Smart Security Door Lock with Pin Access

Project description:

This project is designed to control a door's lock. Omega terminal represents the door lock keypad. Laptop represents the system security administrator which acts as admin. The communication between admin and omega is done over AWS IOT. Admin sets the password for the door and sends it to omega through AWS IOT. This password is saved with omega and later used to check if a user is entering the correct password or not. If any user wants to open the door, he/she must enter their username first. Omega, which acts as door lock keypad asks for this username. The username entered by the user is sent over AWS IOT to admin. Admin has a list of usernames who can access this door. The username entered by the user is verified with the usernames contained at the admin's database. If the username is present in the list possessed by the admin, then it publishes to MQTT client that the username is valid. Omega on receiving this response asks the user for password. If the username is not valid or not present with admin, it sends a message that the entered username is invalid. Omega asks for the correct username again. This takes place until the user provides valid username. Once the username is verified, he needs to enter the correct password or PIN to open the door. In this case, servo motor acts as the door. When the PIN entered is correct, door gets unlocked. The servo motor rotates indicating the door has been opened. If the PIN entered is wrong the omega terminal asks for the correct PIN again. The limit to enter a wrong PIN is three times. If the password entered three times is incorrect, a notification is sent to the system administrator's phone and email via AWS IoT that the PIN has been entered incorrectly for three times. The system administrator can now take further action. In addition to this, the time at which the door was unlocked will also be displayed at the time of door unlocking. The time of unlocking could be displayed on the lock screen i.e., the omega terminal.

Project Overview:

The main aim of this project is increasing the security of a door lock. This project enables smart security locking system for a door where the user can unlock the door only by entering correct password. The password is set by the admin and nobody except the admin has access to the password and the list of users who can enter through the door. It starts of by the admin setting a password to the door. This password is sent to the omega terminal which acts as the door lock. The communication between the admin i.e. laptop and omega takes place over AWS IOT. The password set by the admin is published to a topic in MQTT client of AWS IOT. Omega, which is subscribed to the same topic acquires this password and saves it. When there is any user at the door, the omega asks for the user's username. Since the door should be accessed by only specific

users whose usernames are contained with the admin, the door will not get unlocked for any random user with their respective usernames. The user now enters his username. This username is collected by omega and sent to the admin over MQTT client. The username gets published to the MQTT client's specific topic and the admin which is already subscribed to that particular topic receives the user entered username. The communication between system security admin and door lock over AWS IOT is shown in Fig1.

