Biometric Automobile Ignition Locking System

Guguloth Thukaram
PG Student, Department of Electronics & Communication Engineering, Gokaraju Rangaraju Institute of Engineering & Technology, Hyderabad, Telangana State, India.

Abstract—Automobile theft is the biggest problem in the remote location of the city and neither key lock nor remote keyless system provides reliable solution because key can be copied very easily and remote keyless system encrypted data use radio waves which can be recorded and used to unlock the car. To design a unique key which doesn’t rely on key or radio wave, biometric solution is the only better option. Our purpose here is to provide biometric solution with very low cost hardware and using open source hardware and software too plus it self-installation.

Index Terms—Automobile theft security, automobile ignition control, open source hardware and software, do it your self-installation.

1. INTRODUCTION

Every person has its own unique fingerprint which can be used as unique unlock/lock key of your automobile. This can solve the problem of key theft where your automobile can only be started when it detects unique fingerprint assign to it. Thus the control of the ignition coil and accessories connection of automobile is required.

There are many devices available to protect your automobile to prevent the theft but many of the solutions are available to few international brands and the installation is costly when it comes to third party. Using an open source hardware and software with some knowledge of your automobile allows us to design our own little but reliable security system.

2. MARKET SURVEY ON ANTI-THEFT DEVICES

There are many devices available in the market to give you good antitheft solution. Some of easy install and cheap solution is mechanical immobilizers [1] like steering wheel lock, hood lock, tire lock, gear shift lock, ignition/steering wheel column lock and brake pedal lock. There is electronic version of immobilizers available which can lock the ignition system or fuel system until special key combination (using some keypad) or combination of various pedals are entered.

Mechanical immobilizers are cheap but not so good solution when car key is stolen. While good electronic immobilizers are available with higher price with extra installation and maintenance charge.

The article provided by Delhi police does not give you effective guidance on antitheft [2], thus it is necessary to design a device that lock our automobile with unique key, a biometric locking system.

3. SURVEY ON AUTOMOBILE THEFT

There is a good article written by Indian express [3] about automobile theft in metro city like Delhi. In Delhi 2 case is registered in every 30 minutes. Software-Savvy, Tech-Friendly Thieves Armed with latest technology and tools, come in group of four or five, expert in every field on automobile engineering, they can find a way in [4]. They don’t require to copy the ignition key as they use some Chinese software “the Engine Control Module (ECM) code breaker”, available online and costs Rs 1 lakh. They can break steering lock and gear box lock with specialized tools and start the car. Here is the case study on automobile theft by Indian express [3]
Figure 1 Survey of Indian expresses about car theft and solved case

4. OBJECTIVE

To provide cheaper solution, we use open source hardware and software tools. Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects. The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

5. BLOCK DIAGRAM OF SYSTEM

5.1. Finger Print Module
R305 Fingerprint module is one of the best and cheap sensor. It has TTL serial mode for communication which makes it easier to interface with any microcontroller based system. It comes around 2200 rupees from robokit India. Detailed version of datasheet is given in reference [5]. Adafruit provide good information on module as well as instruction for interfacing with Arduino [6]. They also provide good library to interface fingerprint module with Arduino, the source code is available on GitHub [7].

6. IGNITION KEY DIAGRAM

Figure 3 Standard 4-wire electronics ignition system key locking

7. SCHEMATIC OF SYSTEM

Figure 4 Schematic diagram (designed with diptrace free edition for windows)
8. CIRCUIT BOARD LAYOUT

Figure 5 Circuit board layout (top view on left and bottom view on right) (designed with diptrace free edition for windows)

