Azure Zero Trust IoT Data Collection System

ECE4871 Senior Design Project

Team #8, Azure Zero Trust IoT
Project Faculty Advisor: Professor Brendan D. Saltaformaggio
External Partner: Dr. John D. Williams (Boeing), Michael F. Mitchell (Boeing)

James Thomas Computer Engineering jthomas8@gatech.edu
Aaron J Wasserman Computer Engineering wasserman@gatech.edu
Jayla Williams Computer Engineering jwilliams664@gatech.edu
Noah G Dorfman Computer Engineering n.dorfman00@gatech.edu
Zixuan Kang Computer Engineering zkang35@gatech.edu

Updated and Submitted

December 6, 2021

Table of Contents

Executive Summary

One of the most challenging aspects of analytics, data science, and machine learning is getting quality data. This project focuses on setting up a secured network of Internet of Things (IoT) devices that push data to the cloud, where it can be accessed for analysis.

The team will implement a Microsoft Azure Zero Trust connection between several Nordic microcontrollers using Arm Cortex-M3 chips (sRF52840, nRF9160, or nRF52832) and Azure Lake Storage for data collection. The network will be implemented using Bluetooth Low Energy (BLE) and will require a gateway device to the Azure IoT Broker. Devices in the system will communicate with each other through a Bluetooth Low Energy (BLE) mesh network. There are limited Zero Trust IoT solutions on the market today, making such a system valuable in IoT settings where security is of utmost priority.

The team's goal is to successfully program, install, and test a Nordic-based data collection system and show complete operation by recording data from the IoT sensors connected to the microcontroller and storing it on the remote server. With this functionality achieved, it is possible to demonstrate the security by conducting various network, software, and hardware-based attacks against our implementation. Proving the security will include testing the effective protection against backdoor intrusions such as wired taps through any analog or digital integrated circuit (IC) connections that might capture sensor data input to the microcontroller. We will examine the potential to corrupt the Zero Trust protection through power input irregularities using methods like glitching, also known as voltage fault injection, to cause corruption to instructions that could result in bypassed security checks. Our team will also be analyzing the implementation for susceptibility to side channel attacks like power analysis and timing attacks. Using fuzzing and industry-leading vulnerability scanners to verify the security of the software, we will prove the security of our code and network architecture.

Azure Zero Trust IoT Network

1. Introduction

Azure Zero Trust IoT Network team is requesting $500 amount of funding to develop a Zero Trust Architecture (ZTA) secured IoT mesh network for use in data collection and analysis to better inform management in environments such as factories, workshops, and warehouses.

1.1. Objectives

The objective of the Azure Zero Trust IoT Network is to successfully program, install, and test a Nordic-based system and show complete operation by recording data from the IoT sensors connected to the microcontroller and storing it on the remote server. The team will also demonstrate security by conducting various network, software, and hardware-based attacks against the completed implementation, including testing the effective protection against backdoor intrusions such as wired taps through analog or digital IC connections. Another critical part of the project is examining the potential to corrupt the Zero Trust protection through power input irregularities using methods like glitching, also

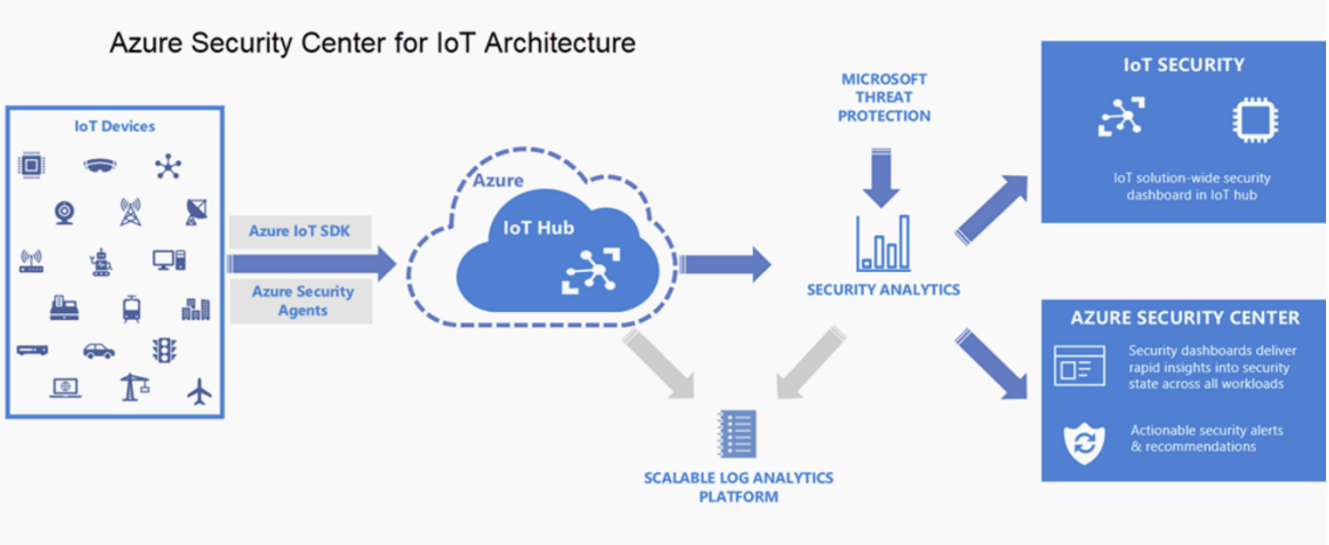

Figure 1. System Overview Block Diagram of the Azure Zero Trust IoT Network.