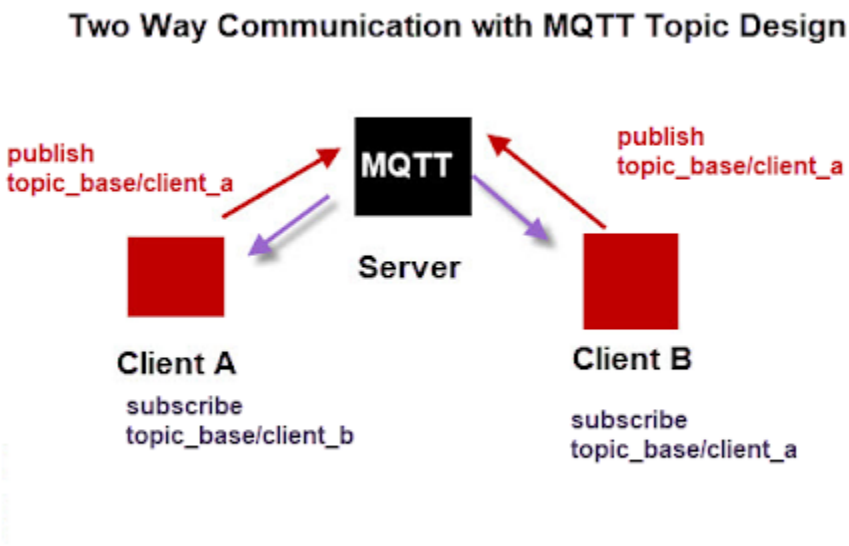

Figure 1: Two-way communication with MQTT

In our case, Client A is the admin (laptop) and Client B is the Omega (door lock). After the admin receives the username, it checks with the list of usernames contained with the admin. If the username is present in the database, then a valid signal is sent to the omega terminal over MQTT Client. If the username is not present in the list, then an invalid signal is sent to the omega. Until the user provides authorized username the door will not prompt for password. If the username provided is valid, then a message is displayed saying, "Username is Valid" and the door now asks for password. If the password provided by the user matches with the password sent by admin, the door gets unlocked. In this project the door lock keypad is represented by the omega and the door is represented by the servo motor. If the password is correct, a message is displayed that the door is being unlocked. The servo motor rotates indicating that the door has been unlocked. The Arduino code sends a confirmation that the door has been unlocked and the same will be displayed on the omega terminal screen – "UNLOCKED". The time and date of unlocking the door will also be displayed on the omega terminal. There are three different topics utilized in this project to ensure the safety of the password over MQTT. Any client subscribed to these topics will be able to see the messages being published on AWS IOT. To overcome this, three different topics are utilized to send messages over different topics. Figure 2 is small block diagram of how the publisher of MQTT works.

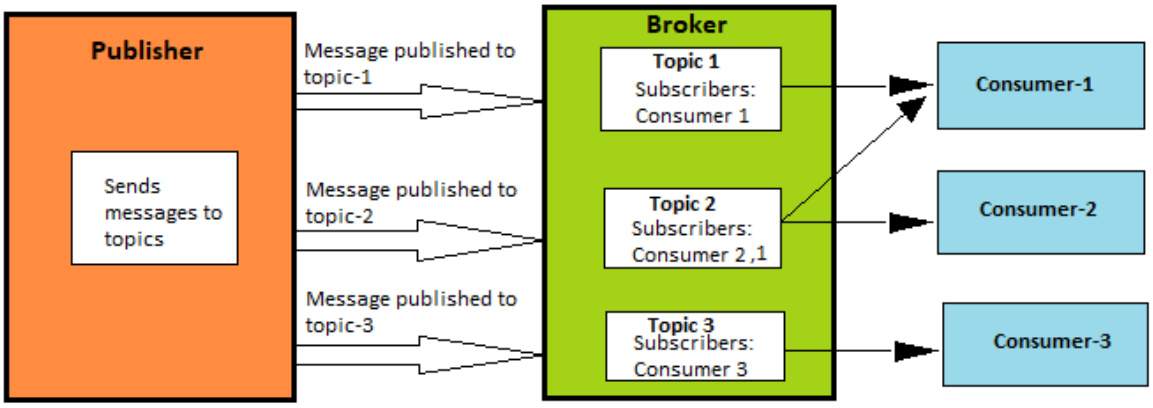

Figure 2: Communication between publisher and subscribers

Publishers are devices. Laptop and omega are the publishers in this scenario. Admin or Laptop publishes password initially, then omega publishes the user entered username, admin again publishes whether the username is valid or not. If invalid, the user is asked to enter the valid username again. It keeps asking for a username until the user enters the correct one. Same with the password. But there is a limit to the number of times a user can enter the password. Limit that has been used here is three. User has three chances to enter the correct password. If the password entered third time is also incorrect, the user is asked to contact admin. Also, an alert is sent to the admin's mobile phone and email that a user is trying to open the door but failed to enter the correct password thrice. Along with this information, the timestamp at which the wrong entry of password for three consecutive times is also sent in the SNS notification. The user who tried to open the door last can be viewed in the laptop's terminal i.e. the security system administrator's terminal. It is now up to the admin to take further action on this activity. If the user is genuine and password seems to be wrong, then the admin can either change the password or provide the present password to the user. Figure 3 has the overall diagram for this project.

