
Developer Guide

Team 5

----

1. Security Response Contact

Group mail : ss-5verflow@googlegroups.com
Contact : SeungWook Cha (chawinner@gmail.com)

2. Transition Package

The following are our artifacts which are documented by Team 5.

Artifacts	Document name	
Presentation	(PPT) Phase 1 Presentation	Presented on Jun 18th
Source	(Site) Github	
Asset	(Doc) Project Asset List	
Requirement	(Doc) User Requirement (Doc) Project Requirement (Doc) Software Requirement (Excel) Security Requirement	Security RQ includes DFD, OWASP and Risk Assessment.
Design	(Doc) Software Architecture Design	
Test	(Excel) Test Cases	
Others	(Excel) Project Planning (Script) Script for installation of the certificate.	

3. Guide for build and run our system

1.1 How to get the source code

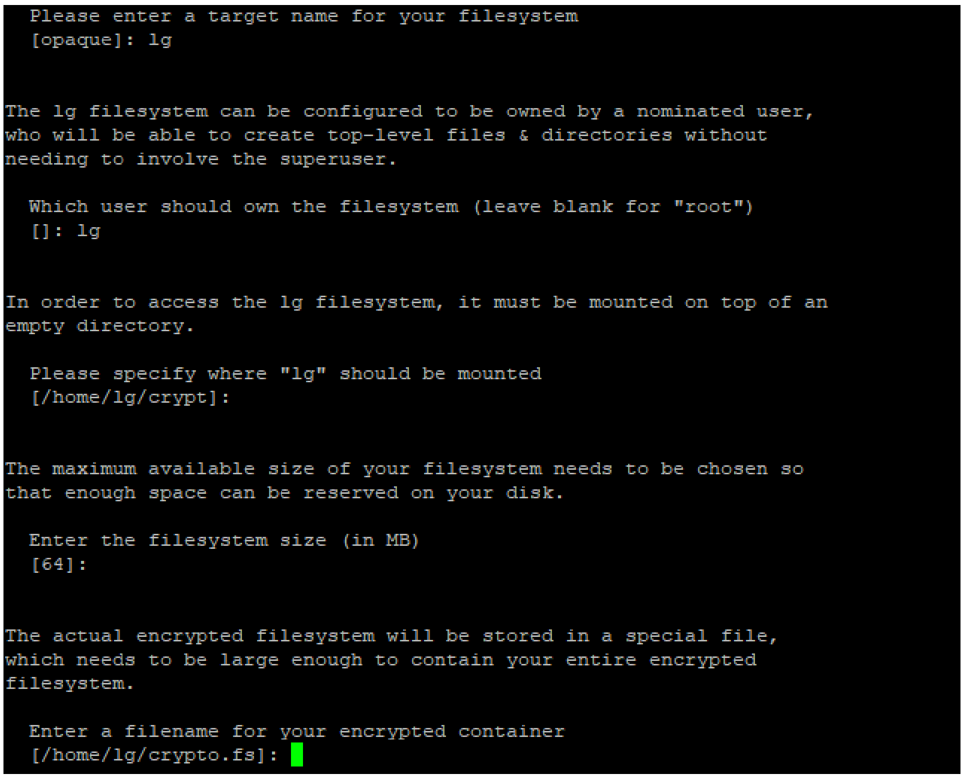
Github : https://github.com/5verFlow/sfid-document

1.2 How to build the system

1.2.1 Server (Linux, Jetson Nano)

Step 1. Install Secure Storage

This project uses cryptmount, an encrypted filesystem, to store registered user image files and credentials.

1. Install cryptmount
   $ sudo apt install cryptmount
2. Set up crypt mount for a target filesystem
   $ sudo cryptmount-setup
   

```
  Please enter a target name for your filesystem
  [opaque]: lg


The lg filesystem can be configured to be owned by a nominated user,
who will be able to create top-level files & directories without
needing to involve the superuser.

  Which user should own the filesystem (leave blank for "root")
  []: lg


In order to access the lg filesystem, it must be mounted on top of an
empty directory.

  Please specify where "lg" should be mounted
  [/home/lg/crypt]:


The maximum available size of your filesystem needs to be chosen so
that enough space can be reserved on your disk.

  Enter the filesystem size (in MB)
  [64]:


The actual encrypted filesystem will be stored in a special file,
which needs to be large enough to contain your entire encrypted
filesystem.

  Enter a filename for your encrypted container
  [/home/lg/crypto.fs]:
```