known as voltage fault injection, to cause corruption to instructions that could result in bypassed security checks. Figure 1 shows the overall structure of the Zero Trust Network. The team will establish a mesh network through BLE on the Nordic Thingy Microcontroller. One node in the mesh network will also be communicating with Azure IoT Hub through LTE. On the software and security side, the team will be taking advantage of Azure Security Center, which provides partial ZTA implementation and Azure IoT Hub for data collection and transmitting. Per the client's (Boeing's) requirement, we will also utilize Azure Lake Storage for data storing and analysis.

1.2. Motivation

Modern IoT devices promise to improve our daily lives by recording and analyzing the health and status of the world around us. This promise will be achieved using thousands of sensors continually recording and mapping temperature, vibration, luminescence, acceleration, velocity, humidity, heartbeat, oxygen content, color, and mood of the human existence. Combined with the promise of 5G, the world as we know it will soon change. However, this revolution will be short-lived without a means to protect and monetize the information and prevent others from stealing and making use of one's data. By developing the Azure Zero Trust IoT Network, the team hopes to increase security in the IoT networks for data collection in an easily deployable, well-vetted package.

1.3. Background

Historically, company servers resided inside the company network or within the company's geolocation. As a result, the company can easily define the border or the protection surface, which in this case is the server center of the company, and it is a feasible solution to prevent potential attacks through verifying the IPv4/IPv6 addresses, blocking access request outside the geolocation these addresses indicated and trusting all accesses within the company network (Castle-and-Moat Approach, CMA). However, as more companies and business models are using cloud services and outsourcing server needs, the conventional CMA is obsolete. Some of the most egregious data breaches happened

because hackers could move laterally through the internal system without much resistance. Under such circumstances, even the access from the internal network cannot be trusted anymore and thus leading to the ZTA, namely, always verifying and stop trusting [1], [2]. When it comes to the cybersecurity of IoT devices, application of this new cybersecurity architecture is of great urgency.

