॥ सा विद्या या विमुक्तये ॥

भारतीय प्रौद्योगिकी संस्थान धारवाड़

Indian Institute of Technology Dharwad

Electronic Design Lab

Objective of the Project

Traffic Pattern Analysis of IoT devices and malicious users detection

In this project , we have to build a set-up consisting of at least three IoT devices, which use WiFi for transmission and a receiver to process these transmissions. One way to do this would be to use an Arduino Uno board with WiFi transmitter module to emulate an IoT device. Similarly, the receiver can also be built using another Arduino Uno board with WiFi receiver module. Each IoT device transmits periodically (say every 3 sec, 7 sec and 10 sec). The receiver should be able to detect the number of IoT devices and their periodicity using energy-based and/or correlation-based methods. Further, a malicious IoT node will also be emulated, whose transmission pattern is not periodic (or has some different periodicity than the legitimate devices). The receiver should detect the presence of this malicious device and alert the legitimate devices.

Work Progress

- Familiarize with arduino and esp8266 module and try to connect them.
- Write a working code for transmitting and receiving signals from 2 different arduino.
- We are currently using Thingspeak for it.
- We are using 2 Arduino UNO, esp8266 wifi module, Esp8266 NodeMCU, jumper wires and a breadboard.

Code for transmitting signals

```
void setup() {
  Serial.begin(9600);
  esp8266.begin(115200);
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");
}

void loop() {
 String getData = "GET /update?api_key="+ API +"&"+ field +"="+String(valSensor);
sendCommand("AT+CIPMUX=1",5,"OK");
 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
 sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
 esp8266.println(getData);delay(100);countTrueCommand++;
 sendCommand("AT+CIPCLOSE=0",5,"OK");
 Serial.print("this is val");
 Serial.println(valSensor);
 Serial.print("this is +1");
 Serial.println(valSensor++);
 valSensor++;

}
```

Code for receiving signals

```
void setup() {
  Serial.begin(115200);
  Serial.println("Start");
  connectWiFi();
  aConst = readTSData(channelID,dataFieldOne);
  delay(100);
}

void loop() {
  delay(2000);
  Serial.println("Wait");
  aConst = readTSData(channelID,dataFieldOne);
  //Serial.println(aConst);
  Serial.println("Waiting...");
}

int connectWiFi() {
  WiFi.begin(ssid, password );
  while (WiFi.status() != WL_CONNECTED){
  delay(500);
  Serial.print("*");
  Serial.println("Connected");
  ThingSpeak.begin( client );
}
}
float readTSData (long TSChannel, unsigned int TSField){
  float data= ThingSpeak.readFloatField (TSChannel, TSField, readAPIKey);
  Serial.println( "DATA" + String(data,9));
  return data;
}
```

Distribution of work throughout the Lab

Evaluation Number	To-Do	Time Taken (Approx)
1	–Familiarize with Arduino and ESP8266 module and connect them -Write a working code for transmitting signal from the 1st arduino and receiving signal from 2nd arduino	2 Weeks 1 Week
2	-Write a working code to detect and receive the 2 incoming signals in the receiver arduino	3 Weeks
3	-Try adding more transmitting signals from 1st arduino or use a new arduino to transmit signal -Modify the code and try to distinguish the incoming signals	1 Week 2 Week
4	-Use one more arduino to generate aperiodic signals and try to distinguish it by their time period	3 Weeks

॥ सा विद्या या विमुक्तये ॥

भारतीय प्रौद्योगिकी संस्थान धारवाड़

Indian Institute of Technology Dharwad

Electronic Design Lab

Second Evaluation

Code to get MAC address

```
#include <WiFi.h>

void setup(){
  Serial.begin(115200);
  Serial.println();

  Serial.println(WiFi.macAddress());
}
void loop(){
}
```

Code for Transmitting Periodic Signal

```
void loop() {
  // Set values to send
  myData.id = 1;
  strcpy(myData.s,"test");
  Serial.println(myData.s);
//  myData.y = random(0,50);

  // Send message via ESP-NOW
  esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));

  if (result == ESP_OK) {
    Serial.println("Sent with success");
  }
  else {
    Serial.println("Error sending the data");
  }
  delay(10000);
}
```

