CMU Project Report (Team6)

Table of Contents

[0. Introduction 3](#_Toc74220137)

[1. Schedule 3](#_Toc74220138)

[2. System Requirement 3](#_Toc74220139)

[3. Security Goals 3](#_Toc74220140)

[4. Security Requirements 3](#_Toc74220141)

[5. Assets 4](#_Toc74220142)

[6. Threat Modeling 4](#_Toc74220143)

[7. Security Risk Assessment 4](#_Toc74220144)

[8. Mitigation 4](#_Toc74220145)

[9. Architecture 4](#_Toc74220146)

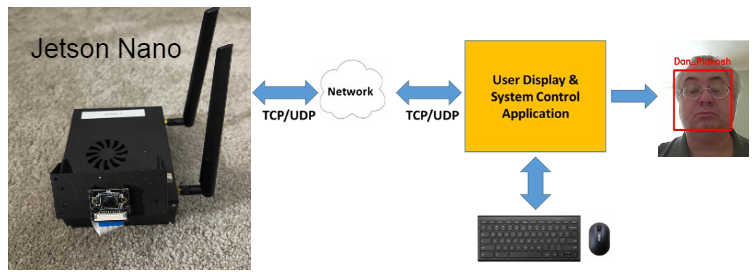
[10. Static Analysis 4](#_Toc74220147)

[11. Fuzz & Penetration Test 5](#_Toc74220148)

[12. Conclusion 5](#_Toc74220149)

# 0. Introduction

TODO: Introduce team member and role and project



# 1. Schedule

TODO: attach schedule sheet

# 2. System Requirement

Mandatory requirements described in the "LG May 2021 Lecture Secure Coding Project Intro V1.1.pptx.pdf" document.

- no vulnerability in the system

- secure architecture

- implement 5 modes (run, test run, learning, secure, non-secure)

- Jetson Nano sends the Camera Image and Face Recognized information. It should be separated.

- Client receives the data above, and displays it after combining it

# 3. Security Goals

Protecting the user privacy information in our system.

# 4. Security Requirements

-- iter 1

[ ] Any information related (personal) privacy SHALL be protected securely. (Friend video/Learned PHOTO)

[ ] Any information related (personal) privacy SHALL be accessible to only authorized entities. (Learned PHOTO)

[\*] The system SHALL use only approved algorithms for cryptographic operation.

[\*] Server and client SHALL communicate over encrypted and authenticated channel.

-- iter 2

[\*] Any information related (personal) privacy SHALL be protected securely. (ID/PASS/Friend video/Learned PHOTO)

[\*] Any information related (personal) privacy SHALL be accessible to only authorized entities. (ID/PASS/Learned PHOTO)

[\*] The system SHALL have a resiliency against key compromise. (TLS Key, Cert)

# 5. Assets

-- iter 1

[ ] Images for transmission over camera cable

[\*] Images for transmission over network

[\*] The Friend video

[\*] Client program itself

[\*] Client program hash code on server side

-- iter 2

[\*] User info. data (ID, type, password)

