ET0731

IOT Security

Group members

Members	Admin No.
Christine Wong Tien Lee	P2037295
Darren Tan Kai Jen	P2032980
Seng Cheong Long Gabriel	P2032357

Project Proposal

1. Introduction

1.1 Problem Statement

Traditional locks and keys have many defects such as easy penetration and poor security. It is susceptible to break ins especially since experienced burglars can break into a house in seconds by picking the lock. Furthermore, the possibility of losing keys is very likely due to human error. When someone carries a bunch of keys, there is a high chance that the keys will be stolen or lost. Research also shows that 6 in 10 of us have left a property and returned to find we have left the door unlocked due to carelessness.

1.2 Overview

SmartDoor is a reliable door entry system that upgrades your home security with more safety, security and convenience. It is protected by advanced security protocols such as cryptosystem and multi-factor authentication in order to prevent the door entry system from getting hacked.

2. Features of the product

2.1 Mobile Application

SmartDoor comprises of a mobile application for unlocking your door remotely and it has the ability to allow someone to connect to the door lock (ESP32) and generate a temporary passkey when approaching his home. SmartDoor application must send the generated temporary passkey to ThingSpeak so that ESP32 will be able to retrieve it for further authentication in order to unlock the door.

2.2 Multi-Factor Authentication

User must use a registered smartphone to connect to the ESP32. If the **BLE connection** between the registered device and ESP32 fails, the door cannot be unlocked, even if the user enters the correct password. In order to generate a temporary passkey and upload it to ThngSpeak, **fingerprint** authentication is required. If the BLE connection is successful, the ESP32 has to **compare the entered password with the temporary passkey obtained from ThingSpeak**. If the entered password matches the temporary passkey obtained from ThingSpeak, the door will unlock.

Note: 3 authentication methods (bold text) need to be successfully verified to unlock the door.

2.3 RSA (Rivest-Shamir-Adleman)

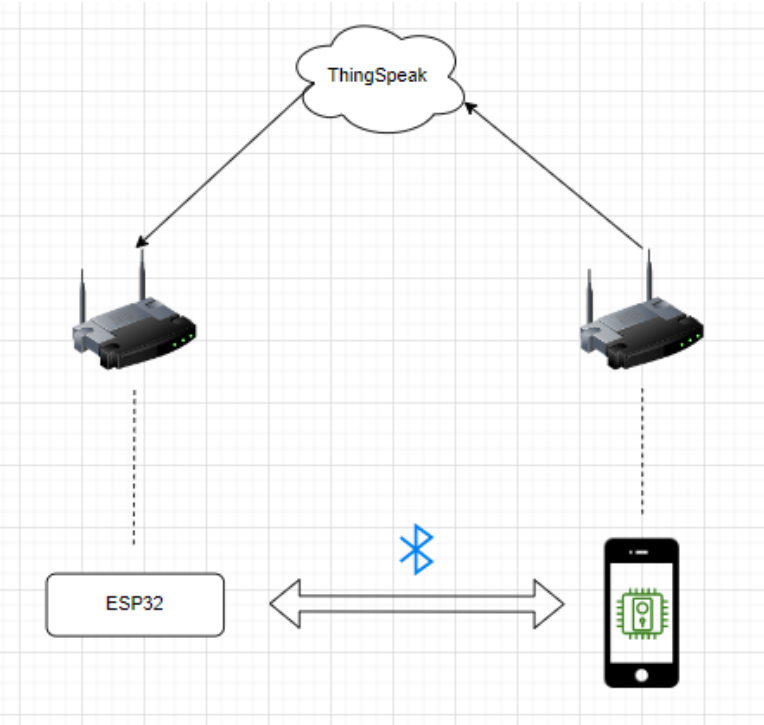
RSA is a public key cryptosystem that will be implemented to secure the data transmission between devices (ESP32 and mobile application) and ThingSpeak. In this project, mobile application must encrypt the generated temporary passkey and send it to ThingSpeak. ESP32 needs to decrypt the encrypted temporary passkey from ThingSpeak to get the original temporary passkey. It is difficult to crack through attacks such as brute force and factoring due to the large prime factor numbers. This reduces the likelihood of unauthorized access, while other common vulnerabilities including eavesdropping and replay attack will be rendered useless.

