COEN 243: Internet of Things
Department of Computer Engineering
Santa Clara University

Remote IoT Weather Sensor Station

Colton Powell, Matthew Salmanpour, Justin Visas, Yezhou Zhao

Advisor
Behnam Dezfouli

June 2018

Introduction

Introduction

Problem Statement

- Intelligently automating functions in smart home, agricultural and horticulture applications such as turning on sprinklers, enabling HVAC, and adjusting blind position would be very valuable, but the sensor data needed to make these decisions is not always readily available

Significance

- Americans waste $130B of energy per year in homes and businesses
- Heating and cooling account for 48% of energy consumptions in homes
- 60% of water dumped/diverted for irrigation is wasted

Existing Solutions and Companies Working in This Domain

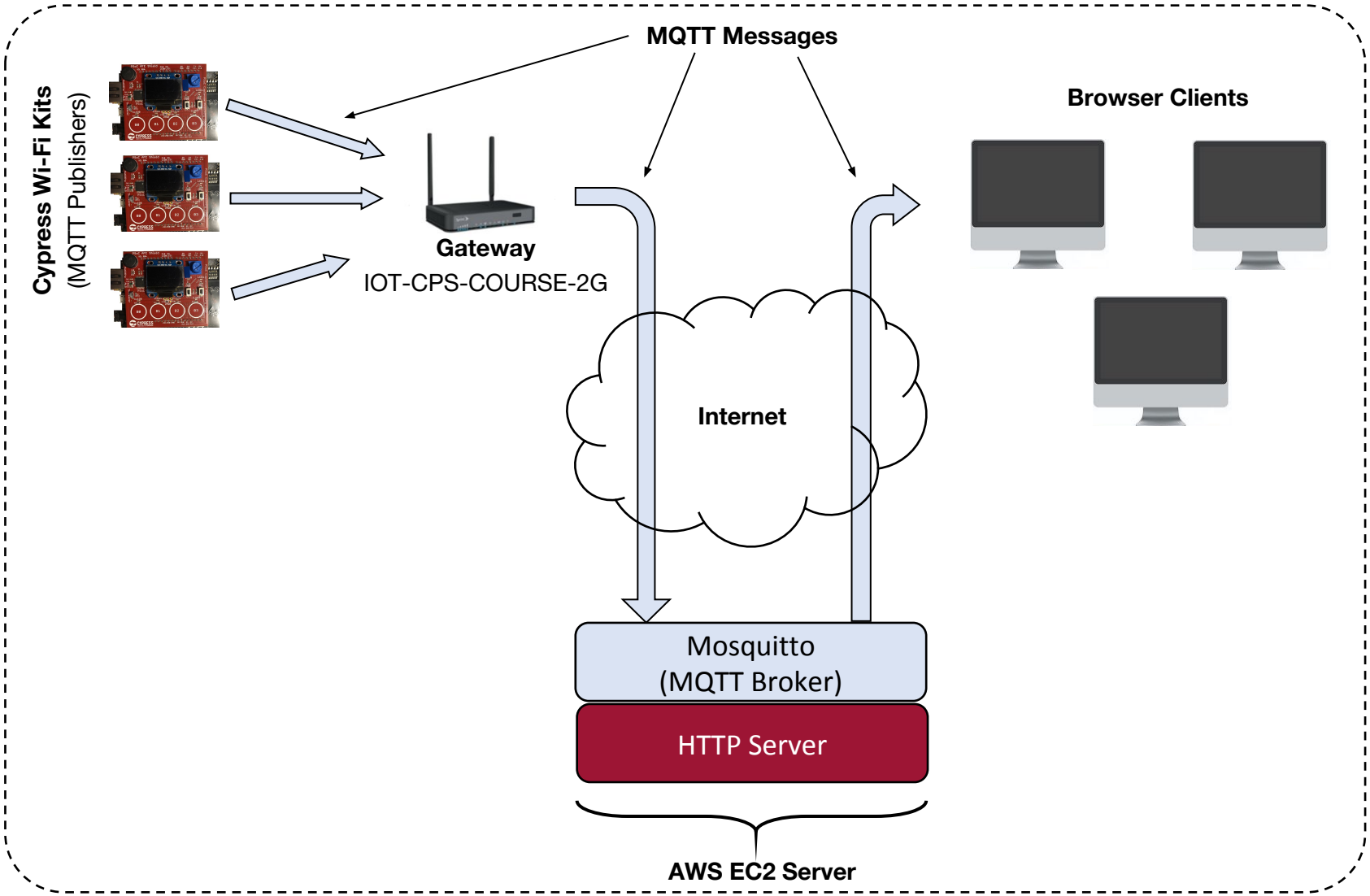
- Nest Wi-Fi Thermostat
- Wiser Air Wi-Fi Thermostat
- Smart Wi-Fi Sprinkler