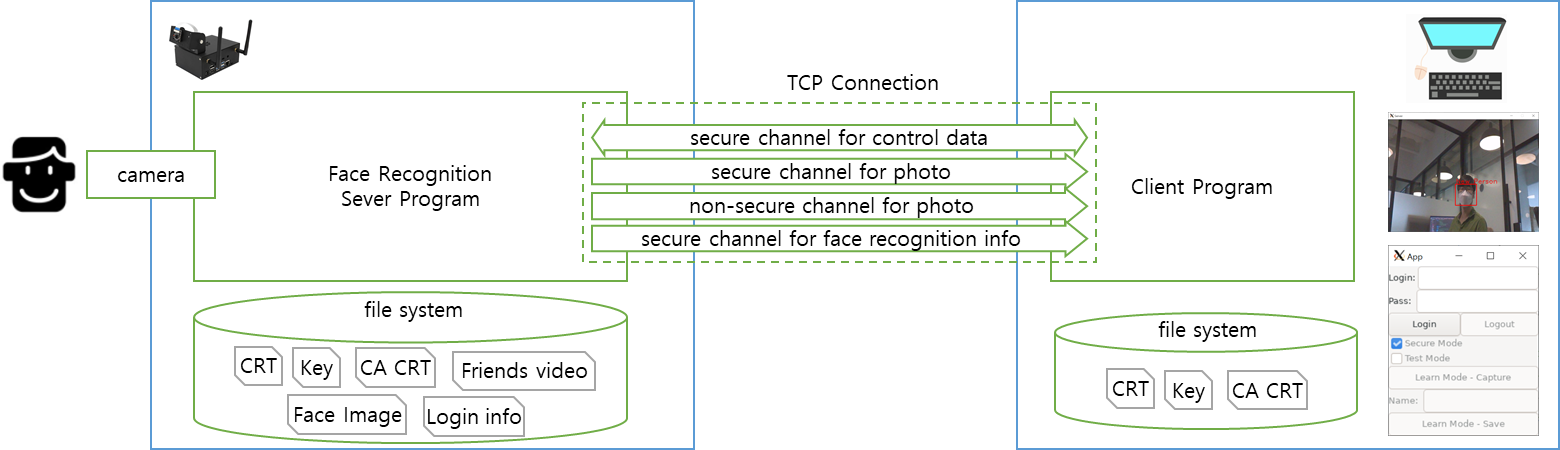
# 6. Threat Modeling

# 7. Security Risk Assessment

# 8. Mitigation

# 9. Architecture

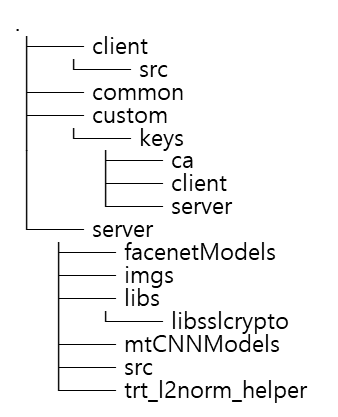
## 9.1. Overall Architecture



## 9.2. Terminology and Definitions

|  |  |
| --- | --- |
| **Terminology** | **Definitions** |
| CA CRT | Self signed Root Certificate |
| CRT | CA signed Certificate |
| Key | Private Key |
| Login info |  |
| Face Image |  |
| Photo |  |
| Secure Mode |  |
| Non Secure Mode |  |
| Run Mode |  |
| Test Run Mode |  |
| Learn Mode |  |
| Secure channel for control data | TLS TCP connection.  The request and response message is transmitted. |
| Secure channel for photo | TLS TCP connection.  The photo data is transmitted from the server to the client |
| Non-secure channel for photo | TCP connection.  The photo data is transmitted from the server to the client |
| Secure channel for face recognition info | TLS TCP connection.  The coordination of the recognized face on the photo and the recognized name is transmitted from the server to the client |

## 9.3. Source Directory



## 9.4. Setup Guide

### 9.4.1. Server

cd source/server/

python3 step01\_pb\_to\_uff.py

rm -rf MTCNN\_FaceDetection\_TensorRT/

git clone https://github.com/PKUZHOU/MTCNN\_FaceDetection\_TensorRT

mv MTCNN\_FaceDetection\_TensorRT/det\* ./mtCNNModels

mkdir build; cd build

cmake -DCMAKE\_BUILD\_TYPE=Release ..

make -j

./LgFaceRecDemoTCP\_Jetson\_NanoV2 5000 9.2.2. Client

### 9.4.2. Client

apt install cmake libssl-dev libgtkmm-3.0-dev libopencv-dev

cd source/client/ && mkdir build; cd build

cmake ..

make

./client

## 9.5. Scenario

# 10. Static Analysis

# 11. Fuzz & Penetration Test

# 12. Conclusion