National College of Ireland

Configuration Manual

MSc Research Project
Cybersecurity

Darshan Chanchad
Student ID: X22244174

School of Computing
National College of Ireland

Supervisor: Khadija Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Darshan Mukesh Chanchad.

Student ID: X22244174

Programme: MSc. Cybersecurity **Year:** 2024

Module: Practicum Part 2

Supervisor: Khadija Hafeez

Submission Due Date: 12-08-2024

Project Title: "INVESTIGATE THE INCORPORATION OF BIOMETRIC DATA (FINGERPRINTS, FACIAL RECOGNITION, ETC.) WITH RFID TECHNOLOGY TO PROVIDE MULTI-FACTOR AUTHENTICATION, INCREASING THE ROBUSTNESS OF SECURITY PROTOCOLS"

Word Count: ..... **Page Count**.....

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: Darshan Chanchad

Date: 12-08-2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Introduction

To integrate biometric data (facial recognition, fingerprint scanning, iris scanning) with RFID tags for authentication, there a combination of hardware, software, and network infrastructure used in these project implementations using the principle of MFA.

Experiential Setup

Device System Configuration
Hardware Components Specifications
Dell Latitude 7300 Laptop
Processor (CPU) 2.69 GHz 6-Core Intel Core i7
Memory 16 GB RAM DDR4
Storage 1TB Solid State Drive
Operating System Windows 11 Home Single
Language - version (23H2)

▽ **Workflow of the system goes as:**

1. RFID Scan: User presents their RFID tag to the reader.
2. Biometric Capture: Depending on the setup, the system prompts the user for fingerprint, facial, or iris verification.
3. Data Matching: The captured biometric data is matched against stored templates associated with the RFID tag.
4. Access Decision: Based on the match, the system grants or denies access.
5. Logging: The result is logged locally or sent to a remote server for monitoring and auditing.

❖ **Here's a comprehensive list of the components used:**

1. Hardware Components
A. RFID System
- RFID Reader: Device to read RFID tags. Examples include RC522, PN532.
- RFID Tags: Cards or key fobs that users carry for identification.

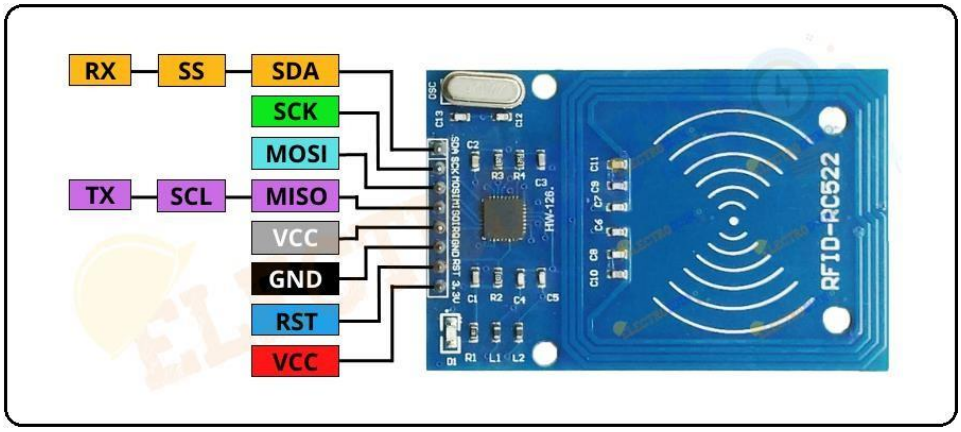

Figure 1: RC522

B. Biometric Sensors

- Fingerprint Sensor: Device to capture and verify fingerprints. Examples include R305, R307, FPC1020.
- Facial Recognition Camera: A camera module that captures images for facial recognition. Examples include the Raspberry Pi Camera Module, USB Webcam, or specialized cameras like Intel RealSense.
- Iris Scanner: Sensor for capturing and analyzing iris patterns. Examples include the IriShield series, Panasonic BM-ET200.

Figure 2: R305

❖ **MicroPython Technical Details**

I believe to "MicroPython," is a lean and efficient implementation of the Python 3 programming language that is designed to run on microcontrollers and in constrained environments.

1. Overview:

- MicroPython is designed for microcontrollers, enabling them to execute Python code directly on the hardware.
- Size The MicroPython interpreter is extremely compact, typically requiring only around 256KB of flash and 16KB of RAM.

2. Supported Platforms:

-Microcontrollers:

- ESP32
- ESP8266
- STM32
- RP2040 (Raspberry Pi Pico)

- SAMD21 (Arduino MKR series)
- NRF52
- Pyboard (a MicroPython development board)

-Other platforms: Unix/Linux, Windows, and bare metal ports

❖ **Thonny Technical Details**

Overview:

- **Purpose:** Thonny is an easy-to-use IDE aimed at Python beginners. It simplifies Python programming by providing a user-friendly interface and integrated tools that help users understand code execution and debugging.
- **Platforms:** Available for Windows, macOS, and Linux.