Remote IoT Weather Sensor Station

Proposed Solution

Remote IoT Weather Sensor Station

Complete System Diagram -

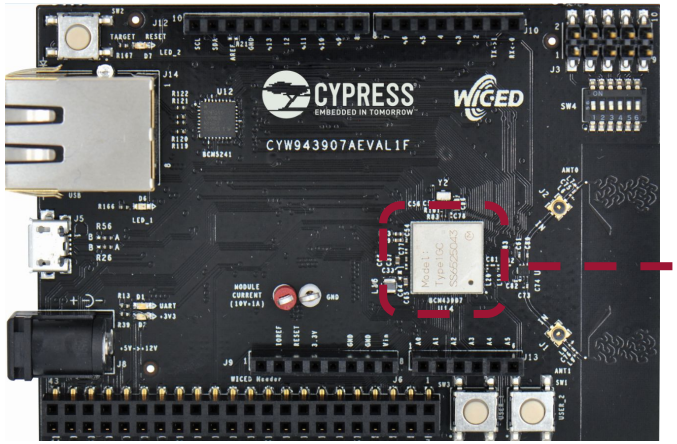


Proposed Solution

Remote IoT Weather Sensor Station

System Diagram – Hardware Overview

Cypress WICED® Wi-Fi CYW43907 Eval Kit

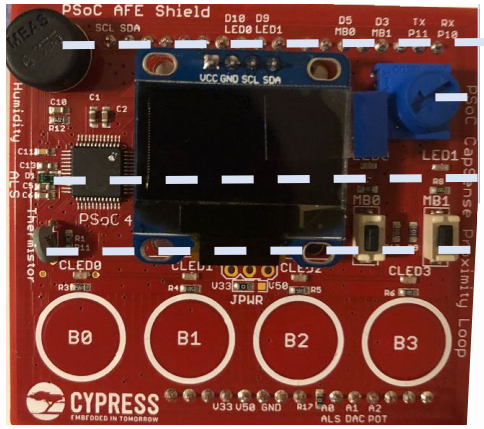


Murata 1GC Wi-Fi Module
Hosting Cypress' CYW43907 802.11b/g/n Wi-Fi SoC

Data Struct where data sent over I2C Messages from AFE Shield to Cypress Wi-Fi Board:

```
{
    float humidity;
    float temperature;
    float potentiometer_position
    float luminosity;              }
```

