SINGAPORE POLYTECHNIC | SP

Internet of Things Security (ET0731)

Project Report

Class: DCPE/FT/3A/25

Topic: Secured End-to-End IoT for Door Entry System

Title: Secure IoT Lock

Student Number	**Full Name**
2032555	Lim Zheng Long
2032568	Lim Jun Hong
2032638	Wang Zheng

Contents

1: Overview of Secure IoT Lock

Secure IoT Lock is a lock that can be unlocked wirelessly via the device webpage on any device. It provides security features such as data encryption and 2 factor-authentication to ensure that data is protected and is not easily accessed by unauthorised users. It is easy to use and is cost effective as compared to other high end smart locks.

2: Features / Components of Secure IoT Lock

2.1: NodeMCU ESP-8266

NodeMCU ESP-8266 is an open-source Lua based firmware and development board specially targeted for IoT based Applications. We are using it as the processor that connects to the internet as well as using it to control the SG90 Servo Motor.

2.2: SG90 Servo Motor

SG90 Servo Motor is a small and lightweight servo motor with high output power, it acts as the lock component in our system and is controlled through the NodeMCU ESP-8266.

2.3: AWS SNS

Whenever a new OTP is generated, a message containing the OTP is sent to the registered phone number's SMS, which allows the user to gain access to the webpage to control the lock. It also acts as an alert for the owner if the message was sent despite not activating the lock. Letting the owner know that someone unauthorised is trying to access the lock.

2.4: Web Page Login System

For the user to unlock the lock, they will have to access a web server where there will be a login system for the user to enter their credentials (Username and Password) which they have preset.

3: Security Features of Secure IoT Lock

3.1: 2 Factor Authentication

After successfully logging in, a one time password (OTP) is generated. The OTP is sent to the user's registered phone number via their short messaging service (SMS) on their phone. The user must then enter the provided OTP into the webpage. If the OTP entered is correct, the user will then have access to the user interface to control the smart lock.

3.2: Account Lockout

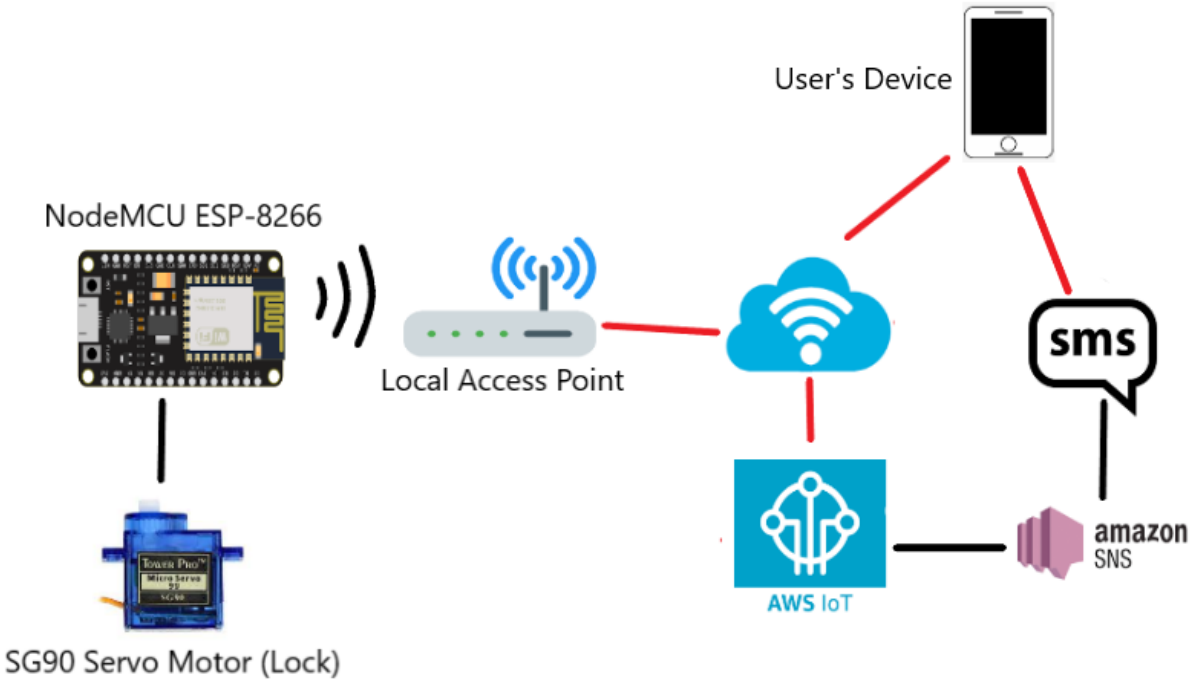
When a user wants to access the webpage, at the initial process, they will have to input their credentials into the login system, if they do not get the credentials right within 4 attempts, they will be locked out and put on a timeout so that they are unable to try new login attempts.