3. Mount(Access) the new encrypted target filesystem
   $ cryptmount ‘target’
4. Make directories below crypt directories for next Step2. Install Certificate
   /home/<user>/crypto/imgs/
   /home/<user>/crypto/ca/intermediate/certs/
   /home/<user>/crypto/ca/certs/
   /home/<user>/crypto/ca/intermediate/private/
5. Unmount the target filesystem when finished using.
   $ cryptmount --unmount ‘target’

Step 2. Install Certificate

The certificate installation given by shell script and configuration files

1. make a folder named ‘ca’ to secure storage
   ```
   $ cd /home/lg/crypt
   $ mkdir ca
   ```
2. download and copy the materials (clean_all.sh, int_openssl.cnf, openssl.cnf, make_cert.sh) to the “ca” folder secure storage (home/lg/crypt/ca)
3. run shell script. the script will automatically creates CA certificate, intermediate certificate, and leaf certificate
   ```
   $ ./make_cert.sh
   ```
4. the system will ask you several times for passphrases for CA, intermediate certificate, leaf certificate.

a brief descriptions of the generated certificates are as follows

- CA certificate: self-signed root certificate for the system’s security chain.
- intermediate certificate:certificate for the server (Jetson Nano), signed by CA. when the server application starts, it will ask the passphrase for the intermediate certificate you typed in.
- leaf certificate: certificate for the client (Windows Client), signed by intermediate certificate.
- chain certificate: chain of CA and intermediate certificate.

Step 3. Setup Log (rsyslog)

This project uses syslog as a logging system, currently the log is stored at /var/log/sfid.log but you can move it to the cryptmount directory if you want to encrypt log also.

1. insert below line into /etc/rsyslog.d/50-default.conf
   ```
   local0.*                              /var/log/sfid.log
   ```
2. restart rsyslog
   ```
   $ sudo systemctl restart rsyslog
   ```

Step 4. Build Command

1. `$ unzip LgSecureCoding2021Distv5.zip`
2. `$ cd LgSecureCoding2021Distv5/LgFaceRecDemoTCP_Jetson_NanoV2`
3. `$ mkdir build && cd build`
4. `$ cmake ..`
5. `$ make -j32`
6. `$ ls -l`
   a. `-rwxrwxr-x 1 jinn jinn   719936 Jun  1 19:36 LgFaceRecDemoTCP_Jetson_NanoV2*`

Step 5. Run Command

1. $ cd LgSecureCoding2021Distv5/LgFaceRecDemoTCP_Jetson_NanoV2/build
2. $ ./LgFaceRecDemoTCP_Jetson_NanoV2
3. $ [[*Secure Passphrase*]] you typed in for intermediate certificate when asked

1.2.2 Client (Windows)

Step 1. Install **Visual Studio 2019 Community**

1. Download **Visual Studio 2019 Community** (https://visualstudio.microsoft.com/ko/vs/community/)
2. Install

Step 2. Install **Qt 5.15.2**

Since the client application is developed on top of Qt, we need to install Qt.

1. Download “Qt Online Installer” from the following website: (https://www.qt.io/download-open-source)
2. Execute the downloaded file.
3. Create “Qt account”. You should verify your email while creating your account.
4. Select "Individual" in the license verification step.
5. Select “Custom installation” at the stage of “Installation Folder”.
6. Select “Qt” → “**Qt 5.15.2**” → “MSVC 2019 32-bit” and “MSVC 2019 64-bit”
7. Install

Step 3. Install **Qt Visual Studio Tools(**Qt extension for Visual Studio 2019)

1. Run “Visual Studio 2019”
2. Select “Extensions” -> “Manage Extensions”
3. Search “**Qt Visual Studio Tools**”, and install.
    a. you can easily find with searching ‘qt’
4. Rerun “Visual Studio 2019”
5. Select “Extensions” -> “Qt VS Tools” -> “Qt Versions”
6. Select and add “qmake.exe” in the directory installed in step 1(add both 32-bit and 64-bit).
    a. usually in “C:\Qt\5.15.2\msvc2019_64\bin” for 64-bit and “C:\Qt\5.15.2\msvc2019\bin” for 32-bit

