SECURED DOORLY

Team Member 1: Soh Tai Rong (2032766)
Team Member 2: Brian Lim Beng Xian (1908873)
Team Member 3: Keegan Lim Beng Yang (1908857)

Link to Project Wiki: https://github.com/BLBX-7/ET0731_Brian_Keegan_TaiRong

1. OVERVIEW

Secured Doorly is a smart lock system for homes that removes the need of using physical key and lock system or RFID cards to gain entry. Although keys, keypad and RFID cards have been effective and convenient methodologies for users to gain access to their homes, we want to take this to the next level. Secured Doorly will provide authorized users a way to unlock their home door remotely from a mobile app securely.

2. DESIGN

2.1 Resources

Hardware:

1. ESP8266
2. Servo Motor
3. ESP32-CAM
4. Android Mobile Phone

Software:

1. Arduino IDE
2. Android Studio IDE

Cloud:

1. Amazon Web Services – Simple Email Service (AWS – SES)
2. Microsoft Azure Cloud – SQL Server
3. ThingSpeak IoT Cloud – MQTT Server

3.2 System Architecture

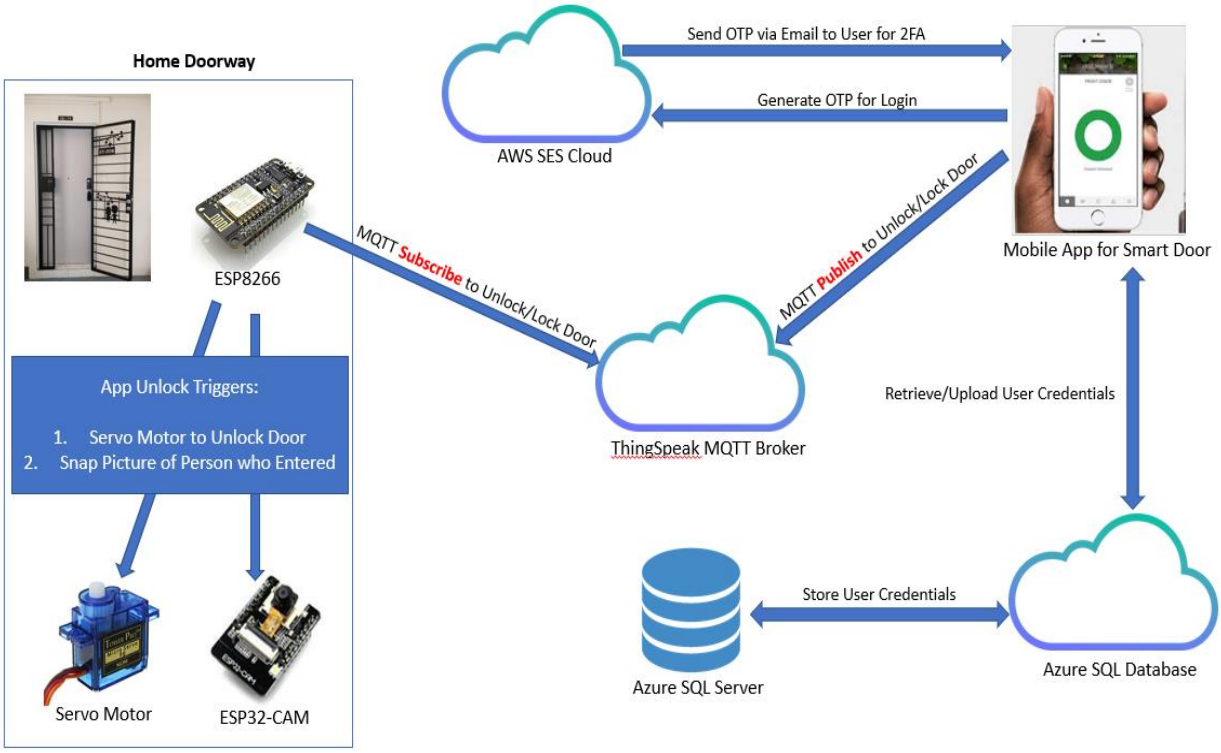


Figure 1: System Diagram

2.3 System Integration

Resources	What It Does?
ESP8266	Subscribe to MQTT Broker and receive data from mobile app to trigger servo motor to unlock/lock door and ESP32-CAM to initiate picture taking.
ESP32-CAM	Camera that takes picture of person that enters door regardless of whether it is an authorized or unauthorized person.
ThingSpeak IoT Cloud	Works as our MQTT broker to enable publish/subscribe from our clients which is the ESP8266 and Android Mobile Phone.
Android Mobile Phone	Application with secure sign in to allow user to unlock/lock their door remotely.
Servo Motor	Used as our lock to unlock/lock smart door.

Microsoft Azure Cloud	Works as our SQL database where user login information (like hashed email and password) for mobile application are being stored.
AWS SES Cloud	Used as our medium for two-factor authentication. OTP is generated from mobile app and is sent to AWS SES cloud. The OTP will then be emailed to the authorized user to login into app.

Table 1: System Integration Table

4. SECURITY FEATURES

Android Mobile Application:

1) **Two-Factor Authentication:**
   - Enforces a mandatory form of two-step authentication, via email OTP, on login for mobile application to prevent identity spoofing.

2) **Enforce Strong Password for Users:**
   - To ensure that users do not use weak passwords that may be easily accessed by threat actors.

3) **Account Lockout for App Login:**
   - To prevent unauthorized users from brute forcing their way into accessing the mobile app.

4) **Hash User Credentials Stored on SQL Database:**
   - Hashed user's username and password for mobile application login with SHA-256 hashing algorithm. This ensures that, in the event of confidentiality is compromised, vital confidential information won't get exposed (hashing is done in the application itself, before being sent, on a secured channel, to Azure SQL cloud database for storing.