3.3: AWS Security

Secure IoT connects to AWS IoT using X.509 certificate over secure TLS connection. The certificate and its private and public keys are used to encrypt the data sent to AWS IoT. Data encryption as well as protection is used when sending the OTP code to the user's device, so that no external party can read the OTP. Logging and monitoring of connections and exchanges is also possible through the AWS SNS dashboard.

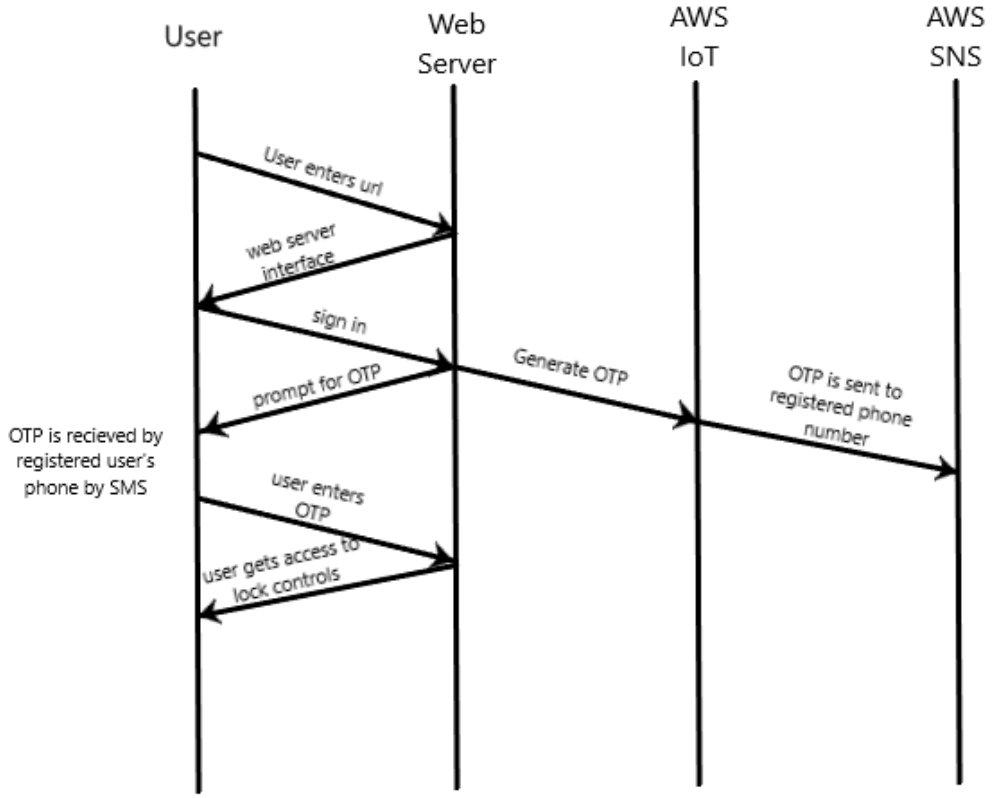
4: Diagrams

Network Diagram:

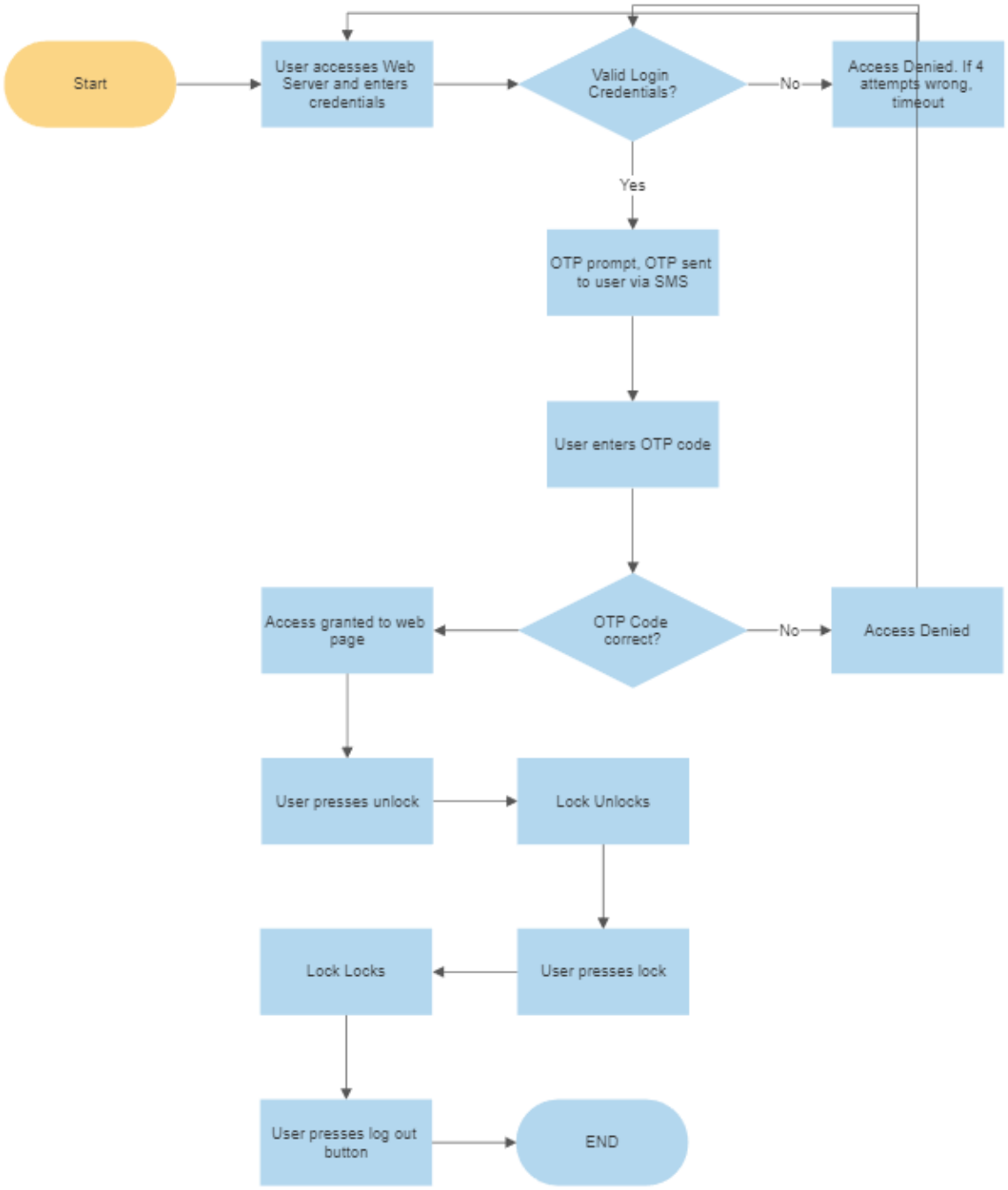


The SG90 Servo Motor which acts as the lock, is connected to the NodeMCU ESP-8266. The ESP-8266 is then connected to the Wi-Fi via the access point where they will have access to the web server. When the user enters the login credentials to control the lock, they will need to enter a OTP code for 2FA, sent via SMS from Amazon SNS service. After entering the correct OTP code, the user will then have access to the lock control.

Timing Diagram:



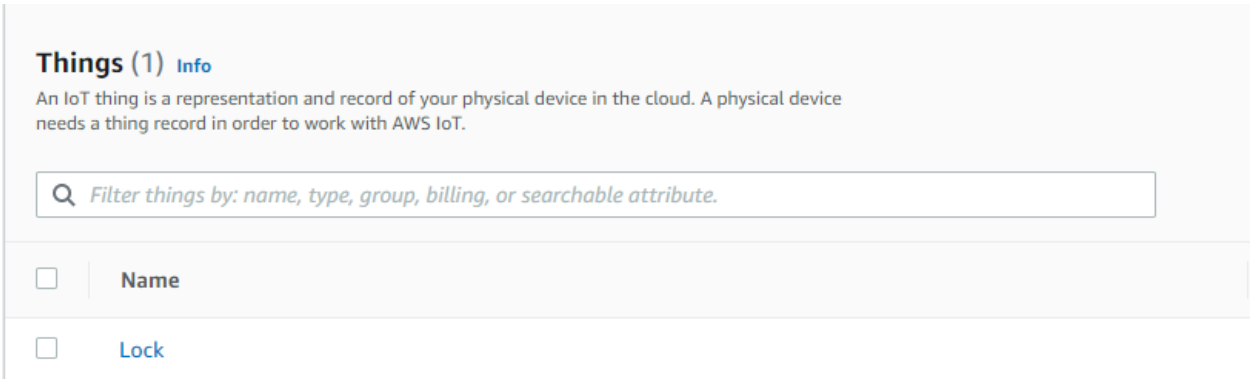
Flow Chart Diagram:



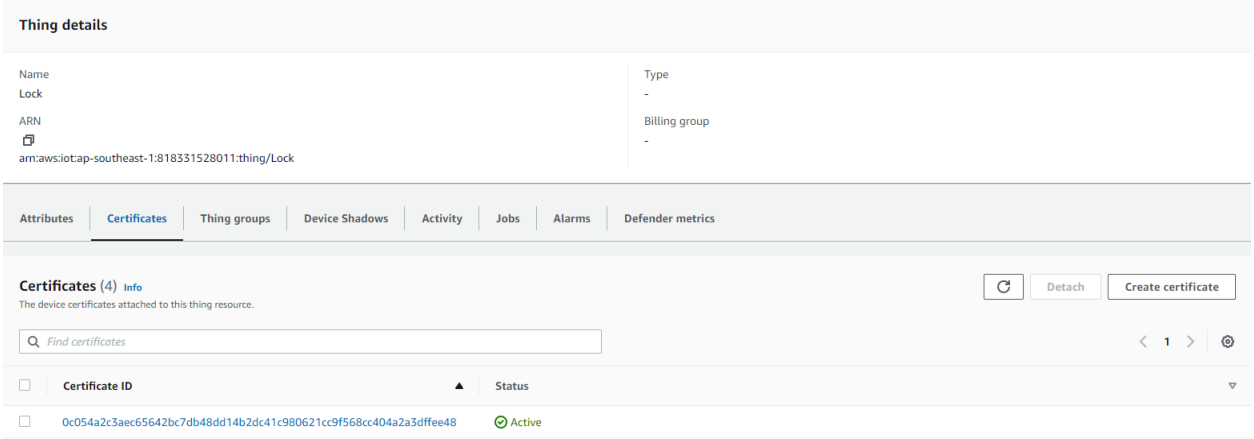
5: Code Pre-requisites

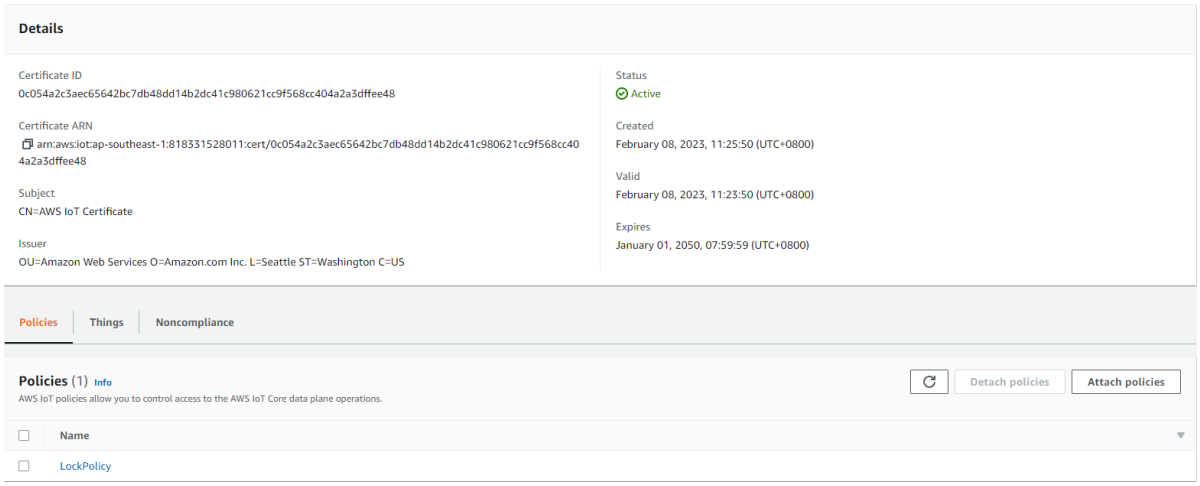
Before we started on our code we needed to do some stuff to enable the AWS MQTT as well as AWS SNS to work. As we have a random number generator in our code to provide the OTP, which will then be published to AWS to be used as the message for the SMS.