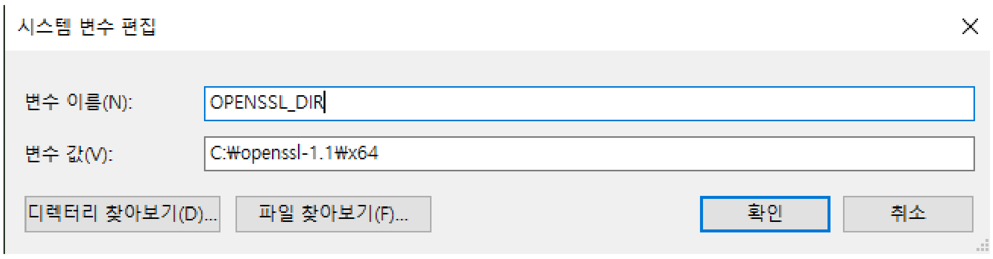
Step 4. Install OpenCV

It is assumed that it is already installed.

Step 5. Install OpenSSL

OpenSSL v1.1.1k is required, which is the latest binary distribution for Microsoft Windows on June 17, 2021 (KST). To install **OpenSSL v1.1.1k**, take the following steps.

1. Download OpenSSL 1.1.1k from the following website:
   Click *Download FireDaemon OpenSSL 1.1.1k Binary Distribution* button on the FireDaemon OpenSSL web site (Website Link).
2. Unzip the downloaded file (openssl-1.1.1k.zip)
3. Add a system variable "OPENSSL_DIR" to the folder containing OpenSSL as the follows

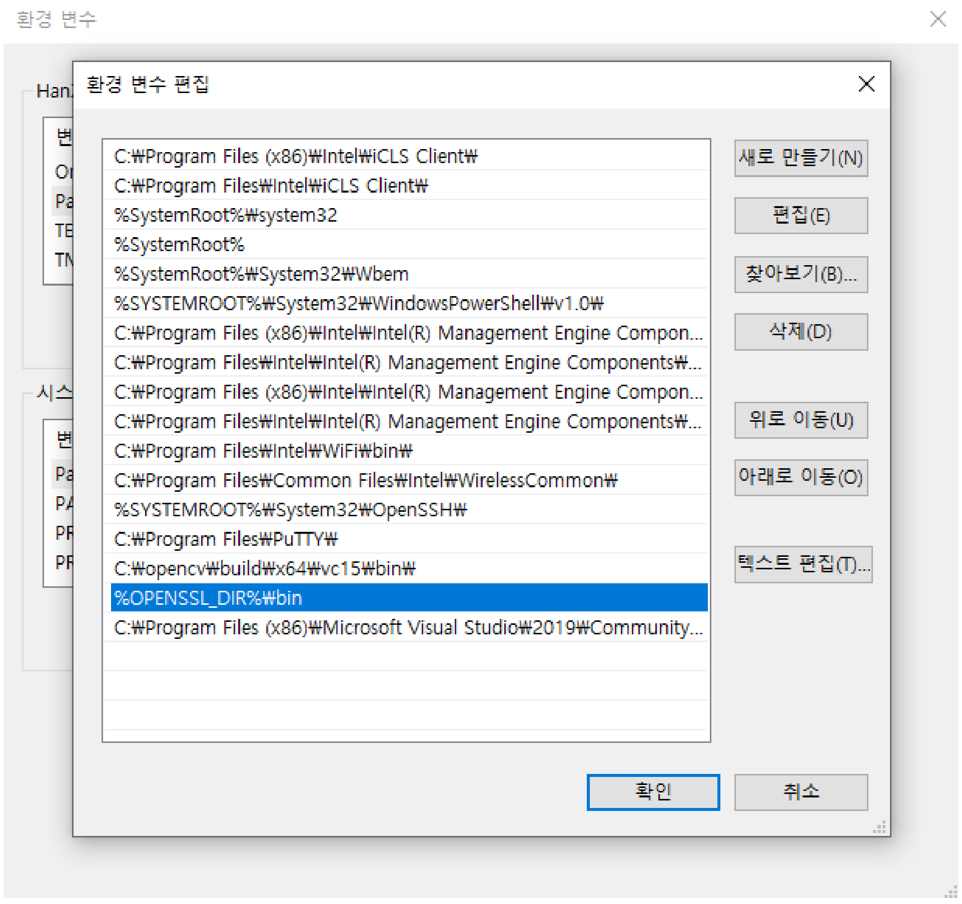


시스템 변수 편집

변수 이름(N): OPENSSL_DIR

변수 값(V): C:₩openssl-1.1₩x64

4. Add a system path to the folder containing binary of the OpenSSL (i.e., openssl-1.1\x64\bin), as shown in the example image.