B. Processing Unit

- Raspberry Pi: A microcomputer capable of processing RFID and biometric data.

- Alternative Microcontroller: NVIDIA Jetson Nano or Arduino (for basic tasks) are used depending on processing requirements.

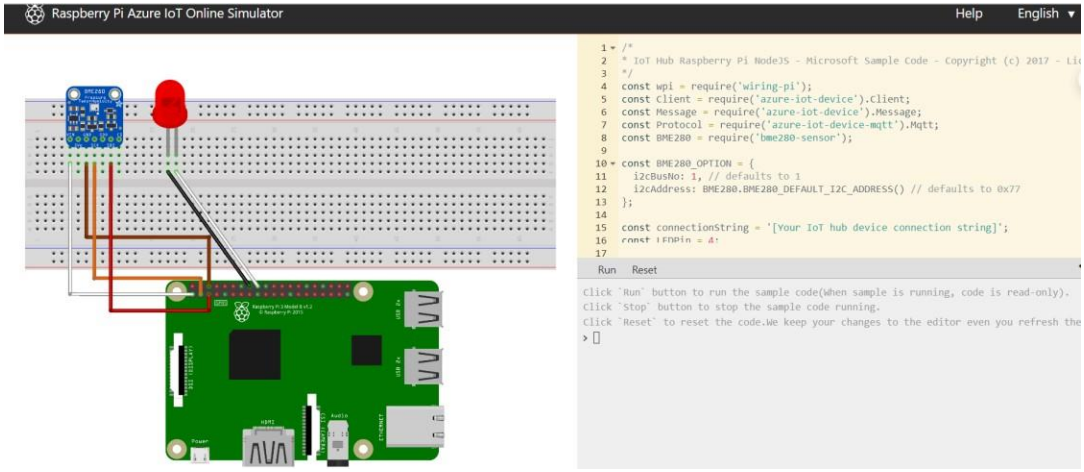

Figure 3: Raspberry Pi

C. Peripheral Components

- Power Supply: To power the Raspberry Pi and other connected hardware.
- GPIO Breakout Board: For connecting multiple sensors and the RFID reader to the Raspberry Pi.
- Display: An LCD or OLED screen for user feedback.
- Housing/Enclosure: To securely mount all components.

3. Software Components

A. Operating System

- Raspberry Pi OS: The default OS for Raspberry Pi, which will be used to run the authentication software.

B. Biometric Software/SDK

- Fingerprint Recognition:
- `Adafruit_Fingerprint` library for Python, compatible with Adafruit fingerprint sensors.

- `libfprint` for handling more advanced fingerprint recognition tasks.
- Facial Recognition:
- `OpenCV` library for image capture and processing.
- `dlib` library for facial recognition using machine learning models.
- Pre-trained models like `FaceNet`, `DeepFace`, or `MTCNN` for accurate facial recognition.
- Iris Recognition:
- Manufacturer-provided SDKs specific to the iris scanner hardware.
- `Libiris` or other open-source libraries that support iris recognition.

C. RFID Libraries

- MFRC522 or SimpleMFRC522 for Python to interface with the RFID reader.

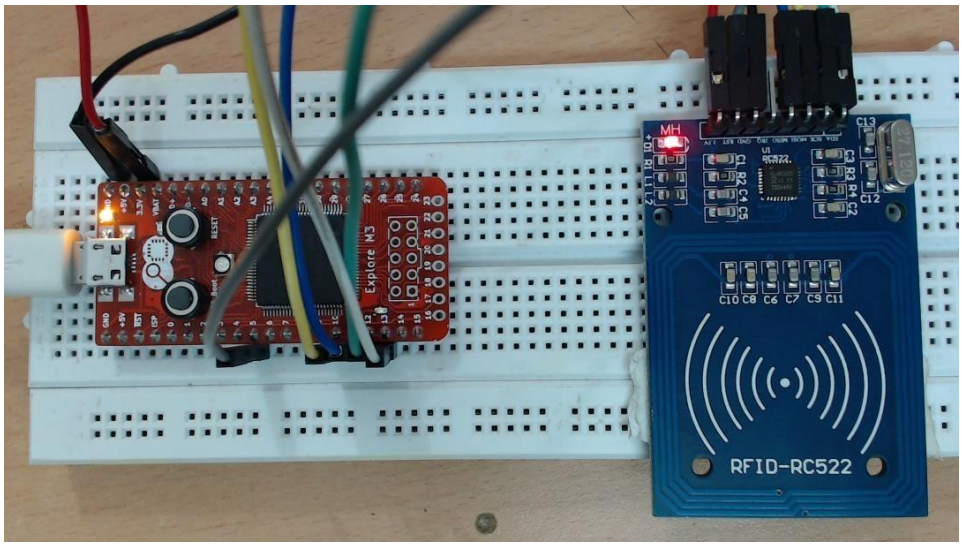

Figure 4: Programming MFRC522 with RFID reader using python.