First we created a thing in AWS IoT Core.

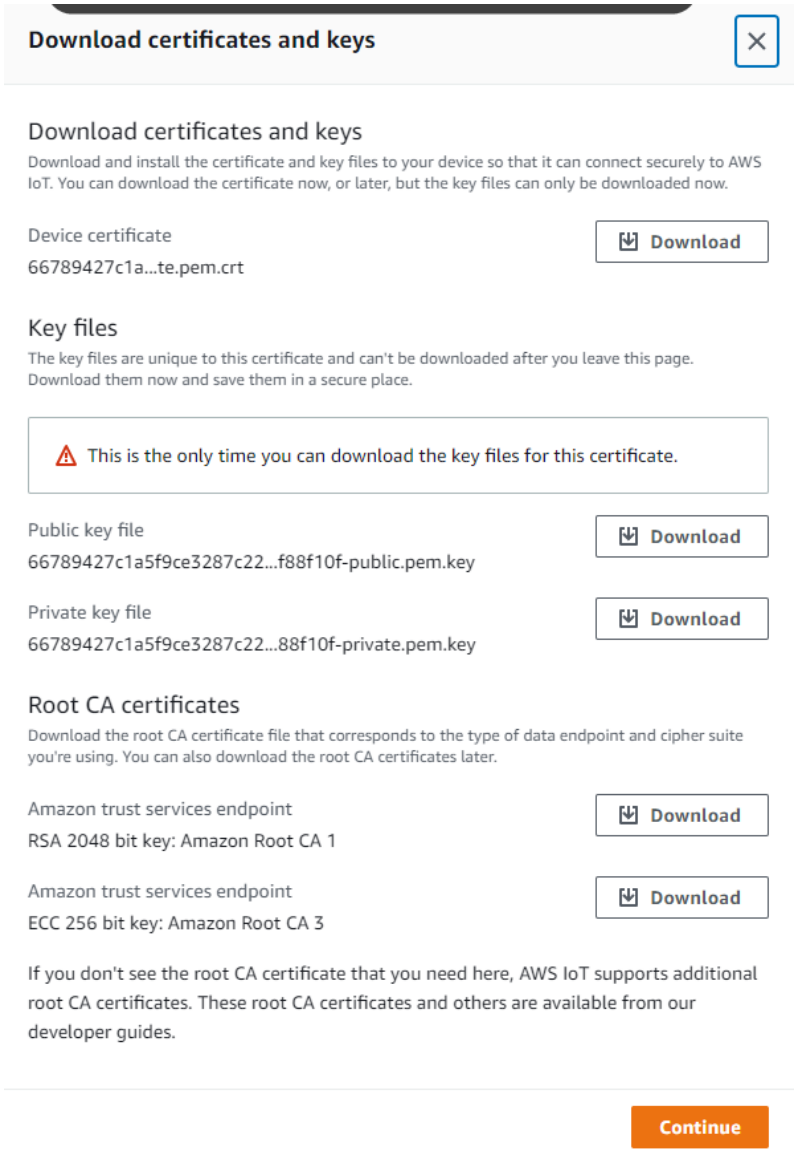


Next, we had to create the policies and download the certificates and keys to be used for the code for the thing to function as what our system needs.