환경 변수

환경 변수 편집

```
C:₩Program Files (x86)₩Intel₩iCLS Client₩
C:₩Program Files₩Intel₩iCLS Client₩
%SystemRoot%₩system32
%SystemRoot%
%SystemRoot%₩System32₩Wbem
%SYSTEMROOT%₩System32₩WindowsPowerShell₩v1.0₩
C:₩Program Files (x86)₩Intel₩Intel(R) Management Engine Compon...
C:₩Program Files₩Intel₩Intel(R) Management Engine Components₩...
C:₩Program Files (x86)₩Intel₩Intel(R) Management Engine Compon...
C:₩Program Files₩Intel₩Intel(R) Management Engine Components₩...
C:₩Program Files₩Intel₩WiFi₩bin₩
C:₩Program Files₩Common Files₩Intel₩WirelessCommon₩
%SYSTEMROOT%₩System32₩OpenSSH₩
C:₩Program Files₩PuTTY₩
C:₩opencv₩build₩x64₩vc15₩bin₩
%OPENSSL_DIR%₩bin
C:₩Program Files (x86)₩Microsoft Visual Studio₩2019₩Community...
```

Step 6. Copy Certificate

The certificate and key files required for connection are generated in Step 2 of Clause 1.2.1. The only steps remaining is to copy the files to the folder the client application is installed in.

1. Copy the chain and leaf certificates as follows:
   1) Connect to Jetson Nano using WinSCP
   2) Change directory to /home/lg/crypt/ca/intermediate/certs
   3) Copy ca-chain.cert.pem and leaf.cert.pem to Windows
2. copy the key for the leaf certificate as follows:
   1) Connect to Jetson Nano using WinSCP
   2) Change directory to /home/lg/crypt/ca/intermediate/private
   3) Copy leaf.key.pem to Windows

Step 7. Build

1. Download Client source code file and unzip it where you want (e.g. D:\sfid-client)
2. Run “Visual Studio 2019”
3. Select “File” → “Open(O)” → “Project/Solutions(P)”
4. Choose sfid-client.sln file at the unzipped folder
   a. e.g. D:\sfid-client\windows_vs2019/sfid-client.sln
   b. Then, sfid-client project would be opened. See project name on solution explorer
5. Click right button on the PROJECT name (sfid-client) and select “Property” (Alt+Enter)
   a. Alternatively, Select “Project”--> “Properties”
   b. Do NOT click on the solution name. It shows a different menu.
6. Choose Configurations to “All configurations”
   a. to apply both debug and release mode
7. Select “C/C++” → “General” on the left side, and then choose “Additional including Directories”. Add including paths for openCV and openSSL libraries.
   a. e.g. %OPENCV_DIR%\..\..\include; %OPENSSL_DIR%\include
8. Select “Linker”--> “General” on the left side, and then choose “Additional library Directories”. Check “$(Qt_LIBPATH_)” is already set. Add library paths for openCV and openSSL
   a. e.g. %OPENCV_DIR%\lib; %OPENSSL_DIR%\lib
9. Select “Linker”--> “Input” on the left side, and then choose “Additional Dependencies”. Add following libraries
   a. opencv_world451.lib;opencv_world451d.lib;libcrypto.lib;libssl.lib
10. Select the “OK” button on the bottom of the window. Project configuration is finished.
11. Select the configuration mode to “Debug” or “Release”, then BUILD. You can have options to build.
    a. Solution build : Build→ Solution build (Ctrl+Shift+B)
    b. Project build : Build → Project build (Ctrl+B)
12. Launch and debug to run the client application
    a. Debug w/ breakpoints : Debug → Start with Debugging (F5)
    b. Start directly on Debug mode : Debug → Start Without Debugging (Ctrl+F5)

1.3 How to start the system?

1.3.1 Server (Linux, Jetson Nano)

Step 1. Connect to Jetson Nano console.

- ssh 192.168.0.100 (lg / lg)

Step 2. Go to the directory.

