illustrated in Fig 6 that can be selected from device manager. The file to be analysed is first added into the common module and then file name is added into the MainActivity.kt of this module and the run is initiated. For this experiment we have used a test file “projecttest” which is a .txt format and which contains data in text format, that would help in understanding on how the application is performing encryption and decryption.

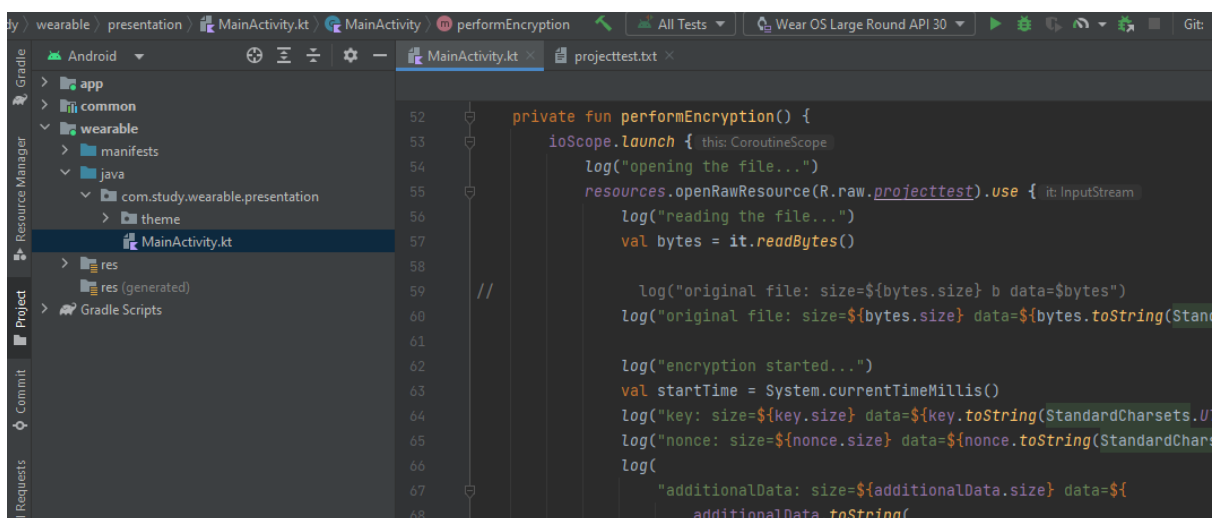


Fig 5: wearable Module



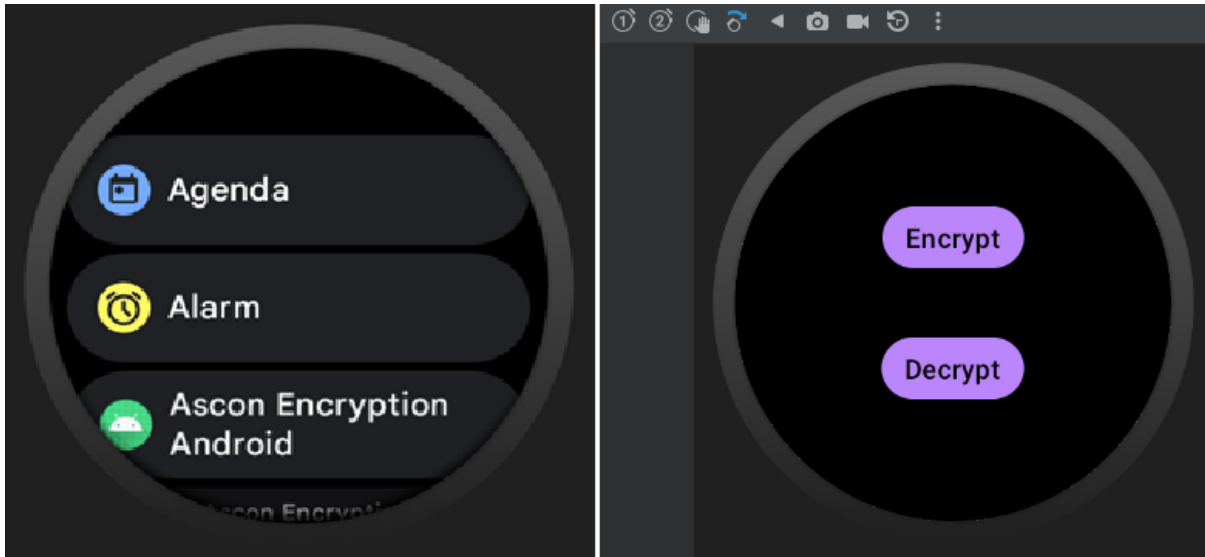


Fig 6: Smartwatch emulator