Cloud:

1) **Azure Cloud Security:**
   - To protect customer data and account, strict IP firewalls are set to only permit access to predefined IP addresses into the SQL server for data management.
   - SQL server authentication using username and password to only authenticate authorized users. User password used meets the complexity requirement recommended by OWASP.
   - SQL databases are encrypted through the use of Azure's Transparent Data Encryption (TDE). This ensures encryption of data at rest.
   - SQL database secure customer data in transit through the use of TLS.

2) **AWS SES Security:**
   - Opportunistic TLS is used. It means that AWS SES always attempt to establish a secured connection to the receiving mail server (GMAIL in our case).
   - Use of MFA with our account: Password meets the complexity requirements recommended by OWASP and enabled Google Authenticator used for MFA.

Physical:

1) **For Door:**
   - Every time the door opens, a picture of the person who unlocked it will be taken. In the case where an unauthorized person has managed to unlock the smart door and enter the house, the user may look at the stored pictures to identify what the person may look like and provide the image to the authorities.

2) **Secured MQTT Publish/Subscribe:**
   - Connection between ESP8266/Android Mobile Phone and broker will be encrypted via TLS to ensure confidentiality. In addition, a certificate thumbprint is used to prevent publish/subscribing to an unintended MQTT server, hence, ensuring that it is the authentic ThingSpeak server.

5. JUSTIFICATION

Secured Doorly is an upgrade from using a physical key and lock system, keypad as well as RFID card entry, for homes. This is because Secured Doorly removes the need for users to carry around a key or card with them, which may be misplaced or stolen, allowing unauthorized users to gain entry into their homes. Thus, users can unlock their home door on a secured app from anywhere allowing authorized users, such as themselves into, their home. Following this, in a situation where the user is not home and a family member or trusted friend wishes to enter the home, the user may unlock the door for them remotely which is more convenient.

As for the cloud platforms chosen for our implementation, each one of them has its perks. Microsoft Azure Cloud is simple and convenient for users, it also poses as one of the most secured cloud platforms for data at rest. Since we are using it to store confidential information of our mobile app user, security is extremely important and the notable security features have been listed in the section above.