PSoC® AFE Shield



Humidity Sensor

Potentiometer

Light Sensor

Temperature Sensor

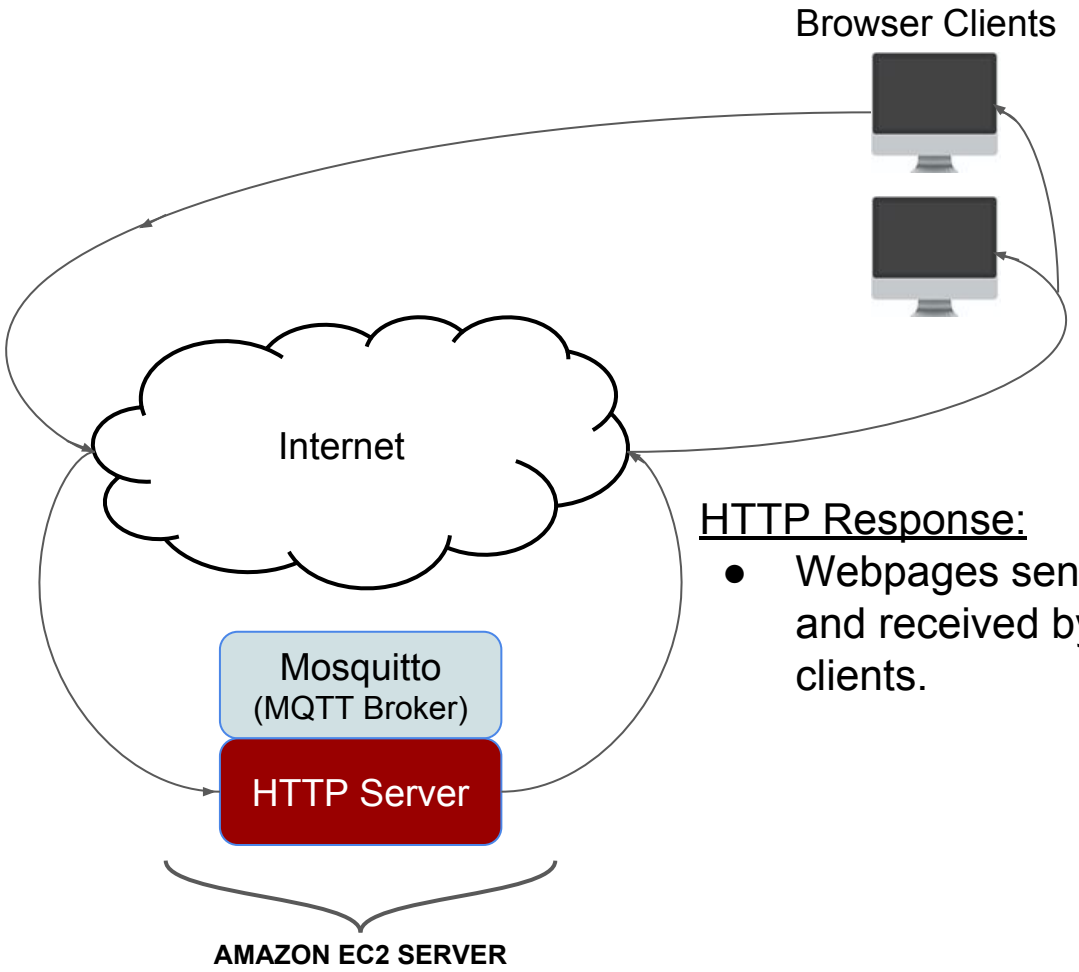
Proposed Solution

Remote IoT Weather Sensor Station

System Diagram - Web Page Subscriber

HTTP Requests:

- Browser requests web page, allowing it to act as MQTT Subscriber.
- Now, browser is allowed to receive weather data and draw graphs accordingly.



HTTP Response:

- Webpages sent to and received by clients.