D. Database/Storage System

- Local Database: SQLite or MySQL for storing RFID tags, biometric templates, and user data.
- Cloud Database: Firebase, AWS RDS, or any other cloud-based database for remote storage and access.

E. Integration Software

- Flask or Django: Python web frameworks for creating an API to manage and authenticate users via RFID and biometric data.
- MQTT: For IoT-based communication if integrating with other devices or services.
- Encryption Libraries: For securely storing and transmitting biometric and RFID data, such as PyCryptodome.

4. Integration Infrastructure

A. Networking

- Wi-Fi/Ethernet: For connecting the Raspberry Pi to the network or the internet.

- Router/Access Point: To provide network connectivity.

B. Cloud Integration

- IoT Platform: AWS IoT, Azure IoT Hub, or Google Cloud IoT for device management and data logging.
- Webhooks/APIs: To send and receive data from cloud services or other web-based applications.

5. Security and Compliance

A. Data Encryption

- TLS/SSL: To secure data transmission between devices and any remote services.
- Data Encryption: Encrypt biometric templates and RFID data at rest and in transit using strong encryption algorithms.

B. Compliance

- Regulatory Compliance: Ensure that the system adheres to data protection regulations like GDPR, CCPA, etc.
- User Consent: Mechanisms to obtain and record user consent for collecting and storing biometric data.

6. Development and Testing Tools

- Integrated Development Environment (IDE): Tools like Visual Studio Code or PyCharm for coding.
- Testing Tools: Software tools to simulate and test the biometric and RFID integration, such as OpenCV test frameworks or RFID emulators.

7. Additional Considerations

- Scalability: Plan for how the system will scale if deployed across multiple sites.
- User Experience: Ensure that the system provides quick feedback and is user-friendly, with minimal delays in authentication

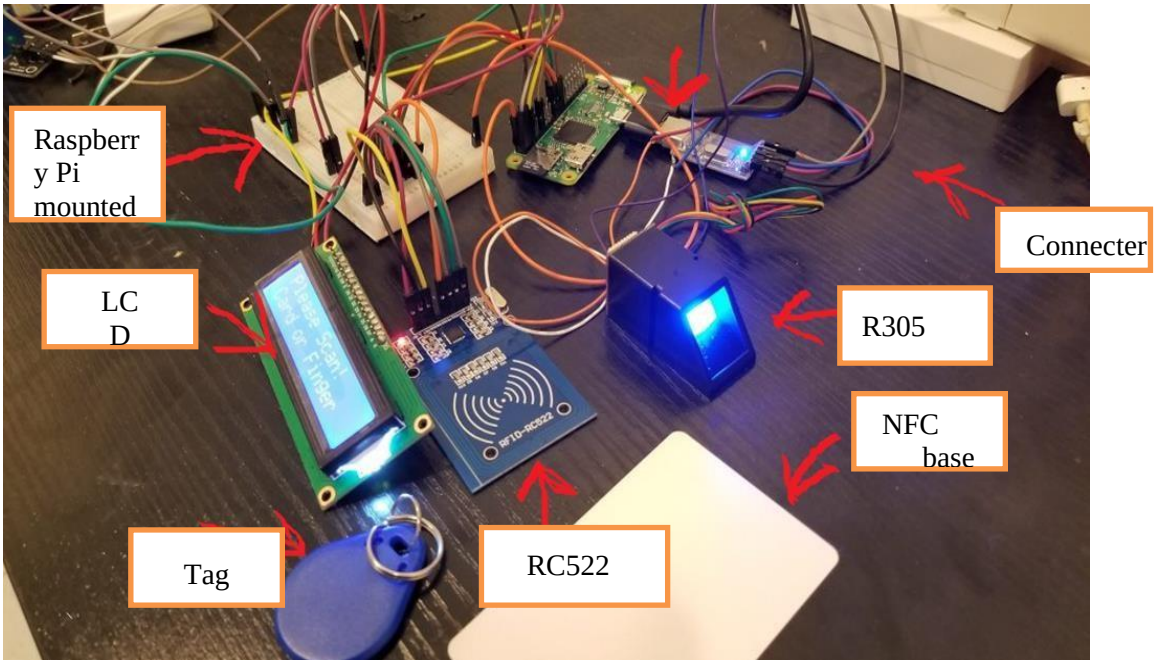

Figure 5: Overall Setup of the project implementation