Project Background

In the past, cyber security has been limited to the digital world, without a way to defend against physical attacks. While digital threats are still a relevant concern, special attention must also be paid to physical security of devices and data. If someone with enough knowledge or expertise were afforded physical access to a computer system, server, or data storage, they could cost an organization several thousands of dollars in damages.

The MAHIVE project has the goal of correlating physical and digital events into a system that can provide total security. It will utilize multiple elements to accomplish this. The main feature of the overall project is an AI-enabled server to correlate events together to identify an intruder or malicious actor. Our project is a smaller part; it covers the individual sensors that will act as the initial data collection.

Recent advances in microcontrollers and microprocessors now allow for sensors to be connected as IoT devices. This would be similar to a smart home setup, with a modular network of sensors reporting data to a single hub device. This hub would then relay all sensor data back to the server. Utilizing IoT devices like this would allow for a modular system that could be adapted to any number of buildings and situations.

Device Pros/Cons

Raspberry Pi 3

* Price: $35

Pros:

* Much greater processing power than other boards
* Low powered computer rather than microcontroller, much more functionality in terms of software
* Many supported libraries, excellent documentation and online resources
* Large variety of open-source software available

Cons:

* Costs more than any other device researched
* Much larger power consumption, would almost have to use AC power
* Overkill for a simple sensor device

Espressif ESP32

* Price: $11

Pros:

* Designed to have very low power consumption
* Built-in Wi-Fi and Bluetooth connectivity
* More processing power and larger memory size than ATmega328p in Arduino Uno
* Small board size ideal for IoT devices
* Supports Amazon FreeRTOS, can be used with AWS IoT Greengrass

Cons:

* Limited support and documentation compared to Arduino
* Development/programming may be more complicated than Arduino or Raspberry Pi
* If multiple units are needed, the cost may still be high

IoT Platform Description

AWS IoT Greengrass

* IoT Greengrass core device acts as a hub; it communicates with FreeRTOS-enabled devices
* Enables local execution of AWS Lambda code (Python code/functions)
* Messaging, data caching, and security built-in
* Would satisfy all 3 requirements with minimal coding
  + Time and energy could be spent on sensor devices themselves rather than the complexities of authentication and authorization

Device Pros/Cons

Arduino Uno

* Price: $19 (Official Model), $11 (Off-brand Equivalent)

Pros:

* Relatively low power consumption (~15mA minimum, could run off 9V battery for about 1 day)
* Highly supported with detailed documentation of libraries and IDE
* Support for many add-ons (shields) that add functionality such as Wi-Fi, ethernet connectivity, etc.

Cons:

* Low CPU power (only basic encryption supported)
* No built-in internet connectivity (Wi-Fi or ethernet) or Bluetooth (shields must be used)
* Extra shields may cost as much as the board itself
* Higher power consumption than other microcontrollers due to extra components on board

Arduino Mega

* Price: $28 (Official Model), $14 (Off-brand Equivalent)

Pros:

* Larger board, greater number of GPIO pins than Arduino Uno
* Uses ATmega2560 for greater processing power, can run basic encryption libraries and have enough resources as a sensor device
* Same level of support with libraries and IDE as Arduino Uno

Cons:

* Cheaper than Raspberry Pi, but still more expensive than almost any other microcontroller
* Greater power consumption, would not be able to run off battery
* Larger size may not be as suitable for IoT device
* No built-in internet or Bluetooth (same as Arduino Uno)

Raspberry Pi Zero W

* Price: $10 (Without Headers), $14 (With Headers)

Pros:

* Costs much less than Raspberry Pi 3
* Has all the processing power and functionality of 1st-generation Raspberry Pi, but in a much smaller size
* Nearly the same level of support and documentation as the full-size Raspberry Pi 3
* Nearly the same availability of open-source software

Cons:

* Still too overkill for a sensor, but not enough physical I/O to be used as a hub device (e.g. no built-in ethernet)
* Less CPU and memory resources compared to the full-size Raspberry Pi 3, meaning not all programs/software would be supported
* May still need to use AC power, could not run off battery

IoT Platform Comparison

Azure IoT Hub/Edge

* Main difference from AWS IoT Greengrass: core edge service is open-source
* Runs in cloud, integrates w/ IoT edge
* Raspberry Pi can act as gateway device
* Arduino/ESP32 connects/authenticates w/ hub
* Open-source runtime runs on Raspberry Pi, which acts as a middle man between sensors and Azure cloud service

No Third-Party Platform (Authentication/Authorization Developed from Scratch)

* Not tied to a single third-party company
* No need for special authorization to use
* Very involved process to program authentication, authorization, and messaging
* Much more time spent on communication and authentication
* Very little time spent on sensors themselves
* Much smaller likelihood of finishing project with proper deliverables

Important Links

AWS IoT Greengrass:

<https://aws.amazon.com/greengrass/>

Raspberry Pi 3 Model B+:

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>

Espressif ESP32:

<https://www.espressif.com/en/products/hardware/esp32/overview>

Design Constraints

* Design must be low-cost (goal for sensors: <$50)
  + No elaborate solutions that take large amounts of money to implement
* Design must use off-the shelf parts/boards
  + Raspberry Pi, Arduino, etc.
  + No custom-designed boards or parts
  + Utilize simple sensor parts readily available online
* Sensors should ideally run off battery
  + Board used for sensors should have be built for low power consumption
  + Device not constantly polling; sensor triggers interrupt instead
* Sensor device should be small and easily placed inside buildings
  + Should not attract attention to prevent tampering