Once the module is selected and run successfully, the emulator can be initiated from the device manager as shown in Fig 7 and Fig 8. The button for encryption selected and the output can be observed in logcat and the performance can be seen in profiler tab respectively, which is present at the bottom of the android simulator application screen.

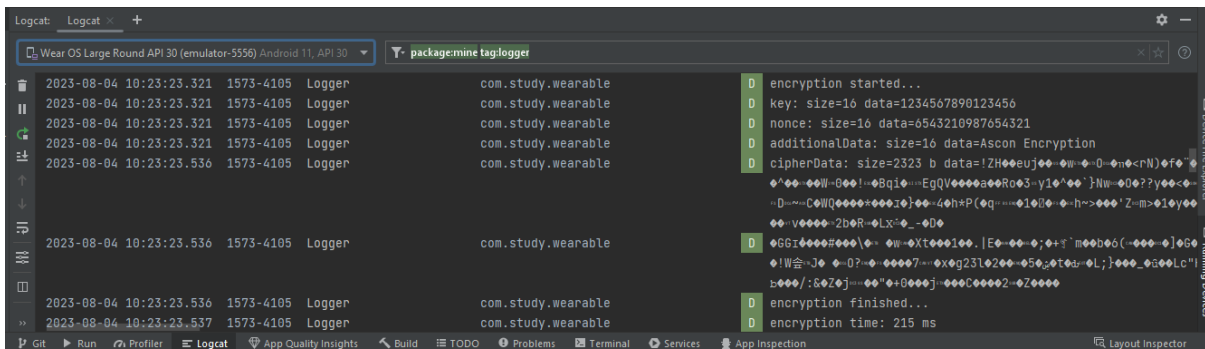


Fig 7: logcat output

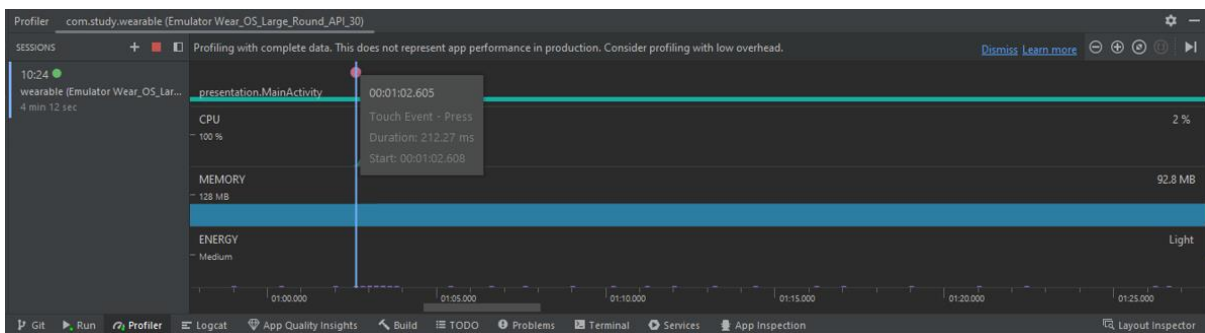


Fig 8: Profiler output

## 5. References

- <https://androidstudio.googleblog.com/2023/05/android-studio-flamingo-patch-2-now.html>
- [https://github.com/Noor17892/Ascon\\_Encryption\\_Project](https://github.com/Noor17892/Ascon_Encryption_Project)