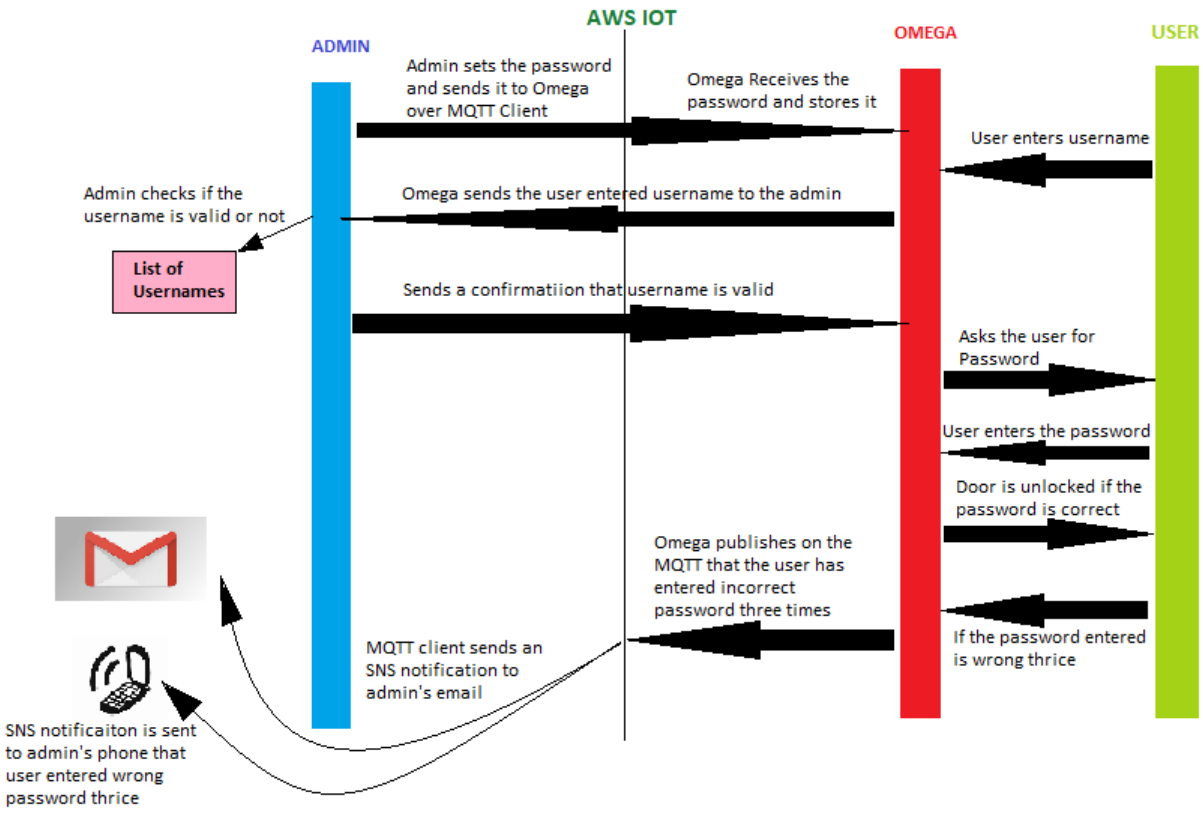

Figure 3: Smart_Security_Door Project overall diagram

Flowchart:

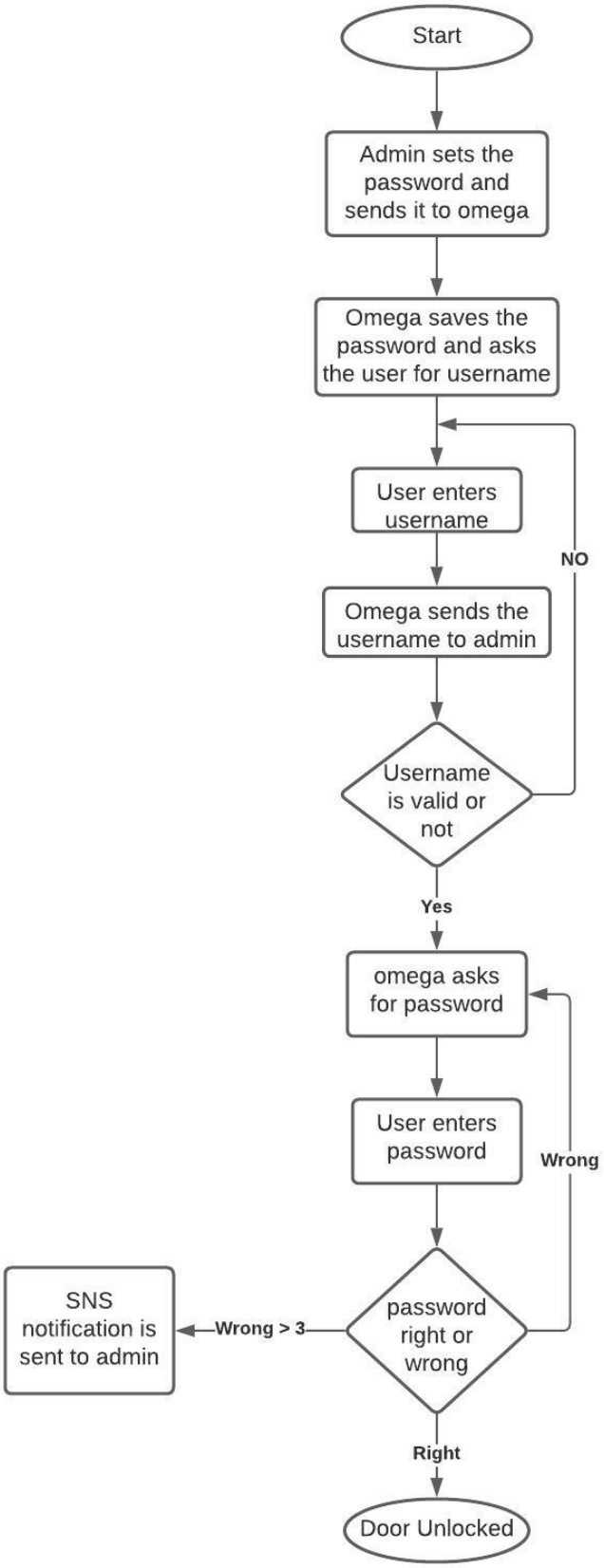

Figure 4: Flowchart of the project

List of materials:

Onion-Omega2 module, Onion Omega Arduino Dock 2, Grove Starter Kit, SG90 9g Micro Servo, Omega2 terminal, Pycharm community edition 2020.3.3.

Step by step instruction of the project:

1. Run the project_arduino.ino and update it to the omega.
2. The Arduino code gets uploaded:
   Rotates the servo motor when the password entered is correct indicating that the door has been unlocked.
3. Open AWS IOT and subscribe to topics- smart_security_password, smart_security_username, smart_security_valid and OmegaA366/alerts. Make sure that QoS is 0.
4. Run project_omega.py on the omega terminal.
5. Check for the configuration details and path of the certificates. Provide the correct paths if they are wrong.
6. The omega gets subscribed to the required topics and awaits the password from admin.
7. Run project_admin_Latop.py in local system.
8. The password gets published to the omega over AWS IoT. The admin set password here is 1234.
9. The omega terminal which represents the door lock keypad asks for the username.
10. Enter the username.
11. If the username is invalid the user is again asked to enter the correct username. The system keeps asking for username until a valid username stored by the admin is given. List of usernames contained by the admin are mb1001-mb1240, jm1001-jm1240, np1001-np1240 and sg1001-sg1240.
12. You can enter any username that falls under these sequences.
13. Once the username is valid the omega terminal asks for password.
14. Enter the correct password (1234).
15. If a wrong password is entered, you are asked to re-enter the password again. This takes place three times. If the password entered is wrong thrice, then the user is asked to contact admin and an SNS notification is sent to phone number +15854068839 and email - jm6433@rit.edu.
16. The SNS message consists of the timestamp showing the date and time at which the user entered wrong password. The admin's console displays the username of that user.
17. On entering the correct password, a message saying, "Unlocking the door…" is shown. The servo motor rotates stating that the door has been unlocked.