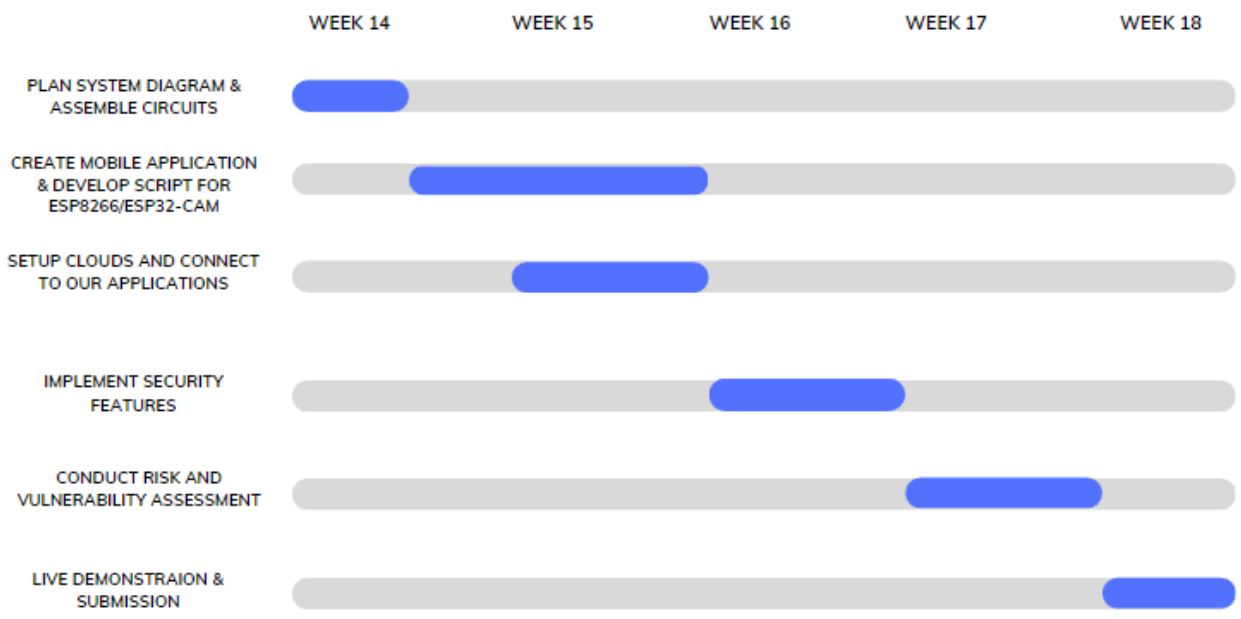
Amazon Web Services (AWS) was chosen as our platform to handle sending OTP via email to our mobile app users to login. AWS provides a simple email service (SES) via SMTP which allows us to send emails to our users securely as AWS SES uses DKIM, DMARC and SPF, all of which are email authentication protocols. In addition, AWS SES provides encryption via TLS, hence our data in transit is protected which is important since we are using it to deal with our two-factor authentication method.

Figure 2: AWS SES features

ThingSpeak IoT Cloud is chosen as our MQTT broker due to its great compatibility with the ESP8266 and its simplicity. It also enables the use of TLS which important for protecting our data in transit. In addition, the MQTT broker is secured as only permits access to the devices that use the generated credentials which can only be used by a single device/user once.

6. TIMELINE

Below is the Gantt chart we will be following to implement Secured Doorly to ensure the most optimal procedure to complete the project in time.



REFERENCES

Azure IOT Central - IOT solution development: Microsoft Azure (no date) *IoT Solution Development | Microsoft Azure*. Available at: https://azure.microsoft.com/en-us/products/iot-central/#overview (Accessed: February 5, 2023).

Fang, J. (2022) *17 Best Digital Locks in Singapore if you're always forgetting your keys, with prices included*, *TheSmartLocal*. Available at: https://thesmartlocal.com/read/digital-locks-singapore/ (Accessed: February 4, 2023).

Metz, M. (no date) *The SES, Amazon*. Watts. Available at: https://aws.amazon.com/ses/ (Accessed: February 5, 2023).