Smart Lock

by Briana Grant

Table of Contents

Definition

- **What is a Smart Lock?**
    - "an electromechanical lock that is designed to perform locking and unlocking operations on a door when it receives when it receives a prompt via an electronic keypad, biometric sensor, access card, Bluetooth, or Wi-FI from a registered mobile device."
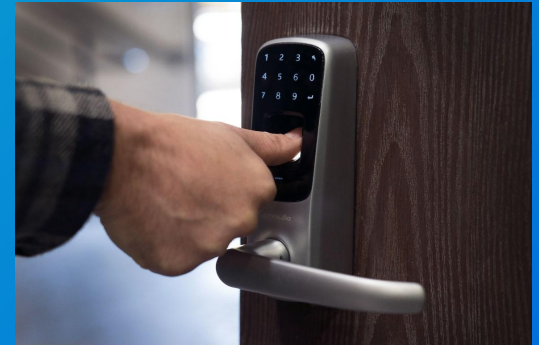
Key Concepts

- **_Internet of Things (IoT)_**

  - the network of physical objects—**"things"**—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet - Oracle

  - Everyday items, household appliances, vehicles, medical devices
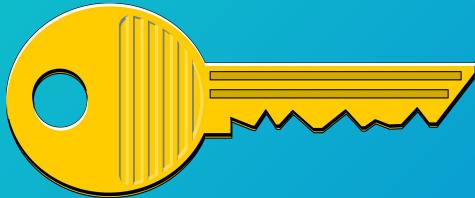
Key Concepts (Cont.)

- *Radio Frequency Identification (RFID)*

  - A device that uses radio waves to identify people or objects.

  - Consist of 2 components: A tag and a reader

    - **Reader**: A device that reads information contained in a wireless device (**Tag**) w/o making any physical contact or requiring a line of sight.
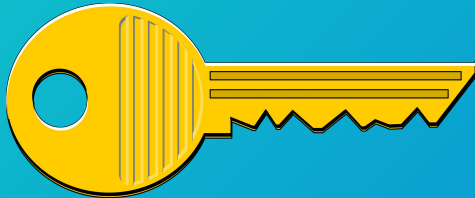
Issues

RFID

- Security Challenges
  - Few standards and rules are in place to direct IoT security
- Vulnerabilities
  - Open-source code vulnerabilities
  - Malware Risk
  - Data Leaks
- RFID tag data can be read by anyone (lack of privacy)

IoT

- Security Challenges
  - Data Privacy
  - "5V Challenges" (volume, veracity, variety, value, & velocity)
  - "Trillion Points of Vulnerability" (CSNDSP 2018)
- Vulnerabilities

Project Scope

- **Purpose:** To provide access to private storage or rooms without utilizing a traditional lock and add convenience for users with the use of **IoT and RFID**

- Utilization of RFID Reader and Tag in order to operate a mechanical lock (Smart Lock)

Implementation - System Configuration

Implementation - Software Configuration

- Arduino
- Libraries
  - SPl.h - allows you to communicate with SPI devices, with the Arduino as the controller device.
  - MFRC522.h - Arduino RFID Library for MFRC522. Allows you to Read/Write a RFID Card or Tag

Implementation: Hardware Configuration

1. Arduino
2. MFRC522 (RFID Reader)
3. RFID Tag (Access Card)
4. Solenoid Lock
5. Jumper Wires
6. 12V Power Source
7. Relay
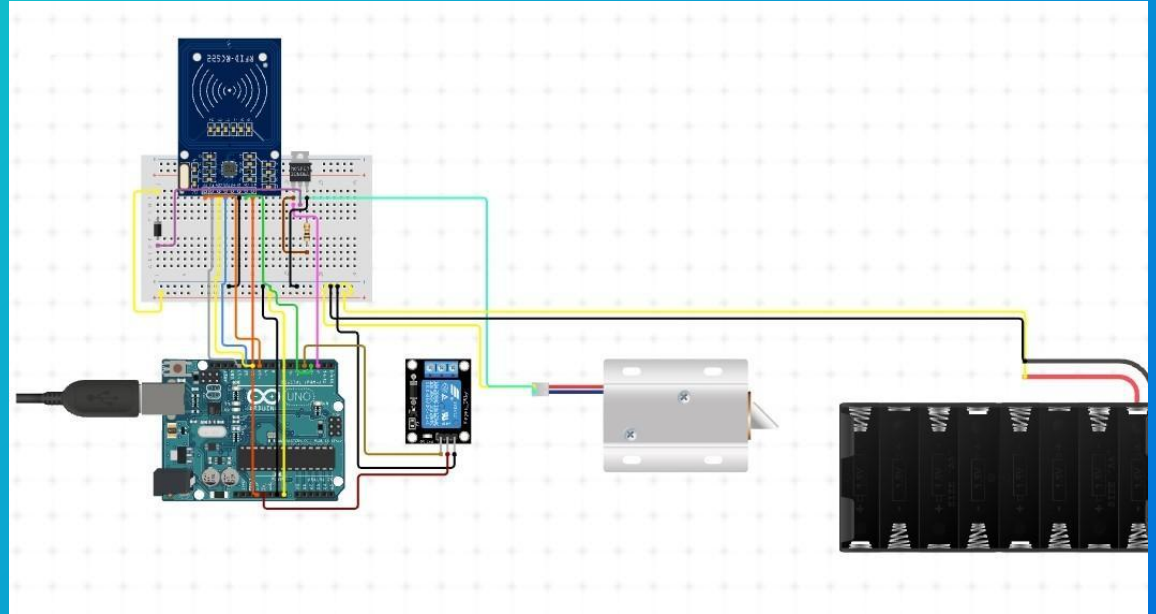
Code Implementation