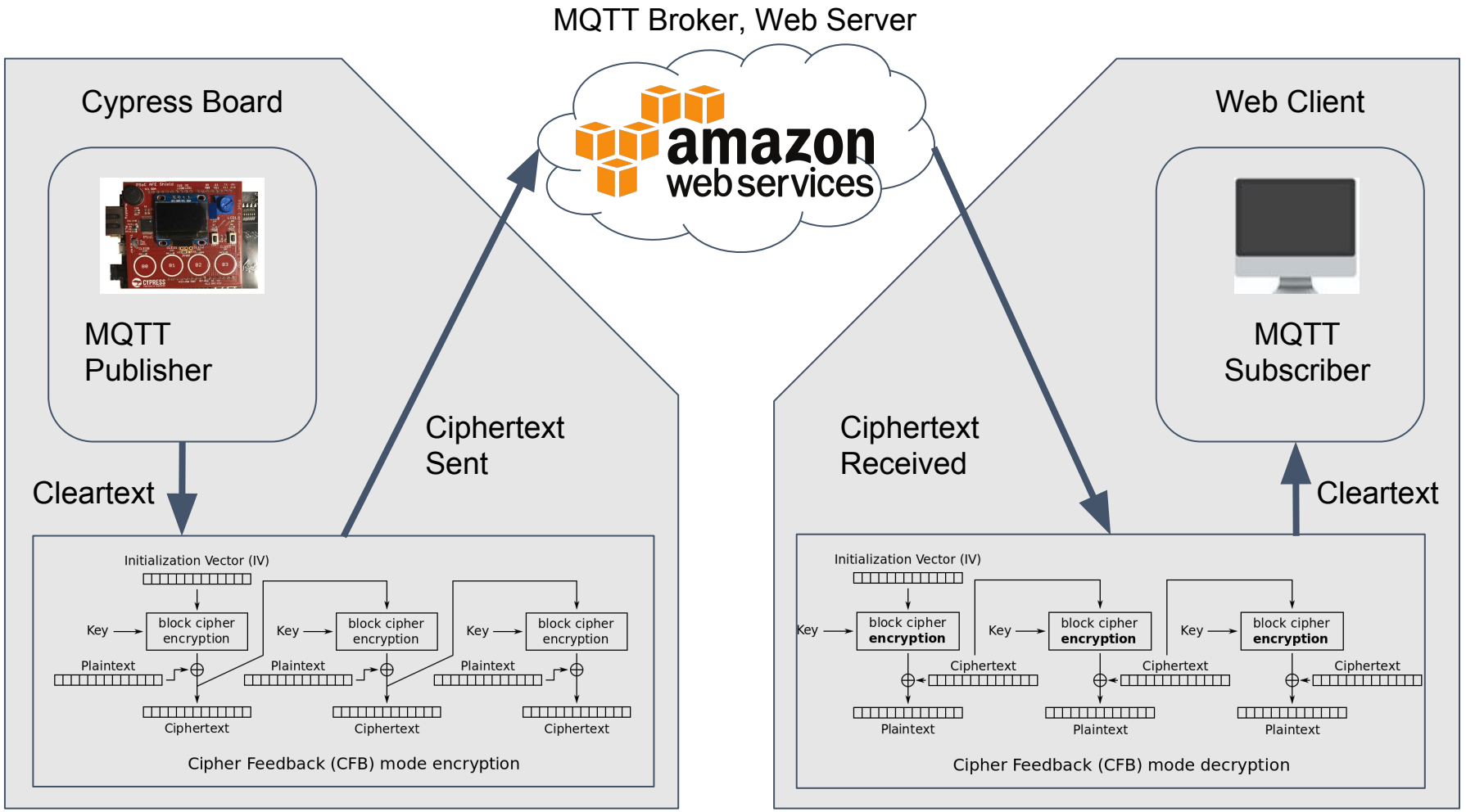
MQTT Subscriber Browser Client

- Displays charts containing information about the weather station: temperature, humidity, luminosity, and potentiometer readings
  - There are separate charts for each of these statistics
  - 4 Charts per weather station
- Dynamically creates charts for new devices; destroys charts for disconnected devices
- Updates every time an MQTT message is received
- Created using HTML + Javascript.
- Javascript Libraries used:
  - Paho MQTT: Allows the webpage to function as a MQTT client (subscriber in this case)
  - Chart.js: Used to draw line charts containing weather station information
  - Aes-js: For security

