```
const int knockSensor = 0;
const int programSwitch = 2;
const int lockMotor = 3;
const int redLED = 4;
const int greenLED = 5;

const int threshold = 3;
const int rejectValue = 25;
const int averageRejectValue = 15;
const int knockFadeTime = 150;
const int lockTurnTime = 2000;

const int maximumKnocks = 20;
const int knockComplete = 1200;

int secretCode[maximumKnocks] = {50, 25, 25, 50, 100, 50, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
int knockReadings[maximumKnocks];
int knockSensorValue = 0;
int programButtonPressed = false;

void setup() {
  pinMode(lockMotor, OUTPUT);
  pinMode(redLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(programSwitch, INPUT);

  Serial.begin(9600);
  Serial.println("Program start.");

  digitalWrite(greenLED, HIGH);
}
void loop() {
  knockSensorValue = analogRead(knockSensor);

  if (digitalRead(programSwitch)==HIGH){
    programButtonPressed = true;
    digitalWrite(redLED, HIGH);
  } else {
    programButtonPressed = false;
    digitalWrite(redLED, LOW);
  }

  if (knockSensorValue >=threshold){
    listenToSecretKnock();
  }
}
void listenToSecretKnock(){
  Serial.println("knock starting");

  int i = 0;
  for (i=0;i<maximumKnocks;i++){
    knockReadings[i]=0;
  }
  int currentKnockNumber=0;
  int startTime=millis();
  int now=millis();

  digitalWrite(greenLED, LOW);
  if (programButtonPressed==true){
     digitalWrite(redLED, LOW);
  }
  delay(knockFadeTime);
  digitalWrite(greenLED, HIGH);
  if (programButtonPressed==true){
     digitalWrite(redLED, HIGH);
  }
  do {
    knockSensorValue = analogRead(knockSensor);
    if (knockSensorValue >=threshold){
      Serial.println("knock.");
      now=millis();
      knockReadings[currentKnockNumber] = now-startTime;
      currentKnockNumber ++;
      startTime=now;
      digitalWrite(greenLED, LOW);
      if (programButtonPressed==true){
        digitalWrite(redLED, LOW);
      }
      delay(knockFadeTime);
      digitalWrite(greenLED, HIGH);
      if (programButtonPressed==true){
        digitalWrite(redLED, HIGH);
      }
    }
    now=millis();
  } while ((now-startTime < knockComplete) && (currentKnockNumber < maximumKnocks));
  if (programButtonPressed==false){
    if (validateKnock() == true){
      triggerDoorUnlock();
    } else {
      Serial.println("Secret knock failed.");
      digitalWrite(greenLED, LOW);
      for (i=0;i<4;i++){
        digitalWrite(redLED, HIGH);
        delay(100);
        digitalWrite(redLED, LOW);
        delay(100);
      }
      digitalWrite(greenLED, HIGH);
    }
  } else {
    validateKnock();
    Serial.println("New lock stored.");
    digitalWrite(redLED, LOW);
    digitalWrite(greenLED, HIGH);
    for (i=0;i<3;i++){
      delay(100);
      digitalWrite(redLED, HIGH);
      digitalWrite(greenLED, LOW);
      delay(100);
      digitalWrite(redLED, LOW);
      digitalWrite(greenLED, HIGH);
    }
  }
}


void triggerDoorUnlock(){
  Serial.println("Door unlocked!");
  int i=0;

  digitalWrite(lockMotor, HIGH);
  digitalWrite(greenLED, HIGH);

  delay (lockTurnTime);

  digitalWrite(lockMotor, LOW);

  for (i=0; i < 5; i++){
      digitalWrite(greenLED, LOW);
      delay(100);
      digitalWrite(greenLED, HIGH);
      delay(100);
  }
}
boolean validateKnock(){
  int i=0;

  int currentKnockCount = 0;
  int secretKnockCount = 0;
  int maxKnockInterval = 0;

  for (i=0;i<maximumKnocks;i++){
    if (knockReadings[i] > 0){
      currentKnockCount++;
    }
    if (secretCode[i] > 0){
      secretKnockCount++;
    }

    if (knockReadings[i] > maxKnockInterval){
      maxKnockInterval = knockReadings[i];
    }
  }

  if (programButtonPressed==true){
      for (i=0;i<maximumKnocks;i++){
        secretCode[i]= map(knockReadings[i],0, maxKnockInterval, 0, 100);
      }
      digitalWrite(greenLED, LOW);
      digitalWrite(redLED, LOW);
      delay(1000);
      digitalWrite(greenLED, HIGH);
      digitalWrite(redLED, HIGH);
      delay(50);
      for (i = 0; i < maximumKnocks ; i++){
        digitalWrite(greenLED, LOW);
        digitalWrite(redLED, LOW);
        if (secretCode[i] > 0){
          delay( map(secretCode[i],0, 100, 0, maxKnockInterval));
          digitalWrite(greenLED, HIGH);
          digitalWrite(redLED, HIGH);
        }
        delay(50);
      }
    return false;
  }

  if (currentKnockCount != secretKnockCount){
    return false;
  }
  int totaltimeDifferences=0;
  int timeDiff=0;
  for (i=0;i<maximumKnocks;i++){
    knockReadings[i]= map(knockReadings[i],0, maxKnockInterval, 0, 100);
    timeDiff = abs(knockReadings[i]-secretCode[i]);
    if (timeDiff > rejectValue){
      return false;
    }
    totaltimeDifferences += timeDiff;
  }
  if (totaltimeDifferences/secretKnockCount>averageRejectValue){
    return false;
  }

  return true;

}
```