2. Project Description, Customer Requirements, and Goals

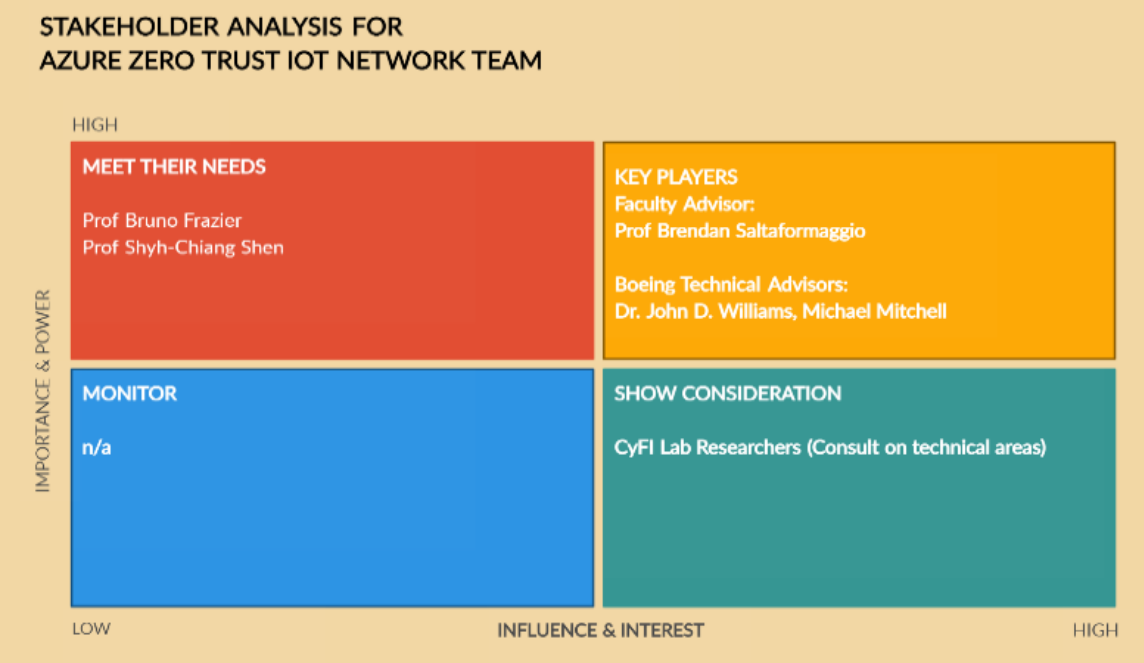

Figure 2. 2x2 Stakeholder Chart

The identified stakeholders for this project are Professor Bruno Frazier, Professor Shyh-Chiang Shen, Professor Brendan Saltaformaggio, the Cyber Forensics Innovation (CyFI) Lab Researchers, Dr. John D. Williams (Boeing), and Michael Mitchell (Boeing). The team identified Professor Frazier and Professor Shen as high importance, but low influence because their requirements on documentation have a direct influence on our grade. However, they are low influence due to the outline of our project needs being set by Boeing and their lack of frequent technical updates. Professor Saltaformaggio has high influence and importance because he is a security expert and will directly influence our grades. He also receives frequent technical updates and is involved in our design decisions. The Boeing advisors are also of high importance and influence because Boeing is our external partner on the project. They set the specifications and general direction of the project and are appropriately updated. The CyFI Lab Researchers are of high influence but low power because they are available for questions on the technical aspects of the project, but they have no effect on our course grade, nor do they have requirements that

the project should meet. Appendix A contains a quality function deployment (QFD) chart detailing the customer needs and engineering requirements.

The customer will require an inexpensive device that is fully functional with the code standards and meets the Zero Trust security standard. The device must be a Nordic Device able to operate on an LTE server that is lightweight with 4-hour battery life. The device must also pass a security assessment proving that it is resistant to wiretapping, not vulnerable to fuzzing, and not visible to access histories.

The design must comply with Azure Zero Trust and the System in Package (SiP) physical Trust solution required to operate a secure Nordic nRF9160 IoT device connected to an BLE mesh. It is supposed to have effective protection against backdoor intrusions such as wired taps through analog or digital IC connections capture sensor data input to the microcontroller with a low potential to corrupt the Zero Trust protection through power input irregularities. It should also demonstrate the trusted connection by recording data from IoT sensors connected to the microcontroller on the server. The device should meet both the Zero Trust Handshake and Microsoft security standards. This product must allow Azure Zero Trust to function on the latest Nordic Semi nRF9160 LTE-M / nB-IoT microcontroller for IoT data collection and a BLE mesh network.

3. Technical Specifications

3.1. General System

Item	Specification
Supported Number of Devices in Network	> 2 nodes
Cost	< $500
Data Collection Frequency	100 Hz
Battery Life (minimum)	4 hours

Table 1. General IoT System Specifications

3.2. Security

Item	Specification
Zero Trust Protocol	1 handshake/transmission
Hardware Access to Sensor Data	0 known vulnerable side-channel vectors
Software Access to Sensor Data	0 leaks to non-authorized accesses
Resistance to “Fuzzing”	0 device crashes

Table 2. Security Specifications for the IoT System

4. Design Approach and Details.

4.1. Design Concept Ideation, Constraints, Alternatives, and Tradeoffs

4.1.1. MQTT vs. AMQP

AMQP and MQTT are two strong candidates for the communication protocol used between IoT devices and cloud services. AMQP is more secure, more configurable, and more compatible with Microsoft Azure since many Azure database services only support AMQP. However, compared to MQTT, AMQP requires a lot of overhead and is more arduous to implement in an embedded system environment. MQTT is ideal for many small messages on low-bandwidth networks. Azure Lake Storage supports MMQTT and provides advanced data analysis, which meets the interests of Boeing. In the worst case where AMQP is necessary, Microsoft Azure's IoT Protocol Gateway offers a bridge between MQTT and AMQP. Azure Data Lake is the recommended endpoint for MQTT, as many of the additional Microsoft Services require AMQP internally.

4.1.2. Raspberry Pi Router Configuration vs. LTE (Verizon) on master node through Nordic Thingy

Currently, there are two ways to communicate with Azure IoT Hub. The team can use either a Raspberry Pi as a router or the LTE module of the Nordic Thingy. The Raspberry Pi route offers the potential to handle more complicated tasks. The Raspberry Pi is also easier to initialize and yields more community support. However, it requires an external device that may convolute the system designed to be lightweight. Employing the LTE module on the Nordic Thingy removes the overhead and decreases the complexity of security implementation. An LTE module also means that it can be useful anywhere with a cellular connection.

4.2. Preliminary Concept Selection and Justification

Many of the project requirements were presented in the design constraints by Boeing at the beginning of the project. The decision to use MQTT protocol was based on Azure's IoT Defender, IoT Hub services, and the Nordic SDK. In the case that MQTT does not supply the functionality desired, Azure IoT Protocol Gateway can be used to translate our packets into AMQP. The key decisions made were the design of the security experiments to test vulnerabilities over both the network and hardware that are addressed in section 4.3. Upon starting the experiments, considerations can be made on how to patch the software for the devices. A reasonable assumption is that vulnerabilities will be found. Simply showing data being deposited in Data Lake will suffice to test the mesh network's functionality and Azure endpoints.