18. Once the door is unlocked Arduino sends an acknowledgement – “UNLOCKED”. The time of unlocking the door is also printed.

Limitations and challenges:

The communication between admin and omega takes place over AWS IoT through MQTT Client. If the communication was one way, then it could have been an easy task to just keep sending information from one end and keep receiving at the other. Since this is a two-way communication, every time a message is published to a topic, all the functions subscribed to the topic gets called and take in the information. This leads to confusion as there are various steps of interaction between the client to server and server to client. We intend to call a specific function, but all the functions get called. To avoid this, I made use of three different topics. Password is sent from the admin to the omega over first topic. Username received by the omega is sent to the admin over second topic. Admin checks for the validity of the username and sends the confirmation over third topic to the omega. If the password is entered wrong thrice, then omega sends this information to MQTT over fourth topic which sends an email and SNS notification to the admin’s phone.

Usage of different topics for this multi-layered communication also strengthened the security of the system. Since there are various number of channels of communication, it would be difficult for the intruders to know which topic to subscribe and know the password or username details.

Another challenge I faced while working on this project was importing the list of usernames to the admin’s system and checking for the valid ones. As the list of usernames gets bigger it might take more time to check through the data, but I made sure that the efficiency while checking for the valid username is high.

Apart from all the above-mentioned limitations, sending SNS notification to the phone and email was the most challenging part. The subscriptions to the topic had to be proper and the selection of the text from MQTT to SNS had to be taken care of as it does not take all formats of information.

Conclusion:

Smart security door lock with pin access is a project that helps in upgrading the security system of a door. It allows only a particular set of users to pass through the door with the help of correct PIN or password. If any user tries to tamper with the lock system by giving number various combinations of passwords, admin gets notified about this activity. Working on this project gave me an exposure to AWS IoT, publishing messages to MQTT Client, receiving the published messages by clients or devices, enabling communication between two clients over several MQTT topics, how to make use of IoT devices to improve the modern lifestyle, improve the security of

household system and various other topics that deal with IoT. I learnt how to communicate from Omega2 to Arduino dock, make use of python to code on omega and send instructions to Arduino dock that controls devices and receive input from the IoT devices likewise. This project taught me how to make use of a code to connect various devices in such a way that an SNS notification and emails could be sent to our personal phones and emails using AWS IoT when there is any discrepancy or alert that we should be intimated about. It was a knowledgeable learning working on this proposal and future work on the same could be much more challenging. Future work could be designed in such a way that the access key is face recognition or iris scan in the place of number lock. Also fingerprint scan is also a great initiative to take this project further. This might involve Machine Learning and Artificial Intelligence in combination with IoT. It might be more complicated than it seems to be but not unachievable. Future work as proposed above could ensure security and better lifestyle for the mankind.

References:

1. Bahga, A., & Madisetti, V. (2015). Internet of Things. http://www.internet-of-things-book.com/
2. https://aws.amazon.com/
3. Connecting to AWS IoT core. (2021). https://docs.aws.amazon.com/iot/latest/developerguide/connect-to-iot.html
4. AWS IoT Tutorials. (2021). https://docs.aws.amazon.com/iot/latest/developerguide/iot-tutorials.html\
5. Send an Amazon SNS Notification. (2008). https://docs.aws.amazon.com/iot/latest/developerguide/iot-sns-rule.html
6. Rules for AWS IoT. (2021). https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html
7. Republish an MQTT message. (2008). https://docs.aws.amazon.com/iot/latest/developerguide/iot-repub-rule.html