Code for Receiving Signal

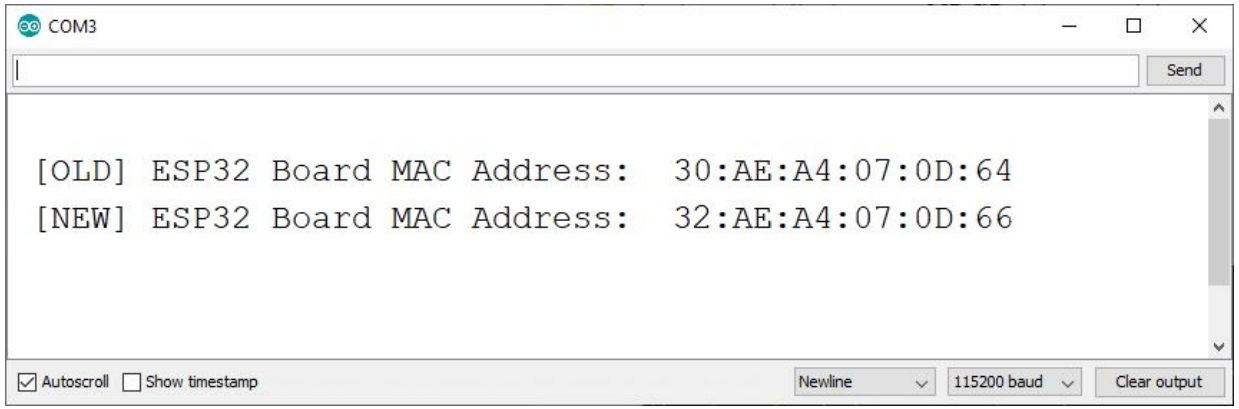
```
void OnDataRecv(const uint8_t * mac_addr, const uint8_t *incomingData, int len) {
  char macStr[18];
  Serial.print("Packet received from: ");
  snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
           mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4], mac_addr[5]);
  Serial.println(macStr);
  memcpy(&myData, incomingData, sizeof(myData));
  Serial.printf("Board ID %u: %u bytes\n", myData.id, len);
  // Update the structures with the new incoming data
  strcpy(boardsStruct[myData.id-1].s, myData.s);

 int starttime=millis();
  Serial.printf("x value: %s \n", boardsStruct[myData.id-1].s);
  //Serial.printf("y value: %d \n", boardsStruct[myData.id-1].y);
  int endtime=millis();
  Serial.printf("time difference is %d",endtime-starttime );
  Serial.println();
}
```

Work Progress

- Currently we are able to send signals locally without the use of ThingSpeak
- We are sending a periodic signal with a delay of 10s and one aperiodic signal with random delays.
- The periodic signal sends a text message - “Test” while the aperiodic signal sends a text message - “Test1”.
- Here we used the MAC Address of the receiving ESP32 to send the message to the correct ESP module.

ESP32


COM3
Send
[OLD] ESP32 Board MAC Address: 30:AE:A4:07:0D:64
[NEW] ESP32 Board MAC Address: 32:AE:A4:07:0D:66
Autoscroll
Show timestamp
Newline
115200 baud
Clear output

Future Plans

- Since we are able to send both periodic and aperiodic signals now, we will try to add one more periodic signal from ESP32 and try to disable the signals that are being received by the aperiodic signals after a couple of times.
- We will try to differentiate the signals based on periodicity and disable the odd one out.

॥ सा विद्या या विमुक्तये ॥

भारतीय प्रौद्योगिकी संस्थान धारवाड़

Indian Institute of Technology Dharwad

Electronic Design Lab

Third Evaluation

Work Progress

- We tried to used RSSI (Received Signal Strength Indicator) to distinguish between the periodic and aperiodic.(due to some errors we dropped it)
- Currently we are using Mac addresses as the unique id for different transmitting nodes.
- So now we are able to distinguish between periodic and aperiodic signals
- And accept data only from the periodic transmitters.

Code we used for RSSI

```
// Serial.println("scan start");


  // WiFi.scanNetworks will return the number of networks found
//  int n = WiFi.scanNetworks();
//  Serial.println("scan done");
//  if (n == 0) {
//      Serial.println("no networks found");
//  } else {
//    Serial.print(n);
//    Serial.println(" networks found");
//    for (int i = 0; i < n; ++i) {
//      // Print SSID and RSSI for each network found
```

```
//      Serial.print(i + 1);
//      Serial.print(": ");
//      Serial.print(WiFi.SSID(i));
//      Serial.print(" (");
//      Serial.print(WiFi.RSSI(i));
//      Serial.print(")");
//      Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)?" ":"*");
//      delay(10);
//}
//  }
  delay(60000);
```