3. How it works

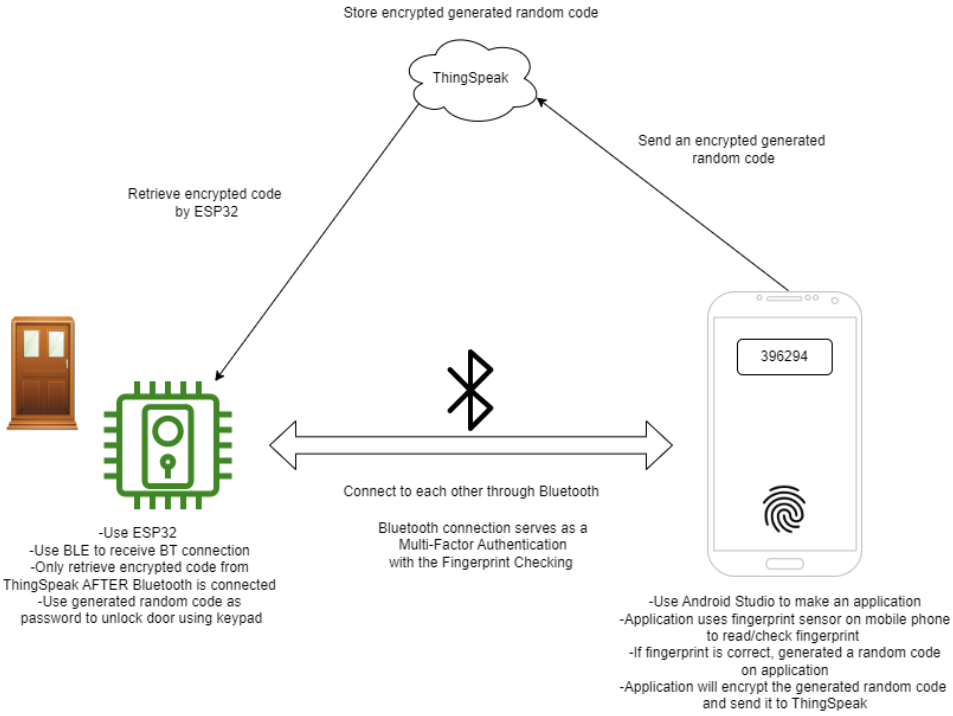
User wants to unlock his house door from approximately 15m away. He opens the SmartDoor application and connects it to the ESP32 within the door lock system. Once the devices have connected successfully (BLE connection), SmartDoor application will prompt him to use his fingerprint for authorisation in order to generate a temporary passkey. The fingerprint is the same as the one used to unlock the phone and is registered in his phone beforehand or the phone security pin can be used instead. SmartDoor application will also encrypt the generated temporary passkey and send it to ThingSpeak. The encrypted temporary passkey will be stored as a field in ThingSpeak. Meanwhile, ESP32 retrieves the encrypted temporary passkey from ThingSpeak and decrypt it to acquire the original temporary passkey. All the above steps should be able to complete just as the user nears the door, and then he will need to enter the password by using a built-in keypad. ESP32 compares the entered password and the temporary passkey. If they are the same, the door will be unlocked.