Proposed Solution

Remote IoT Weather Sensor Station

System Diagram – Encryption



Evaluation

Evaluation

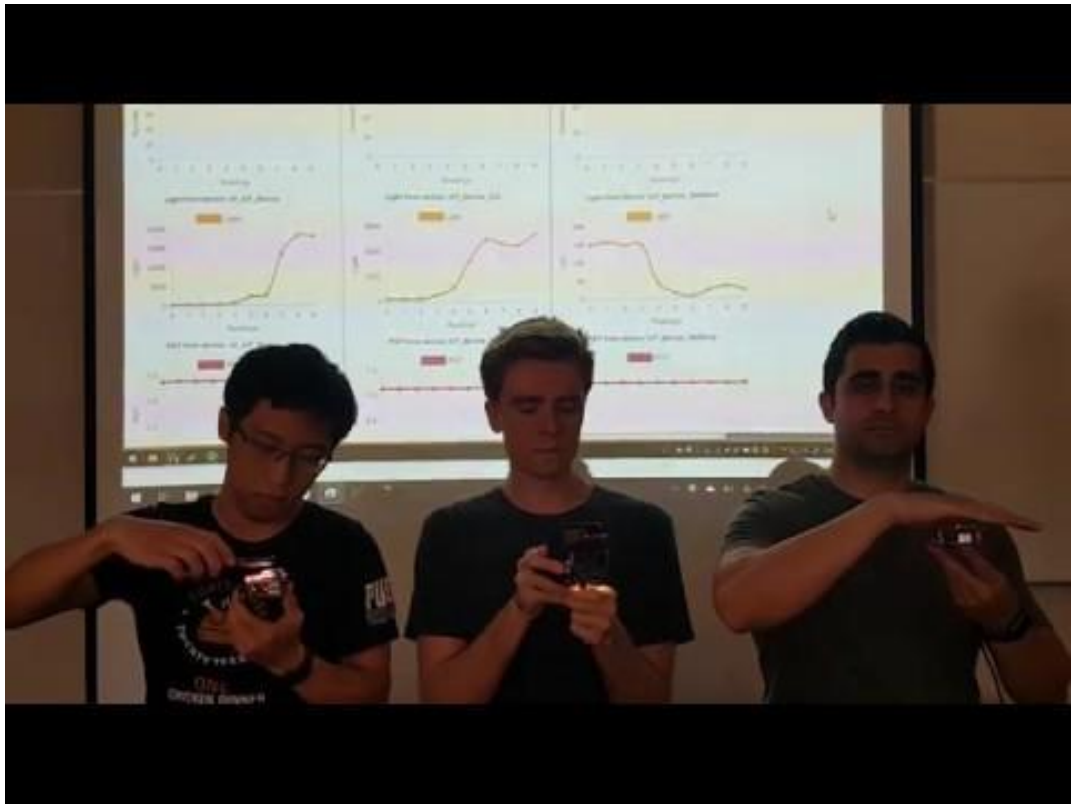
Remote IoT Weather Sensor Station

- System evaluation to ensure valid sensor data and system functionality:
  1) Connection Test:
     - Adding/removing Wi-Fi kits to the network to test dynamic expansion of sensor graphs on browser clients
  2) Humidity Test:
     - Breathing on humidity sensor on board to see changes in data
  3) Temperature Test:
     - Breathing on temperature sensor on board to see changes in data
  4) Luminosity Test:
     - Covering and shining light on light sensor on board and checking for the appropriate actions in the data
  5) Potentiometer Test:
     - Adjusting position of potentiometer to see changes in data

Remote IoT Weather Sensor Station Demo Video on next slide

Evaluation

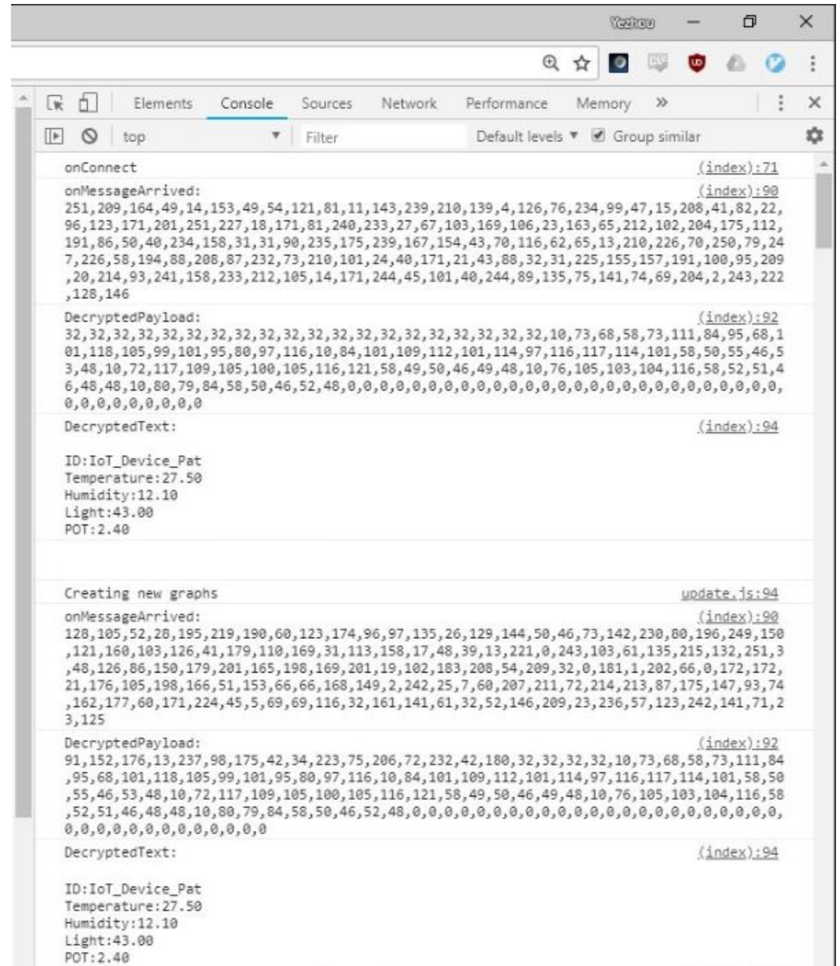
Video Demo



Encryption Results

Cypress Board Serial Out (MQTT Publisher)

Web Browser Console (MQTT Subscriber)



Evaluation

Further Potential Development

- Applications:
  - Smart Garden / Greenhouse
  - Smart Climate Control in Homes
  - Smart Farms