1. Create instances for Relay, MFRC522, and PIN(s)

1. Setup

   a. Initialize MFRC522 (RFID Reader)

   b. Initialize SPI bus

   c. Initializer Serial Communication

```cpp
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9

#define RELAY 3 //relay pin

#define ACCESS_DELAY 2000
#define DENIED_DELAY 1000
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance (RFID Reader)

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);   // Initiate a serial communication
  SPI.begin();          // Initiate  SPI bus
  mfrc522.PCD_Init();   // Initiate MFRC522

  pinMode(RELAY, OUTPUT);

  digitalWrite(RELAY, HIGH);
  Serial.println("Put your card to the reader...");
  Serial.println();
}

void loop() {
  // put your main code here, to run repeatedly:
  // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
```

Implementation (Cont.)

3. Read Tag UID

4. Loop Conditional Statements

    a.   First check if tag is new or unavailable

    b.   If card has valid UID, read UID open solenoid, return message

    c.   Else, read UID, do nothing, return rejection

```
//Show UID on serial monitor
Serial.print("UID tag :");
String content= "";
byte letter;
//Prints RFID UID (Tag ID)
for (byte i = 0; i < mfrc522.uid.size; i++)
{
   Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
   Serial.print(mfrc522.uid.uidByte[i], HEX);
   content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
   content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
Serial.println();
Serial.print("Message : ");
content.toUpperCase();

 //if valid card with valid UID is presented, authorized message
returned
 if (content.substring(1) == "63 E2 4B 94") //change here the UID of the
card/cards that you want to give access
 {
   Serial.println("Authorized access");
   Serial.println();
   delay(500);
   digitalWrite(RELAY, LOW);
   //Solenoid Unlocked
   delay(ACCESS_DELAY);
   digitalWrite(RELAY, HIGH);
 }

 else  {
   //if invalid card is presented, rejection message returned
   Serial.println(" Access denied");
   delay(DENIED_DELAY);

 }
   //Reference :
 }
```

Assessment (Demo)

Limitations

- Arduino doesn't have it own power supply, currently requires USB adapter

- Code is currently configured for one unique tag, not multiple

- Lock not manual (No manual button implemented)

Conclusions

- Project can be adapted to include WIFI or bluetooth capabilities for opening Smart Lock via mobile device

    - Can provide added convenience

- Smart Locks are a widely available IoT device

- Security Risk

References

- Farhan, Laith, Kharel, Rupak, Kaiwartya, Omprakash, Quiroz, Marcela, Alissa, Ali and Abdulsalam, Mohamed (2018) A Concise Review on Internet of Things (IoT)- Problems, Challenges and Opportunities. In: 11th International Symposium on Communication Systems, Networks, and Digital Signal Processing (CSNDSP 2018), 18 July 2018- 20 July 2018, Budapest, Hungary.

- "Libraries," Libraries - Arduino Reference. [Online]. Available: https://www.arduino.cc/reference/en/libraries/. [Accessed: 02-Nov-2022]

- "What is the internet of things (IOT)?," What Is the Internet of Things (IoT)? [Online]. Available: https://www.oracle.com/internet-of-things/what-is-iot/. [Accessed: 02-Nov-2022]

- https://www.hackster.io/mertarduino/solenoid-door-lock-contol-rfid-527505