4. System diagrams

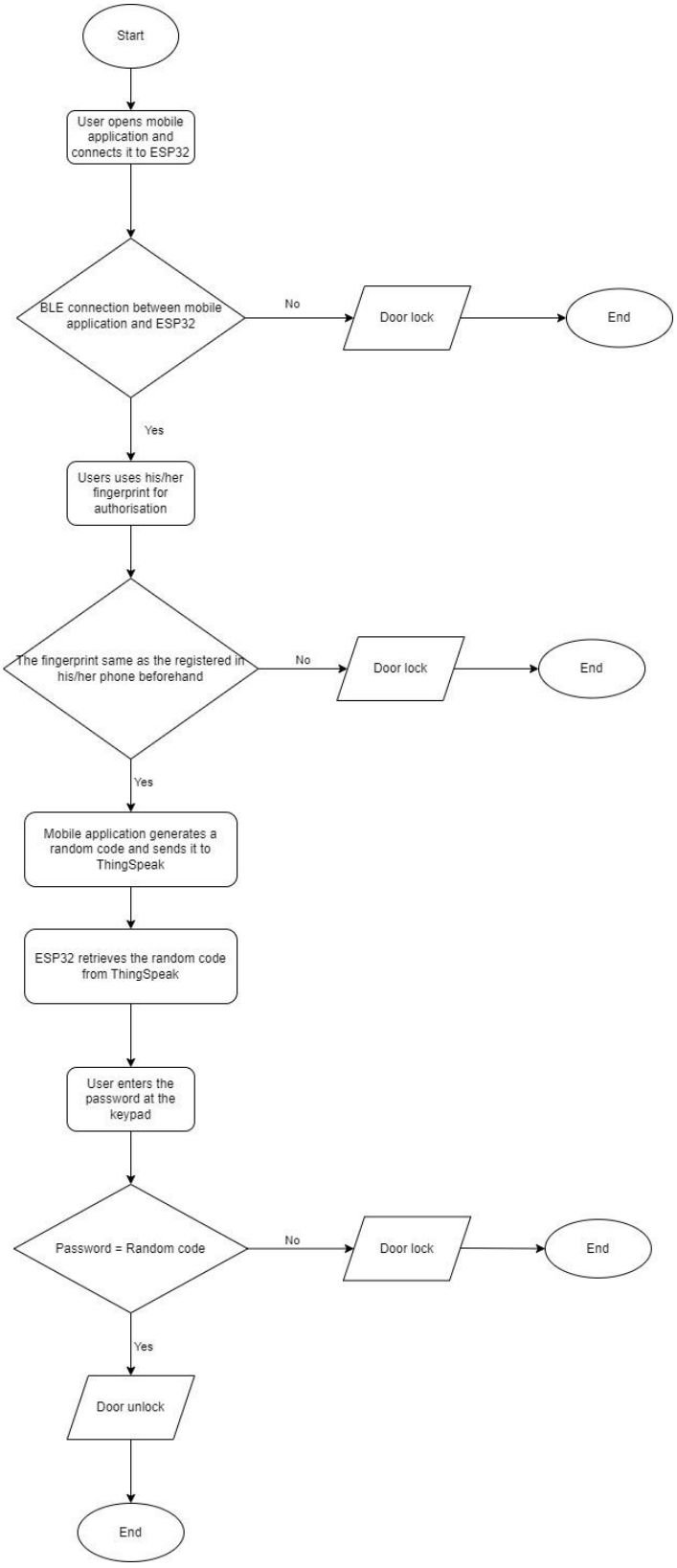
4.1 Network Diagram



4.2 Data Flow Diagram



4.3 Flow Chart

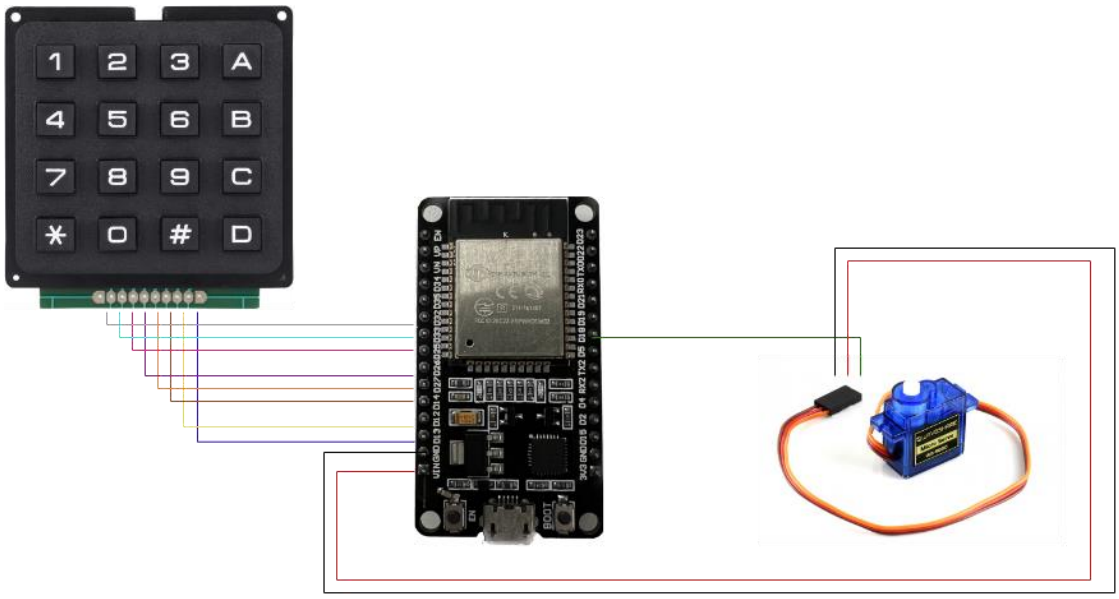


5. Components and Wiring

5.1 Hardware

- ESP32
- Keypad
- Servo motor

5.2 Schematic diagram



ESP32 and Servo motor wire connection:

ESP32	Servo motor
GND (Black)	Ground (Brown)
VIN (Red)	5V (Red)
D18 (Green)	Signal (Orange)

ESP32 and Keypad wire connection:

ESP32	Keypad
D32	C1
D33	C2
D25	C3
D26	C4
D27	R1
D14	R2
D12	R3
D13	R4

6. Gantt Chart

	Week 14	Week 15	Week 16	Week 17	Week 18
Research					
Plan					
Write program (ESP32 & Android Studio)					
Debug					
Kali Linux					
Prototype					
Wiring					
Report					
Presentation					

7. References

70% of US forget if we've locked our doors - build magazine (2018) *Build review*. Available at: https://www.build-review.com/2018-70-of-us-forget-if-we-ve-locked-our-doors/ (Accessed: January 31, 2023).