Code using Mac address

```
if (mac_addr[0] == 0x24 &&  mac_addr[1] == 0xD7 &&  mac_addr[2] == 0xEB &&  mac_addr[3] == 0x0E &&  mac_addr[4] == 0xD3 &&  mac_addr[5] == 0x50 ) {
   if(boardsStruct[0].flag == 1 ){
    memcpy(&myData, incomingData, sizeof(myData));

    Serial.printf("Board ID %u: %u bytes\n", myData.id, len);
    // Update the structures with the new incoming data
    strcpy(boardsStruct[0].s, myData.s);

    Serial.printf("x value: %s \n", boardsStruct[0].s);
   }
  }

   if ((mac_addr[0] == 0x10 &&  mac_addr[1] == 0x52 &&  mac_addr[2] == 0x1C &&  mac_addr[3] == 0x75 &&  mac_addr[4] == 0xF5 &&  mac_addr[5] == 0xE0)) {
     if(boardsStruct[1].flag == 1) {
```

```
memcpy(&myData, incomingData, sizeof(myData));

    Serial.printf("Board ID %u: %u bytes\n", myData.id, len);
    // Update the structures with the new incoming data
    strcpy(boardsStruct[1].s, myData.s);

    Serial.printf("x value: %s \n", boardsStruct[1].s);
    }
  }

  if ((mac_addr[0] == 0x10 &&  mac_addr[1] == 0x52 &&  mac_addr[2] == 0x1C &&  mac_addr[3] == 0x75 &&  mac_addr[4] == 0xF5 &&
mac_addr[5] == 0xE0) && j < 10) {
    counter2[j] = millis();
   // Serial.printf("successful from board2 c2[j]= %d\n", counter2[j]);
    j++;
  }

  if ( (mac_addr[0] == 0x24 &&  mac_addr[1] == 0xD7 &&  mac_addr[2] == 0xEB &&  mac_addr[3] == 0x0E &&  mac_addr[4] == 0xD3 &&
mac_addr[5] == 0x50 ) && i < 10) {
    counter1[i] = millis();
   // Serial.printf("successful from board1 c1[i]= %d\n" , counter1[i]);
    i++;

  }
```

```
//to check counter arrays as a whole

//  if(i==10){
//    for(int pos=0;pos<10;pos++){
//      Serial.printf("counter 1 values are : %d/n",counter1[pos]);
//    }
//  }
//
  if(j==10){
    for(int pos=0;pos<10;pos++){
      Serial.printf("counter 2 values are : %d/n",counter2[pos]);
    }
  }

Serial.printf("i value is %d, j value is %d\n",i,j);
  if (i == 10) {

    for (int k = 1; k < 9; k++) {
      Serial.printf("Board 1 Value is %d\n", counter1[k] - counter1[k - 1]);
      if (abs((counter1[k + 1] - counter1[k])- (counter1[k] - counter1[k - 1])) > 50) {
        flag1 = false;
       break;
      }
      else {
        flag1 = true;
      }
    }
i=0;
}
```

```
if (j == 10) {

  for (int k = 1; k < 9; k++) {
   Serial.printf("Board 2 Value is %d\n", counter2[k] - counter2[k - 1]);
    if (abs((counter2[k + 1] - counter2[k]) - (counter2[k] - counter2[k - 1]) ) > 50) {
      flag2 = false;
      break;
    }
    else {
     flag2 = true;
    }

  }
  j=0;
}
if (flag1==true){
  boardsStruct[0].flag=true;
  }
  if (flag2==true){
  boardsStruct[1].flag=true;
  }
  if (flag1==false){
  boardsStruct[0].flag=false;
  }
  if (flag2==false){
  boardsStruct[1].flag=false;
  }
        Serial.printf("flag1%d and flag2%d\n",flag1,flag2);

}
```

```
void setup() {
  //Initialize Serial Monitor
  Serial.begin(115200);

  //Set device as a Wi-Fi Station
  WiFi.mode(WIFI_STA);

  //Init ESP-NOW
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }

  // Once ESPNow is successfully Init, we will register for recv CB to
  // get recv packer info
  esp_now_register_recv_cb(OnDataRecv);
}
```

Future Plans

- We will try more security breaching schemes and will try to eliminate the malicious transmitter.
- We will try to print data(or output) directly to spreadsheet
- We will add more transmitter node.

॥ सा विद्या या विमुक्तये ॥

भारतीय प्रौद्योगिकी संस्थान धारवाड़

Indian Institute of Technology Dharwad

Electronic Design Lab

Fourth Evaluation

Data Hack

- This time we incorporated data size too as a important factor in determining the malicious transmitter.
- So , we will set a data limit (basically size limit on the amount of data coming at the receiver end)
- Depending on the size limit we will determine whether the transmitter is malicious or not.
- The one which crosses the limit size will be pointed out .
- And considering the initial periodicity of each transmitter.

Related Code

```
memcpy(&myData, incomingData, sizeof(myData));
      if (4 <= strlen(myData.s) <= 20) {
        strcpy(boardsStruct[0].s, myData.s);


        Serial.printf("x value: %s \n", boardsStruct[0].s);
      }
       else{
          Serial.printf("Virus Detected");
          }
      boardsStruct[0].flag = false;
    }
    else {
      for (int item = 1; item < 5; item++) {
        if (abs((boardsStruct[0].counter[item] - boardsStruct[0].counter[item - 1]) - (counter1[item] - counter1[item - 1])) > 50) {
```

Code

```
boardsStruct[0].flag = 0;
          Serial.printf("Virus detected");
          break;
        }
      }
      if  (boardsStruct[0].flag == 1) {
        memcpy(&myData, incomingData, sizeof(myData));

        if (4 <= strlen(myData.s) <= 20) {
          strcpy(boardsStruct[0].s, myData.s);
//          Serial.printf("x value: %s \n", boardsStruct[0].s);
          boardsStruct[0].flag = false;
        }
        else{
//          Serial.printf("Virus Detected");
          }
```