**ENGR 498**

Jenifer Delgado Rodriguez

Swapnil Modani

Randa Shady

Group Progress Report due 16th June 2019

**Project Overview**

Our target is to build an IoT-based user programmable home automation system.

One model will have temperature, motion, sound and a smoke sensor/detector controlled with multiple ESP8266-NodeMCU units. The other model will have a camera and it will be controlled through a Raspberry Pi 3B+. The data collected by the sensors is to be encrypted and sent to a server (or Cloud) where it will become accessible for the user from anywhere in the world through a website. The camera will also be accessible anytime from anywhere live.

The users are given the control to turn on/off their home-devices based on the readings from the sensors. The user will decide how the devices will operate based on their preferences(if-then-that.) For example, if the temperature in the room where the device is operating is more than 21 (degrees Celsius) turn on the AC. Or, if the user wishes to turn on the vacation mode, he/she will get instantly notified if movement is detected in his/her house and he/she would have live access to the camera as far as there is motion and up to 5 minutes after the motion ends.

**Goals from last week**

* Sensor integration with the ESP8266-NodeMCU.
* Localhost setup.
* Cloud communication setup.
* Motion detection with the camera.

**Accomplishments/Discoveries from last week**

* The sensors were successfully integrated to the ESP8266-NodeMCU separately.
* Since NodeMCU has only a single analog pin but we have 3 sensors that need analog input. To counter this, we found a piece of hardware (ADS1115 ADC) to add additional analog ports to the ESP8266-NodeMCU.
* The code for each sensor was written separately and tested. After all the individual testing, we integrated all the sensors together and they worked perfectly.
* We managed to send the sensor data collected on the ESP8266-NodeMCU to the Google Firebase server (cloud).
* We read about cryptography protocols that are used in cybersecurity. We ran codes for them on several domains including C, C++, JavaScript.

**Current project state**

The circuit is built with all the sensors (motion, sound, smoke and temperature), the microcontroller (ESP8266-NodeMCU) the relay and the ADS1115 ADC. All sensors are working successfully together. Data has been successfully sent using a local host from the microcontroller to a phone using the ESP8266 Web Server. The data has also successfully been sent to the Google Firebase (Cloud.) All data from the sensors is successfully displayed and gets updated live. For example, when heat is encountered on the temperature sensor, an increase in temperature can be noticed in the received data at both cloud and localhost platforms. Also, we used OpenCV is used to make the web camera detect motion in the house for security purposes; the camera starts to record a video whenever motion is detected and it saves it. Regarding cryptography, the code for SHA256, AES, Triple DES works perfectly on ESP8266 which guarantees integrity, but some modifications are needed to get the confidentiality(encryption) part of the system to work. We are working on HTML5, CSS3 and JavaScript simultaneously for the front-end website development of the website for the user interface. This is a long process and hence it goes parallelly with the rest of the work.

**Goals for this week**

1. Circuit integration with all sensors to the ESP8266-NodeMCU and the ADS1115 ADC:

We are working with 4 major sensors, i.e. PIR motion sensor, temperature sensor, sound sensor and the smoke sensor. The sensors have to be integrated with the microcontroller (in this case ESP8266 NodeMCU) the ADS1115 ADC and the Relay. The ADS1115 ADC needs to be integrated in order to extend the number of analog ports available. The code (Arduino) for each sensor needs to be written and checked jointly. Once all sensor codes are joint, we will test for the response of the circuit.

2. Camera integration with Raspberry-Pi:

To start testing, we will need to learn some basic image processing (OpenCV Python,) write the code to be able to save images, save videos and life stream from the built-in camera of a laptop. Then find/write/implement a code that detects motion and it is effective. The technique used in the written code is based on frames differences; the difference between the consecutive frames is observed to detect motion if the difference exceeds the threshold. Only while motion is detected in the room, the camera should be recording a video, and pictures should be taken if the user wants to. If the room is unoccupied (motion is not detected,) the livestream video should be accessible for the user. This feature could be used for safety purposes while vacation mode is ON.

2. Local Host Setup:

Before we begin moving towards cloud communication, we want to implement the camera system to work in a local host. Same conditions as explained in the previous step will be tested with local host. There must be a live video that gets sent to the localhost and also the sensor data should be received that gets updated every time the page gets refreshed. To implement this, we are going with Flask (micro web framework)

3.Database management

Database management has to be over the Firebase Realtime database that is hosted on the cloud. We implement real-time database viewing and access. Instead of typical HTTP requests, the Firebase Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

4.Data Encryption

The goal is to gather the data from the sensors, encrypt it and then send it to the cloud where it will be decrypted again. We aim at applying the CIA system (Confidentiality Integrity Availability) in our project. We learnt about different cryptographic protocols including: zero trust networks, rot13, encoding and decoding base 64, RSA, AES, DES, triple DES, SHA256, and SHA256 HMAC. The code used for SHA256 works perfectly, however we’re having some difficulties in implementing the code for SHA256 HMAC and AES on ESP8266. However, the code found for triple DES seems to be working fine, but further testing is still needed. We also learnt about security based on hardware like the TPM module, but we thought that understanding how the code works will help us learn more.

5.Website

In order to have accessible the user-friendly website and achieve to build a user programmable home automation, we have to learn: HTML, CSS and JavaScript. The website has to be implemented with user-friendly interface. There must be a live feed of the camera setup for safety. Along with this it must have the interface where the user can set on/off any device of easily implement the IFTTT (If this then that). To do this and integrate it with the Google Firebase requires to use the Firebase API that needs to be coded in JavaScript. This allows us to integrate the website with the Firebase Cloud Realtime Database.