Buying three Nordic Thingy:91 is needed before we can start evaluating a working model. Three devices are necessary for us to construct a mesh network and test functionality of a device being connected to the gateway. The team must also verify that a device that has not be instantiated as part of the network is denied access. Setting up this model will be done over the next few months during the summer. It follows that any experiments resulting in a vulnerability that needs to be patched will have to meet regression tests. As the working models are developed, regression tests will have to be developed alongside the working code to ensure our end-product works as intended.

For clarity, the system will require three Nordic Thingy:91 devices, the cost of the entire project will need to be less than $500. The data collection frequency for the devices to be competitive on the market will be 100hz and the standard battery life is 4 hours. In the Zero Trust architecture, it is typical that a device will have 1 handshake per actionable item given. If a device is sending a thermal reading, that device will be authenticated for each sensor reading sent to the cloud. After each actionable item, the authentication for the device will be dropped and a new handshake will be required. Zero known side

channels from our network and hardware experiments will be present in the final design. Fuzzing will be used to identify any possible software failures before they can become critical.

4.3. Engineering Analyses and Experiment

Proving the functionality of the design is an interesting challenge because security is of paramount importance and there are not explicit performance metrics. The few hard requirements that exist are easily proven. It is easy to show the network contains 3+ nodes and showing compliance with the minimum sampling rate can be done easily in a database. From these basic experiments, the team will work to validate the security by testing the implementation against several networking, software, and hardware-based attacks and showing resistance to these vectors. Said attacks are detailed below.

On the networking side, prototype testing will be done on both external wireless attacks and wired network attack vulnerabilities. This project will focus on attacks from fuzzing, use of the Nessus Vulnerability Scanner, and the use of AFLNET. The experiments will demonstrate the effectiveness, or lack thereof of Azure Zero Trust when implemented for IoT. If any vulnerabilities are found, they will be patched to ensure the design meets specifications.

On the hardware-side of the implementation, attacks can be classified into three categories: fully invasive, semi-invasive, and non-invasive [3], [4]. Fully and semi-invasive attacks are involved processes that require decapsulating ICs in question and using specialized lasers or ion beams to flip bits during operation [3]. These attacks have very high success rates but are costly and outside of an expected adversary's scope. As such, non-invasive attacks are the subject of this review. One widespread non-invasive attack is hardware glitching [3]. Hardware glitching induces brief "glitches" into an embedded device's power line and can cause instructions to be skipped or results to be corrupted [5]. Another standard non-invasive attack is power analysis [3]. Power analysis attacks depend on relationships between computation input values and the device's power consumption [4]. Correlations built using large sample sets can leak information about secret keys. Finally, timing attacks utilize variations in runtime

induced by different inputs. Operations can take different lengths of time to execute depending on the input data, and precise measurements of these times can allow an adversary to work backward and calculate the input [6]. The team plans to extensively test the implementation against all these attack vectors to ensure security against the most common hardware-based attacks.

4.4. Attacker Model

4.4.1 MQTT Threats

Our group plans to use MQTT as our machine-to-machine connectivity protocol. As such, it is important to consider attacks against the MQTT protocol or against common configuration errors that may be set by default through the Nordic API's MQTT implementation. The first critical configuration setting we will test against is authentication [1]. In MQTT, authentication is optional and unencrypted by default [1]. This plaintext-transmission of credentials subjects the system to man-in-the-middle attacks [1]. Additionally, MQTT brokers don't typically limit the number of authentication attempts [1]. To target this fact, we will attempt to use Ncrack to break the authentication.

4.4.2 Hardware Threats

Our team assumes that a would-be attacker has physical access to the hardware in the form of the Nordic data collection nodes. As such, we are also considering physical/hardware-based attacks that could be conducted. The most basic of which is simple wiretapping and ensuring that no JTAG/SWD/etc debugging ports are left open on the device. If such interfaces can't be disabled, we will investigate what information leakage or control is yielded and work to mitigate these effects. We will also attempt VCC-glitching aka Voltage fault injection (V-FI). Hardware glitching induces brief "glitches" into an embedded device's power line and can cause instructions to be skipped or results to be corrupted. Our aim would be to introduce glitches precisely timed to disrupt authentication routines. We will conduct these attacks using the ChipWhisperer platform. Finally, we will explore the Nordic development board's potential

susceptibility to power analysis or timing attacks if there are any cryptographic functions carried out on chip during the routine. Again, we will conduct these attacks using the ChipWhisperer platform.