- Functionality:
  - Additional Sensor Types
  - Switching Capabilities
  - GPS Reporting Capabilities
  - Bluetooth Integration

- Visualizations:
  - Mobile App
  - Improved UI
  - Additional Statistics and Analyses
  - Device Health and Location Tracking

References

1. AWS. “AWS IoT Core Documentation.” Developer Guide, AWS IoT, www.docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html.
2. AWS. “Amazon Elastic Compute Cloud Documentation.” Amazon Elastic Compute Cloud Documentation, User Guide for Linux Instances, www.aws.amazon.com/documentation/ec2/.
3. Cypress Semiconductor. CYW943907AEVAL1F Evaluation Kit User Guide. CYW943907AEVAL1F Evaluation Kit User Guide, Cypress, 2017.
4. Eclipse, Mosquitto. “Eclipse Mosquitto™ An Open Source MQTT Broker.” Eclipse Mosquitto, Documentation, www.mosquitto.org/.
5. “Spooky Statistics About Energy And Water Waste.” Www.erc-Co.org/, 13 July 2016, www.erc-co.org/spooky-statistics-about-energy-and-water-waste/.
6. “Welcome to My Smart Home: The 12 Best Devices to Make Your House Smart.” Fortune, Fortune, www.fortune.com/2017/02/17/smart-home-tech-internet-of-things-connected-home/.

Contributions of Group Members

Contributions of Group Members

Colton Powell

Contributions

- Created and administered the Amazon AWS EC2 server and the programs on it (httpd, mosquitto)
- Established communications between the IoT boards, the MQTT broker, and instances of the web application
- Assured that messages were formatted correctly and that they were correctly parsed and stored by instances of the web application
- Dealt with the dynamic creation and removal of graphs.

Challenges

- Properly drawing the graphs: Chart.js had a number of options to be set to ensure the graphs were drawn correctly
- Dealing with randomly-timed removal of graphs: Javascript is single threaded with essentially no concurrency controls, which made random, shared data structure access a struggle, though we eventually found a solid solution.
- Getting the MQTT communication pipeline between IoT boards, the MQTT broker, and instantiations of the web application working flawlessly

Contributions of Group Members

Matthew Salmanpour

Contributions

- Project introduction and definition of key target applications as well as value add for each application based on project's functionality
- Evaluation of AWS IoT for project implementation, including successfully publishing and subscribing to MQTT topic in AWS IoT console using Cypress Wi-Fi kit and modifications to WICED SDK example code
- Development of further potential areas and applications project can be expanded into as well as validation of sensors used in project
- Final report, presentation and demo video content creation and editing

Challenges

- Technical issues with establishing MQTT connection with AWS IoT and publisher/subscriber functions. Mitigated using Cypress documentation and additional example code.
- Lack of prior networking knowledge and comprehensive programming experience to initially understand all parts of project. Resolved by learning from Colton and additional research on resources outside of lecture notes

Contributions of Group Members

Justin Visas

Contributions

- Proposal slides and initial project schedule.
- Evaluation of AWS IoT for project implementation for publishing and subscribing to MQTT topic in AWS IoT console using Cypress Wi-Fi kit and modifications to WICED SDK example code.
- Made critical decisions for project timing such as demoing and recording to meet project timeline.
- Directed test setup and videos to enable optimal system and concept realization from viewer perspective.
- Final report, presentation and demo video content creation and editing.

Challenges

- Connection issues with AWS IoT and Cypress Eval board. Specifically, connecting to AWS MQTT broker. Met with Dr. Dezfouli and determined root cause is most likely a certification file issue.
- Generally inexperienced and unfamiliar with programming and related items in terms of background and knowledge of computer networks and internet architecture. Learned the majority of the background to be able to explain functionality at a system level from Colton / group project meetings.

Contributions of Group Members

Yezhou Zhao

Contributions

- Project planning and misc activities - participant
- Demonstration video - participant
- Final Report - content editing
- End to end encryption
  - Hardware crypto engine on Cypress board
  - Integrating aes-js crypto library into EC2 server
  - Encryption slides in presentation

Challenges

- Debugging Cypress board without OS or convenient debugging interfaces. Solutions required intuition and trial-and-error approach.
- Lack of documentation on hardware crypto engine interfaces and internals. Correct interface header found by comparing Cypress samples for software crypto to available headers in WICED.