9. SOURCE CODE

```c
#include <Adafruit_Fingerprint.h>
#include <LiquidCrystal.h>

boolean toggle = false;
uint8_t p; char c; uint8_t d; uint8_t id; uint8_t
getFingerprintEnroll();

void setup() {
  pinMode(sw1, INPUT); //mode switch
  pinMode(sw2, INPUT); //ok switch
  pinMode(Bike_start, OUTPUT);
  pinMode(bat_start, OUTPUT);
  digitalWrite(Bike_start, LOW);
  digitalWrite(bat_start, LOW);
  Serial.begin(57600);
}

void loop() {
  mode_switch_stat==LOW; ok_switch_stat==LOW;
  delay(1000); lcd.clear();

  if(!finger.verifyPassword()) {
    lcd.clear();
    lcd.print("Did not find");
    lcd.setCursor(0, 1);
    delay(2000);
    lcd.print("Enrolling Master");
    lcd.setCursor(0, 1);
    lcd.print("Fingerprint");
    id=0;
    while (!getFingerprintEnroll() );
  }
}
```

```c
constint sw2=11; constint sw1=10; intmode_switch_stat=0;
constint Bike_start=8; constint bat_start=9;
constint RS=7; constint EN=6; constint D4=5;
constint D5=4; constint D6=3; constint D7=2;

void setup() {pinMode(sw1, INPUT); //mode switch pinMode(sw2, INPUT); //ok switch pinMode(Bike_start,OUTPUT);
pinMode(bat_start,OUTPUT);
digitalWrite(Bike_start,LOW);
digitalWrite(bat_start,LOW);
Serial.begin(57600);

lcd.begin(16, 2); lcd.setCursor(0, 0);
```

```c
lcd.print("fingerprint bike"); lcd.setCursor(0, 1);
lcd.print("unlocked system"); delay(1000); lcd.clear();
lcd.print(" Initializing"); delay(1000); finger.begin(57600);
```

```c
// set the data rate for the sensor serial port

if (!finger.verifyPassword())
  {
    lcd.clear();
    lcd.print("Did not find");
    lcd.setCursor(0, 1);
    delay(2000);
    lcd.print("Enrolling Master");
    lcd.setCursor(0, 1);
    lcd.print("Fingerprint");
    id=0;
    while (!getFingerprintEnroll() );
  }
}
void loop() {
  mode_switch_stat==LOW; ok_switch_stat==LOW;
  delay(1000); lcd.clear();

  lcd.print(" Waiting for ");
  lcd.setCursor(0, 1);
  lcd.print(" valid finger ");
  stat=255;
  if(stat==255) {
    delay(1000);
    lcd.clear();
    lcd.print("error!!try again");
  }
  else {
    mode_switch_stat=LOW;
    delay(100);
    mode_switch_stat=digitalRead(sw1);
    delay(1000);
    if(stat==0 && mode_switch_stat==HIGH) {
      //special mode open delay(1000)
      lcd.setCursor(0, 1);
      delay(100);
      mode_switch_stat=LOW;
    }
  }

  if(stat==0 & mode_switch_stat==HIGH) {
    //special mode open delay(1000)
    lcd.setCursor(0, 1);

    lcd.print(" 3.delete");
    delay(4000);
    lcd.clear();
    lcd.print("enter choice ");
    d=readnumber();
```
lcd.setCursor(0, 1); lcd.print(" "); lcd.print(d); switch (d){ case 1: delay(1000); lcd.clear(); lcd.print("Ready to enroll"); lcd.setCursor(0, 1);
lcd.print("a fingerprint!");delay(1000); lcd.clear(); lcd.print("Please Type the"); lcd.setCursor(0, 1);lcd.print("ID "); id = readnumber(); delay(1000); lcd.clear(); id = readnumber(); delay(1000); lcd.setCursor(0, 1); lcd.print("ID "); id = readnumber(); delay(1000); lcd.clear(); id = readnumber(); delay(1000); id = readnumber(); delay(1000); if(id==0) {
delay(1000); lcd.clear(); lcd.print("you cannot
delete"); lcd.setCursor(0, 1); lcd.print("Master id");
} else{
delay(1000); lcd.clear(); lcd.print("deleting ID ");
deleteFingerprint(id); }
else{
delay(1000); lcd.clear(); lcd.print("you cannot
delete"); lcd.setCursor(0, 1); lcd.print("Master id");
} else if(stat!=255) {
toggle = !toggle; digitalWrite(Bike_start,toggle); digitalWrite(bat_start,toggle); delay(1000);
if(toggle==1) lcd.print(" bike started"); else lcd.print(" bike stopped"); delay(5000); 
} else {
delay(1000); lcd.clear(); lcd.print("wrong entry try
gain!"); } } else if(stat!=255) {
delay(1000); lcd.clear(); lcd.print("Waiting for id "); lcd.print(id); lcd.setCursor(0, 1); lcd.print("valid"); while (p != FINGERPRINT_OK) { p = finger.getImage(); delay(1000); switch (p) {case FINGERPRINT_OK:
lcd.clear();lcd.print("Image taken"); break; case FINGERPRINT_NOFINGER:
lcd.print(".");break; case FINGERPRINT_PACKETRECIEVII 
RR:
lcd.clear();lcd.print("Communication err");break; case FINGERPRINT_IMAGEFAIL: lcd.clear(); lcd.print("Imaging error"); break; default:
lcd.clear();lcd.print("Unknown error");break; } // OK success!
p = finger.getImage(); while (p !=" FINGERPRINT_OK) { p = finger.getImage();
} uint8_t getFingerprintIDez() { p=255; } uint8_t readnumber(void) { mode_switch_stat=LOW; ok_switch_stat=LOW;
uint8_t num = 0;
while (ok_switch_stat==LOW) {
mode_switch_stat=HIGH; num++; while(mode_switch_stat!=LOW
&ok_switch_stat!=LOW) 
return num; }
uint8_t getFingerprintEnroll() { int p = -1;
delay(1000); lcd.clear(); lcd.print("Waiting for id ");
if(p==FINGERPRINT_OK) {p = finger.getImage(); delay(1000);
switch (p) {case FINGERPRINT_OK:
lcd.clear();lcd.print("Image taken"); break; case FINGERPRINT_NOFINGER:
lcd.print(".");break; case FINGERPRINT_PACKETRECIEVII 
RR:
lcd.clear();
switch (p) {case FINGERPRINT_OK:
lcd.clear();lcd.print("Image converted");break; case FINGERPRINT_IMAGEFAIL: lcd.clear();
}
} // returns -1 if failed, otherwise returns ID 
// found a match!
return finger.fingerID; p=255; }
}
case FINGERPRINT_PACKETRECIEVEE RR:
lcd.clear();lcd.print("Communication err");
return p;
case FINGERPRINT_FEATUREFAIL:
lcd.clear();lcd.print("Could not find ");
lcd.setCursor(0, 1);lcd.print("fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
lcd.clear();lcd.print("Could not find ");
lcd.setCursor(0, 1);lcd.print("fingerprint features");
return p;
default: lcd.clear();lcd.print("Unknown error"); return p;
}
delay(1000);lcd.clear();
lcd.print("Remove finger"); delay(1000);p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage(); delay(1000);
}
delay(1000);lcd.clear();
lcd.print("ID ");lcd.print(id); p = -1;
delay(1000);lcd.clear();
lcd.print("Place same "); lcd.setCursor(0, 1);
while (p != FINGERPRINT_OK) { p = finger.getImage(); delay(1000);
    switch (p) {case FINGERPRINT_OK: lcd.clear();
lcd.print("Image taken"); break;
    case FINGERPRINT_NOFINGER: lcd.print("."); break;
    case FINGERPRINT_PACKETRECIEVEE RR:
    lcd.clear();
lcd.print("Communication err"); break;
    case FINGERPRINT_IMAGEFAIL: lcd.clear();
lcd.print("Imaging error"); break;
    default: lcd.clear();
lcd.print("Unknown error");break; }
}
// OK converted!
delay(1000);lcd.clear();
p = finger.createModel(); delay(1000);lcd.clear();
if (p == FINGERPRINT_OK) { lcd.print("Prints matched!"); }
else if (p == FINGERPRINT_PACKETRECIEVEE RR) {lcd.print("Communication err"); return p;}
elif (p == FINGERPRINT_ENROLLMISMATCH) {lcd.print("Fingerprints ");
    lcd.setCursor(0, 1);
    lcd.print("did not match");return p; else {lcd.print("Unknown error");
    return p; return p;}
}
}
uploadFingerprintTemplate(uint16_t id)
{uint8_t p = finger.loadModel(id);
delay(1000); lcd.clear(); switch (p) {
case FINGERPRINT_OK:
lcd.print("fingerprint "); lcd.print(id);
}
lcd.setCursor(0, 1);lcd.print("available");
break;
case FINGERPRINT_PACKETRECIEVEE
RR: lcd.print("Communication err");
return p;
default: lcd.print("fingerprint");
lcd.print(id); lcd.setCursor(0, 1);
lcd.print(" not available");return p;
} // OK success!

10. FUTURE DEVELOPMENT

We can install GPS with GSM for vehicle tracking and to send alert on repetitive unauthorized unlock request. This helps us to completely lock down automobile and immediate tracking.

We can install remote control automobile controller by which we can control security features like remote ignition locking, remote steering locking, remote gear box locking, remote door looking, or remote alarming system.

11. CONCLUSION

We can design our own security system with our own unique hardware, which cannot be crack by thief easily. Car unlocking method is only valid fingerprint which is unique for each user. Designing hardware is very easy because it is based on open source hardware, which provide easy to design hardware solution which remove the hassle of study of internals of controller. Software code is open source having so many API for control purpose which helps to develop our source code with ease. Taking the help of Adafruit library and open source Arduino IDE we can develop source code without hassle of internal architecture of Fingerprint Module.

REFERENCE