4.4.3 Wireless Threats

When using Azure Zero Trust, we assume that all activity is malicious until proven otherwise. As a result, every access request is fully authenticated, authorized, and encrypted before granting access. We expect attackers to target the Nordic device using the Nessus Vulnerability Scanner and AFLNET. The Nessus Vulnerability scanner performs its scans by utilizing plugins, to run against each host on the network to identify vulnerabilities. AFLNET will act as a client and replay variations of the recordings of messages sent to the server and keep variations that identified progressive regions in the state space to steer in that direction. Any vulnerabilities found will be patched.

4.5. Codes and Standards

Azure IoT Central offers three choices for messaging: HTTP, AMQP, and MQTT. Between the three options offered we decided to go with MQTT. MQTT does have a community that will help inform decisions about security [7], however the AMQP protocol was designed with that in mind and is pushing to become the new standard [8]. MQTT was designed with resource constrained hardware in mind but more importantly we see it is supported in the Nordic SDK [7, 9]. The MQTT messages are simple and typically viewed as an exchange of data rather than a full messaging system [7]. Our plan is to push forward with MQTT because of its existing support in the Nordic and Azure communities [10]. HTTP seems to provide ease of implementation and the added benefit of the features that would normally come with HTTP, however comparing HTTP and AMQP or MQTT is challenging. AMQP and MQTT are IoT oriented whereas HTTP's use cases are varied [7, 8].

Presumably, our team is going to accomplish the firmware programming of the Nordic device through C/C++ [9]. Boeing also expressed to us that they would like our code to be publicly released as

a reference for implementing such a system and as such, it should follow all standard coding practices [11].

Per Boeing's request, our team will be using Nordic Microcontroller, Microsoft's Azure Zero Trust Architecture and Microsoft's IoT Hub to establish the network. Microsoft provides a list of Nordic Microcontrollers that are Microsoft certified [12] which can be served as a clear guidance about which device our team can choose. Microsoft also provides developers ample amount of guideline to follow for better compartmentalization, compatibility and thus less code upkeeping effort. Presumably, the team's code will be largely following the roadmap provided in the Microsoft Zero Trust Deployment Center, which will be separated into Secure Identity, Secure endpoints, Secure data and Visibility and Automation [13].

As detailed in the Project Description, our team will be implementing a Microsoft Azure Zero Trust connection between the latest Nordic Semi nRF9160 LTE-M / nB-IoT microcontroller. The outline in the NIST paper gives an abstract definition of zero trust architecture, general deployment models. and associated threats. These will affect the strategies that will be used to break into the device. One of the threats outlined is visibility on the network. The enterprise that cannot perform deep packet inspection or examine the encrypted traffic and must use other methods to assess a possible attacker on the network [14]. When attempting to break into the device using both external wireless and wired network connections, visibility will play a huge role.

Since Zero Trust Architecture relies heavily on identity verification, it is of great importance to establish a system that can handle authentication and error reporting when an unrecognized device is trying to connect to the mesh network. A useful Identity Authentication can provide developers/customers with the best way to authenticate valid devices while identifying threat without having major impact on user interface. For example, it is an option to establish an assurance level system for better Digital Identity Risk Management and create finer granularity to minimize the impact when a leak occurs [15].

5. Project Demonstration

One challenging element of demonstrating this system is the lack of hard, performance-based metrics that are easily displayed in a demo. Instead, the focus of this implementation is based heavily in security.

To first demo compliance with specifications, the team will present a functional mesh network of the Nordic devices that actively measure and transmit data to the database. Accompanying the mesh will be a laptop computer displaying the incoming data feed. Heating and cooling elements will be available to demonstrate changing sensor readings to any interested parties. This portion of the demo verifies adherence to the minimum specification of three or more Nordic devices and shows that we meet the minimum sampling rate.

Then, to present the more abstract notion of the system's security, the team plans a two-part system. Firstly, a poster or slide deck will detail all the attacks tested against the system showing robustness against all common attack vectors. For a more interactive/demo-friendly portion, a "dummy system" will be set up using a commonly used device for IoT such as a Raspberry Pi or Arduino with a barebones network setup. The team will have an interactive demo showing how easily one can extract secret data before demonstrating the same attack vectors failing to compromise the team's implementation.

6. Schedule, Tasks, and Milestones

The target for successful implementation of the Azure Zero Trust IoT network is the end of Summer 2021. This plan allows the team to use the Fall semester to identify vulnerabilities to attack vectors and harden the system against them and the remainder of the semester for expo preparation. Appendix B contains a Gantt chart with the major tasks, start and end dates, and the person(s) assigned to each task. Appendix C contains a PERT chart with corresponding analysis. The estimated completion times factoring in expected risk are included in this analysis, and the critical path is enumerated. The result of this analysis found that there is a 100% chance of completing the project one week before the Capstone Expo. This strong likelihood because the team plans to continue engineering efforts through the summer instead of adjourning until the Fall semester.