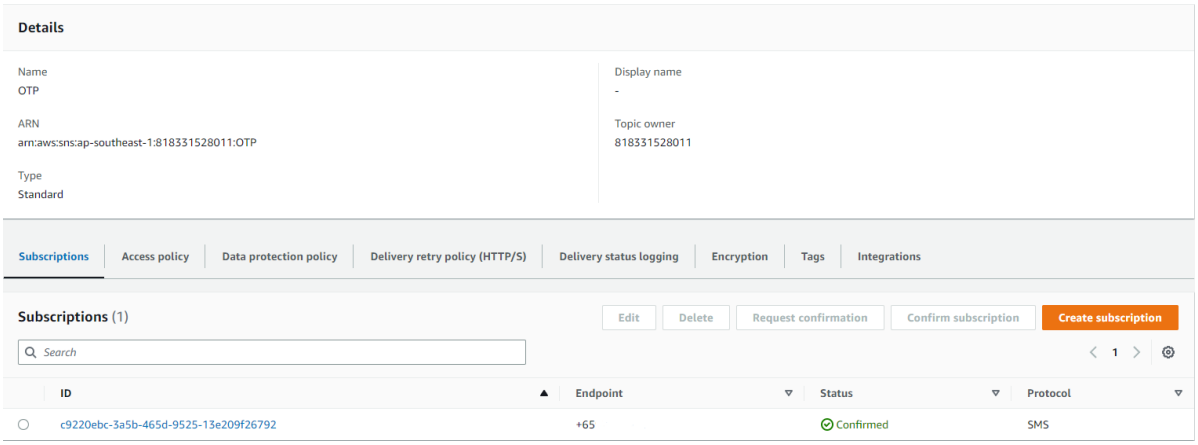




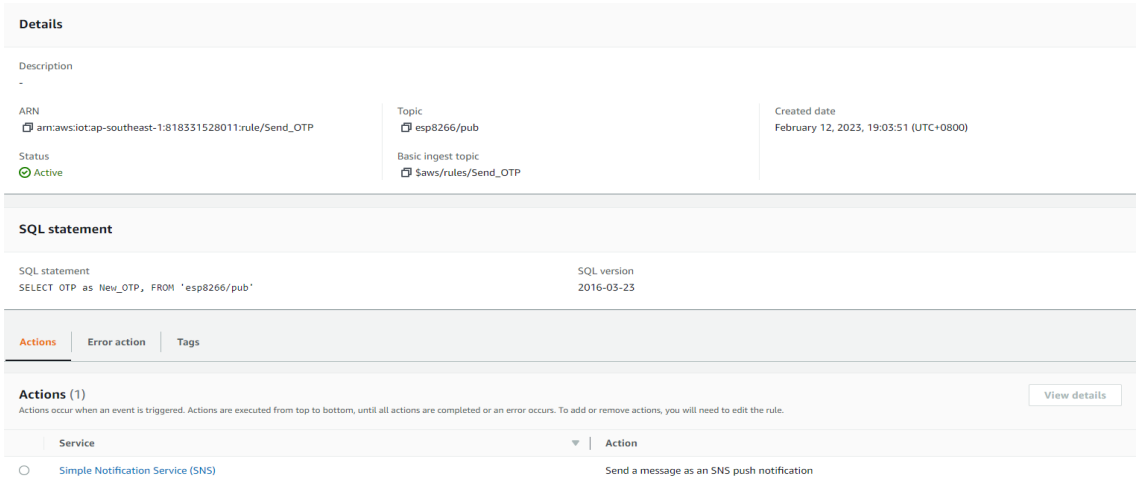
Example of certificate and keys:



Next, a topic is created in AWS SNS and the user's phone number is registered under it.



Next, for the user to receive the OTP code via SMS we had to create rules that links the MQTT topic where the OTP was published to, to be sent to the user. The SQL statement would get the OTP generated from the IoT device and route it to the AWS SNS which sends the message containing the OTP to the user's phone number which has been registered.



6: TR64 Compliance Checklist

Attack Surface	Check List	TR64 Reference	Description
Web Page	Device control on web page is secured against unauthorised access, 2FA used for access, lockout for repeated unauthorised login attempts	CK-IA-01 CK-AP-01	Unique username and password with 2-factor authentication used to access webpage, account lockout after too many unsuccessful logins
AWS	For MQTT, TLS is used to encrypt the connection between device and the broker. Data in transit to and at rest in AWS IoT is secured by using TLS and is encrypted	CK-NP-03 CK-DP-01	AWS message broker encrypts all communication in transit using TLS version 1.2 All data in AWS IoT at transit and at rest are encrypted
Hardware (SG90 Servo Motor & ESP-8266)	Housed in tamper resistant box, no exposed wires and ports	CK-AP-03 CK-AP-04	Hardware components are not easily accessed, wires and ports cannot be connected to
Security Audit	Logging of significant events	CK-AU-01	Logging of successful and unsuccessful attempts to access webpage, logging of lock control
System Wide	Conducted threat modelling to identify, analyse and mitigate as much threats as possible	CK-LP-01	Conduct threat modelling using STRIDE framework