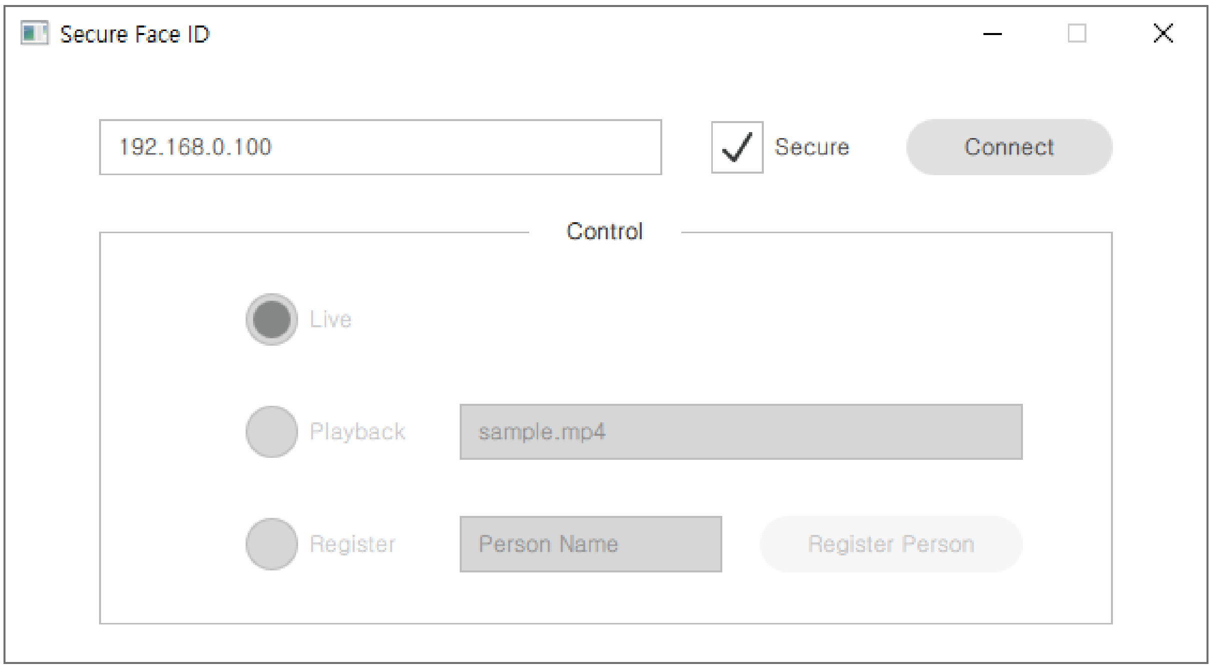
- cd LgSecureCoding2021Distv5

Step 3. Run the Jetson Nano application. You don't need to specify the port.

- ./LgFaceRecDemoTCP_Jetson_NanoV2

1.3.2 Client (Windows)

Step 1. Run the client application



Choose Communication Mode

1. secure mode
   a. Input a valid IP address to connect with server
   b. Check the secure box.
   c. Click the connect button.
2. Insecure mode
   a. Click a valid IP address to connect with server
   b. Uncheck the secure box.
   c. Click the connect button.

Choose Operation Mode

Basically Live mode using the camera is default.
You can change the operation mode during the connected environment

1. Live (* = Run mode)
   a. Select the Live button.
2. Playback (* = Test mode)
   a. Select the Playback button.
3. Register (* = Learning mode)
   a. Select the Register button.
   b. Write a name you want on the Person Name input
   c. Click the “Register Person” button.

Note 1 : Input validation policy at Client application

When a user wants to write something through the client application, some input validation policies are run at GUI side.
For IP address input:

- Only numbers within 0~255 and the separator(.) are allowed to be input.

For Register user input:

- Basically, numbers, English alphabet, and space are allowed to be input.
- Special characters are not allowed to input.
- Starting or ending with space is not allowed to send a register order.(button is not activated)
- over 256 characters cannot be written

Note 2 : Registration name policy

You can write any valid names as input even if there are two or more people who have the same name. The server will store images with index numbers to prevent overwriting. When a registered person is at the front of the camera again, only the registered name without index number will be seen on the video window.

Note 3 : Registration conditions

If a person wants to register the face, there should be only one unregistered(unknown) person. If there is no face detected or there are so many unknown faces, the client denies the registration order even though the name written is valid.