7. Marketing and Cost Analysis

7.1. Marketing Analysis

IoT and smart automation is not a new market. There are several existing solutions, and the market is growing. Current smart home automation offers in the market are priced at $750 for DIY models, while having a professional install and support a solution with features can be thousands of dollars. Custom solutions can exceed hundreds of thousands of dollars in some cases like Savant or Elan. Pricing the team's system at $1500 would be appropriate for the home automation market. The table below shows comparable home automation costs. However, more market analysis is needed to understand pricing for industry-based solution which is more in line with our solution.

	Fixr.com [16]	Homeadvisor.com [17]	Improvenet.com [18]
Home Automation Cost Average	$5500	$1000	$2077

Table 3. Competitor Pricing Information for Consumer Market

Current Industrial IoT, or IIoT, offerings provide a helpful community with plenty of documentation which the current business model does not offer. Directing more time and effort towards documentation and integration with existing platforms would be helpful next steps for growth. The key difference between home and industry automation is integrating with other business-to-business (B2B) platforms. Ignition and Predix both stand out as popular industry solutions with Ignition being recommended over Predix on enthusiast sites like reddit. The most basic solutions do not cover installation or penetration testing. Ignition is software you can deploy to any cloud system, while Predix is a subscription service through GE. For industry clients, we must re-evaluate the service being offered as compared to the savings on the bottom line of the client. Perhaps offering custom pricing on a per client basis.

	Ignition [19]	Predix [20]
Smart Industry	$16,850 (+$3,176 annually for support)	$48,000/yr (basic, everything else is very expensive)

Table 4. Competitor Pricing Information for Industrial Market

7.2. Cost Analysis

7.2.1.Bottom Line and Savings

Industrial Internet of Things solutions have exhibited success with several aspects that effect a company's bottom line. Many solutions are marketed as preventive maintenance. One hour of interruption for a manufacturer could mean hundreds of thousands of dollars lost. Ignition offers a website on case studies [21] as does GE’s Predix [22]. The use cases range from companies to cities and utilities, in some examples saving 7.2% on costs with only a few months required to set up the system.

7.2.2.Parts/Materials

The system implementation consists of an LTE gateway via the Thingy:91 and two other Thingy devices built in a network mesh to prove scalability and security. Access to the Azure Data Lake endpoint for data via personal laptops will be available using a python interface. The Raspberry Pi 4 will be used to prove alternative gateways for the BLE mesh network as well. Many of these items are already on hand and will not cost anything more. The Monthly IoT Hub rate for 3 devices is the price after the first free trial year.

Component Costs for Prototype

Item	Unit Cost	Quantity	Cost
NRF6943 (Thingy:91)	126.25	3	378.75
Power Supplies	8	4	32
Raspberry Pi 4	35	1	35
ChipWhisperer-Lite	250	1	250
Monthly Verizon Data Plan	30	6	180
Monthly IoT Hub For 3 devices	75	6	450
Total Cost			1325.75

Table 5. IoT System Prototype Cost Breakdown

7.2.3. Development Costs and Anticipated Selling Price and Profits

Five engineers will complete the design and development of the Azure Zero Trust IIoT solution. The total labor hours for the project per engineer are listed in the **Development Hours Per Engineer** table.

Development Hours Per Engineer

Task	Hours
Weekly Meetings	32
Reports	3
Research	7.5
Presentation	2
Assembly and Coding	15.1
Vulnerability Testing	23.7
Total Hours	83.3
Labor Cost per Engineer	2707.25
Labor Cost for Team	13536.25

Table 6. Engineering Labor Cost Breakdown

Labor costs were calculated using the total labor hours from the **Total Development Costs** table and an assumed salary of $65,000 [23]. The development cost for the prototype was determined from parts as $1325.75, which includes 3 units, resulting in a unit cost of $445.75. Assuming 30% fringe benefits of labor and 120% overhead on materials/labor/fringe benefits, the total development costs is shown in **Total Development Costs** table below.

Total Development Costs

Development Components	Cost
Parts	1325.75
Labor	13536.25
Fringe Benefits. % of Labor	4060.875
Subtotal	18922.875
Overhead, % of Material, labor, and fringe benefits	22707.45
Total Cost	41630.325

Table 7. Cost Summary

The profit over 5 years of production is given in the 5-Year Project below with the following assumptions. Assume 12 units a month are sold, and typically a mesh will require more nodes than this, but 12 units would be appropriate for small industries. A unit consists of a mesh of 2 BLE nodes and a gateway (3 Nordic Thingy:91). This estimate might be conservative for making a mesh network but could work in the worst-case scenario of home automation, which is a much smaller market, but still profitable. 10% sales expense was added on to the selling price as well. With a selling price of $1500, a profit of $240 per unit would be made, resulting in a 16% profit. This is a $172,821.60 profit over a fire year production. The selling price is low compared to other sources average price as seen previously, especially for a custom security-based implementation.

