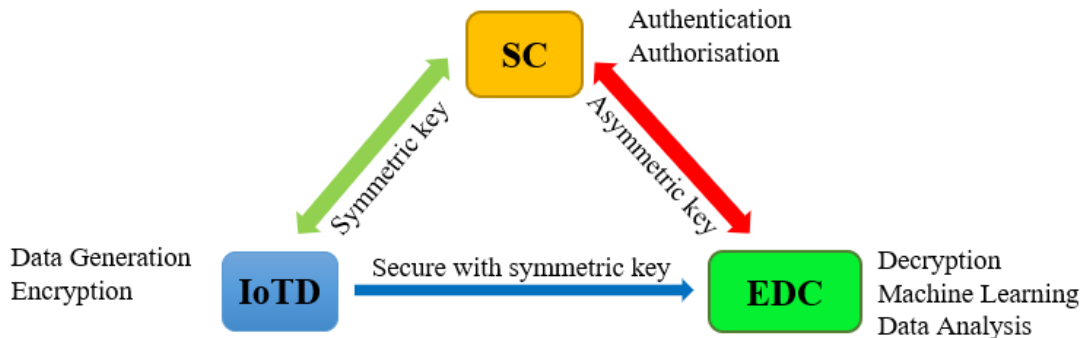


Introduction

This repository contains the files of the research project titled

ACCESS CONTROL IN IOT GATEWAY USING SOFTWARE-DEFINED PERIMETER CONTROLLER AND DECISION TREE.

The scripts used in the three components (SC, IoTD and EDC) are available in the folders with the respective names. The folder named 'Graph' contains the graphs and plots used in the document.



The folders named after the components contain the **Dockerfile** used to create the docker image. The docker images available here for the **armv32** architecture (Raspberry Pi) and may not run in ordinary systems.

Reproducibility

In order to reproduce the system, the required system and software requirements are

- Three SSH enable Raspberry Pi with Docker installed.
- A static IP

The three components must run parallely to reproduce the model. Docker images for the components are uploaded in the **www.dockerhub.com** and one must pull it to the system to execute it. For the details of execution see the video named **Execution_Details.mp4** in the root directory

1. SC

Pull the image from the docker using the command

```
docker pull stanlysac/sc:controller_pi_ip
```

After pulling the image run it as follow

```
docker run -it -v <Location_in_the_system>:/logs stanlysac/sc:controller_pi_ip  
<Mqtt ip>
```

Eg.

```
docker run -it -v /home/pi/Documents:/logs stanlysac/sc:controller_pi_ip
54.196.9.248
```

Location_in_the_system is given to save the error details for further analysis

2. EDC

It requires a couple of docker images

```
docker pull ivanmarban/armhf-mongodb
docker pull eclipse-mosquitto
docker pull ashokjjk/benchmark:decrypt
docker pull stanlysac/edc:publish
docker pull stanlysac/edc:ml_pi
```

Once all the images are in the EDC system, the folder (EDC) contains a file named **run.sh** which has the scripts to run all the images together. Copy the file to the system and enable it executable. Run the files as

```
./run.sh <Mqtt ip> <delay>
```

The delay here means the time gap between the requests, and it is in seconds. Eg:

```
./run.sh 54.196.9.248 30
```

3. IoT D

Pull the image from the docker using the command

```
docker pull ashokjjk/urbanapi:pi
```

Execute the image as

```
docker run -d ashokjjk/urbanapi:pi <Mqtt ip> <Mqtt ip>
```

Eg:

```
docker run -d ashokjjk/urbanapi:pi 54.196.9.248 54.196.9.248
```