7: Threat Modelling

Threat modelling checklist	Y/N	Supporting materials
Identify the potential target(s) to be protected	Y	Authorised users of the lock Access to the web server controlling the lock
Define the security problem	Y	Brute force attacks to login to the website and gain access to the lock (Elevation of privilege)
Conduct risk assessment	Y	High reproducibility, exploitability and discoverability
Determine the security objectives	Y	Unauthorised users should be stopped from gaining access to the lock
Define the security requirements	Y	Brute force attempts can be mitigated with a lockout Authentication of authorised users
Design and implement the capabilities	Y	Users will be unable to log in after multiple failed attempts User will need a SMS OTP to log in to access the lock
Validate and verify that the capabilities address the security requirements adequately	Y	Account lockout and OTP for 2 Factor Authentication works and is able to prevent unauthorised access into the webpage

8: Security Testing

We planned to use Kali Linux with the Aircrack-ng to find the password for WLAN. Aircrack-ng is a network software suite consisting of a detector, packet sniffer, WEP and WPA/WPA2-PSK cracker and analysis tool for 802.11 wireless LANs.

1. We will need a network adapter with monitor mode so that we can capture packets from networks we are not connected to.
2. We open a command prompt and type “sudo apt install aircrack-ng” to install the aircrack-ng.
3. We will type “sudo airmon-ng start wlan0”, this enables monitor mode.
4. Type in “airodump-ng wlan0” to show us the available BSSID, channel and more.
5. Take note of the BSSID or MAC address of the targeted network
6. Type in “sudo airodump-ng -c 1 -w Kali -bssid xx:xx:xx:xx:xx:xx wlan0”, the bssid being the bssid we took note of earlier.
7. We will deauthenticate the bssid so that when it tries to reconnect with the host, necessary information can be captured.
8. Open a new command prompt and type “airplay-ng -0 0 -a bssid wlan0”.
9. Finally, to find the password, we can go online to download a password wordlist, with many passwords in a text file.
10. Type “aircrack-ng -w wordlist.txt kali-01.cap” for the password matching process to start. The hash compares with passwords inside the text file.
11. Every line will convert into a hash, and when the dump file hash matches with the WPA/WPA-2 PSK, we will either have a successful match with which we can connect to the wireless network or the passwords in the wordlist will run out, leaving us without finding the password.

9: Additional Links

Project Wiki
https://sites.google.com/view/secure-iot-lock/home

Project Proposal
https://docs.google.com/document/d/1DXXDQlfyMb8tY1Nnp-OWDbWNy8GgiyAWns1oJdmGwAU/edit?usp=sharing

Project Slides
https://docs.google.com/presentation/d/1cWoRYavOhz6-To7Eb2qCSyHxeZzWL5jAP_BFfM4dIwM/edit?usp=sharing

10: References

1. *IMDA IoT Cyber Security Guide* (no date). IMDA. Available at: https://www.imda.gov.sg/-/media/Imda/Files/Regulation-Licensing-and-Consultations/ICT-Standards/Telecommunication-Standards/Reference-Spec/IMDA-IoT-Cyber-Security-Guide.pdf (Accessed: January 28, 2023).
2. Németi Florian *et al.* (2017) *IOT: L'émancipation des objets*, *Amazon*. Editions G. d'Encre. Available at: https://docs.aws.amazon.com/iot/latest/developerguide/iot-security.html (Accessed: February 8, 2023).
3. Treichler, R. and Hardmeier, C. (2005) *Amazon SNS Security*, *Amazon*. SAB Verl. Available at: https://docs.aws.amazon.com/sns/latest/dg/sns-security.html (Accessed: February 8, 2023).
4. *How to use aircrack in Kali? hacking the wireless network in 5 simple steps* (2021) *Learn Ethical Hacking and Penetration Testing Online*. Available at: https://www.hackingloops.com/how-to-use-aircrack-kali/ (Accessed: February 14, 2023).