5-Year Projection

Expense or Income Component	Dollar Amount
Parts Cost	445.75
Assembly Labor	10
Testing Labor	15
Total Labor	25
Fringe benefits, % of labor	7.5
Subtotal	478.25
Overhead, % of Material, labor, and fringe benefits	573.9
Subtotal, Input Costs	1052.15
Sales Expense	150
Amortized Development Costs	57.81989583
Subtotal, All Costs	1259.969896
Profit	240.0301042
Selling Price	1500
Selling Price	**$1500**
Unit Profit:	**$240.0301042**
Percent Profit:	**16%**

Table 8. Future Financial Projection

Platforms being offered by other companies show that there is a fair amount of integration with other B2B services, and the documentation is more thorough, still, many platforms are built on top of Microsoft Azure IoT Hub and charge upwards of $3.7 million dollars per year for the use of a system in the case of GE. This cost is excluding custom applications or apps built on top of the analytics. Targeting industry markets instead of home automation while considering the past and predicted growth of the IoT market alongside what the IIoT solution offers customers bottom lines, a re-evaluation of what we are selling the product could yield significantly higher profits future.

8. Current Status

The primary focuses of this semester have been planning and research. Premilitary research is 100% complete. The team divided initial research into following subtasks: reading the Choi paper, reading the Cloud City paper, the project design standards & codes assignment, the project job budget & costing assignment and completing the technical review papers. At this stage, the planning is at about 90% complete. The completion of the subtasks: project design standards & codes assignment, the project job Budget & costing assignment, project Gantt chart, PERT Analysis, project summary form, and team skills matrix attributed to the percentage. As of now, the remainder of the planning phase will depend on feedback from both the faculty advisor and the company. Although implementation has not begun, the immediate next step is to order the Nordic devices. Once they have arrived, the summer semester will be dedicated to the initial setup and testing of basic functionality.

9. Leadership Roles

To effectively utilize the skillsets of each team member and ensure all project critical tasks are completed, leadership roles were assigned for the 4871 and 4872 semesters.

4871 Leadership Roles:
Jayla Williams - Networking Lead
Aaron Wasserman - Hardware Security Testing Lead
Noah Dorfman - Documentation, Embedded Hardware Lead
James Thomas - Advisor point of contact, Azure IoT Software Lead
Harry Kang - Boeing point of contact, Zero Trust Architecture Consultant, Embedded Software Lead

Jayla served as the networking lead and was responsible for researching the networking theory necessary for our implementation and all software/network-based security vulnerabilities that may occur. Aaron served as the hardware security testing lead and researched modern security vulnerabilities affecting the hardware of IoT systems to ensure we can test our system against a capable attacker. Noah was the documentation coordinator and was responsible for ensuring that our documentation was complete and coherent. He was also the embedded hardware lead and researched the Nordic devices to be used in the team's implementation. James acted as a point of contact with our advisor and set up all meetings between the team and Professor Saltaformaggio. James also served as the Azure IoT software lead, where he informed design decisions based on his research of the available services and protocols. As the Boeing point of contact, Harry was responsible for coordinating all meetings and email exchanges between our team and Boeing's technical advisors. In addition to this role, Harry was the Zero Trust Architecture Consultant, making him responsible for researching the Zero Trust Architecture to inform design decisions.

4872 Leadership Roles:
Jayla Williams - Webmaster, Networking Lead
Aaron Wasserman - Expo Coordinator, Hardware Security Testing Lead
James Thomas - Azure IoT Software Lead
Noah Dorfman - Documentation, Embedded Hardware Lead
Harry Kang - Implementation Testing, Embedded Software Lead

As the networking lead, Jayla will be responsible for analyzing the function of the implementation's network components and testing the security against a range of software/network-focused attacks. She will also serve as webmaster. Aaron will act as the expo coordinator and prepare all presentation materials and demos in advance of the showcase. He will also serve as the Hardware Security Testing lead, where he will attempt to exploit hardware-focused attack vectors to ensure the security of the implementation. James will act as the Azure IoT software lead and work to implement our Nordic mesh network with the Azure services. Noah will continue as documentation coordinator and ensure that all documentation is complete and coherent. He will also lead the Nordic devices' tasks, including data collection/sensor reading and setting up a mesh network. Harry will be responsible for verifying that our implementation successfully measures environment data and stores it in the selected database. Additionally, he will lead in the embedded software/firmware development to implement the system.

10. References

[1] Pratt, M. K. (2018, January 16). “Security for a New World” in *What is Zero Trust? A model for more effective security*. CSO Online. https://www.csoonline.com/article/3247848/what-is-zero-trust-a-model-for-more-effective-security.html

[2] SANS Institute, & Marchany, R. M. (2019, July 23). *Zero-Trust Networks: The Future Is Here* [Presentation and Panel Discussion]. SANS Blue Team Summit 2019, Blacksburg, Virginia. https://www.youtube.com/watch?v=EF_0dr8WkX8

[3] C. Quast, Conference Presentation, Topic: "Common Attacks on IoT Devices,” Embedded Linux Conference Europe, Edinburgh, UK, October 23, 2018. Available: https://elinux.org/images/f/f8/Common-Attacks-on-IoT-Devices-Christina-Quast.pdf

[4] S. Mangard, E. Oswald, and T. Popp, Power Analysis Attacks: Revealing the Secrets of Smart Cards. New York, NY: Springer, 2007.

[5] NewAE Technology, “Part 2, Topic 1: Introduction to Voltage Glitching (MAIN),” *readthedocs.io*, para. 2019. [Online]. Available: https://chipwhisperer.readthedocs.io/en/latest/. [Accessed March 7, 2021].

[6] S. Takarabt., A. Schaub, A. Facon, S. Guilley, L. Sauvage, Y. Souissi, and Y. Matthieu, “Cache-Timing Attacks Still Threaten IoT Devices,” *Codes, Cryptology and Information Security*, p. 13-30, March 2019. Available: https://www.researchgate.net/publication/332411858_Cache-Timing_Attacks_Still_Threaten_IoT_Devices. [Accessed March 7, 2021].

[7] *OASIS MQTT*, Version 3.1.1, Oct. 2019 [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf [Accessed March 6, 2021]

[8] “AMQP v1.0,” *AMQP*, 07-Oct-2011. [Online]. Available: https://www.amqp.org/sites/amqp.org/files/amqp.pdf. [Accessed March 26, 2021].

[9] *Nordic Semiconductor Infocenter*, March 2017. [Online]. Available: https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.sdk5.v13.0.0%2Findex.html. [Accessed March 26, 2021].

[10] dominicbetts, DCtheGeek, v-lakast, v-hearya, philmea, robinsh, YutongTie-MSFT, DennisLee, v- kents, and BryanLa, “Introduction to the Azure Internet of Things (IoT),” *Introduction to the Azure Internet of Things (IoT) | Microsoft Docs*, January 15, 2020. [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-introduction. [Accessed March 26, 2021].

[11] *Embedded C Coding Standard*, Edition: BARR-C: 2018, BARR Group. [Online]. Available: https://barrgroup.com/sites/default/files/barr_c_coding_standard_2018.pdf. [Accessed March 26, 2021]

[12] *Azure Certified Device catalog*. [Online]. Available: https://devicecatalog.azure.com/. [Accessed March 26, 2021].

[13] garycentric, Kellylorenebaker, and cmcclister, “Zero Trust Deployment Center,” *Microsoft Docs*, September 30, 2020. [Online]. Available: https://docs.microsoft.com/en-us/security/zero-trust/. [Accessed March 26, 2021].

[14] S. Rose, “Zero Trust Architecture,” August 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf. [Accessed March 26, 2021].

[15] M. E. Garcia, P. A. Grassi, and J. L. Fenton, “NIST Special Publication 800-63-3: Digital Identity Guidelines,” *National Institute of Standards and Technology*, March 2, 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-3.pdf. [Accessed: March 26, 2021].

[16] “2021 Home Automation Cost: Smart Home Costs,” *Fixr.com*, 04-Mar-2021. [Online]. Available: https://www.fixr.com/costs/home-automation. [Accessed: April 16, 2021].

[17] “Learn how much it costs to Install a Home Automation System.,” *HomeAdvisor*. [Online]. Available: https://www.homeadvisor.com/cost/electrical/install-or-repair-a-home-automation-system/. [Accessed: April 16, 2021].

[18] “How Much Does it Cost to Install Home Automation Systems?,” *2021 Home Automation System Costs | Smart Home Automation*. [Online]. Available: https://www.improvenet.com/r/costs-and-prices/home-automation-system-cost. [Accessed: April 16, 2021].

[19] “Transparent Pricing. Flexible Options.,” *Ignition Software Pricing for SCADA, IIoT, MES and More*. [Online]. Available: https://inductiveautomation.com/pricing/ignition. [Accessed: April 16, 2021].

[20] J. LaChance and J. Seay, “Competing to Win,” in *GE Digital Alliance Program*, April 16, 2021.

[21] “Case Studies,” *Featured | Inductive Automation*. [Online]. Available: https://inductiveautomation.com/resources/casestudy. [Accessed: April 16, 2021].

[22] “Customer Stories: Industrial Internet: GE Digital,” *Customer Stories | Industrial Internet | GE Digital*. [Online]. Available: https://www.ge.com/digital/customers. [Accessed: April 16, 2021].

[23] “What is ECE at Georgia Tech?,” *School of Electrical and Computer Engineering at the Georgia Institute of Technology*. [Online]. Available: https://www.ece.gatech.edu/what-ece-georgia-tech. [Accessed: April 16, 2021]

Appendix A – Quality Function Deployment (QFD)

See the following pages for the team quality function deployment (QFD).

Appendix B – Project Gantt Chart

See the following pages for project Gantt chart.

Appendix C – Project PERT Chart and Analysis

See the following pages for project PERT chart and corresponding analysis.