Computer Science Project

40 Credits

IoT enabled physical intrusion detection and foot traffic monitoring system

Matthew Quaynor

DA B.Sc. Computer Science with Digital Technology Partnership (PwC)

ID: 2136446

Supervised by: Eike Ritter

Word Count: 5151

Project Repository: https://git.cs.bham.ac.uk/projects-2023-24/maq046

School of Computer Science

College of Engineering and Physical Sciences

University of Birmingham

2023-24

Abstract

Physical security is a fundamental desire and necessity throughout society - for homes and businesses alike. The act of securing a building or property against physical threats often relies on some form of foot traffic monitoring, a frequently desirable offering for businesses that rely on the efficient movement of people in a building. Analysis of such patterns in foot traffic can be beneficial for many businesses - such as retail stores, warehouses and office buildings - that may wish to analyse movement patterns to improve efficiency or gain insight into consumer behaviour. The Internet of Things can be used to implement solutions for both physical intrusion detection and foot traffic monitoring, utilising sensors that interact with the physical world to autonomously generate data. Despite the great overlap in functionality, very few products on the market implement a system that offers both intrusion detection and foot traffic insights. In addition, many such systems are offered to consumers at extremely high cost, whether through purchase of hardware and installation, subscriptions to software offerings or both. In this work, we aim to implement a secure, low-cost system that can be utilised for both detecting physical intrusions and monitoring foot traffic within a building. Our solution utilises the Internet of Things and sensors for receiving foot traffic data, which is interpreted by a central web application for detecting intrusions and storing sensor data.

Contents

CHAPTER 1

Introduction

Security is often a fundamental desire for much of society. Whether to secure the safety of oneself, loved ones, possessions or data, the issue of security is a problem as old as society itself. Often this involves ensuring buildings such as homes or places of work are secured and monitored for potential threats of physical harm or theft. As such, this can include foot traffic monitoring, a desirable feature to many businesses such as those operating retail stores, warehouses or office buildings. Foot traffic data is frequently used by such businesses to gain insights into employee efficiency and consumer behaviour. A monitoring task like this is laborious and monotonous; a task best be left to autonomous systems.

There exist several security and foot traffic monitoring solutions on the market. Many of these systems come at high cost to consumers, possibly inhibiting financially conscious individuals and small businesses from adopting them. It is our opinion that if a similar, reasonably effective solution can be developed at a lower cost to end consumers, this should be made available to them - even if this means sacrificing features that are ”nice to have”. Given the overlap in functionality between security and foot traffic monitoring systems, we also believe it possible to implement a system encompassing both of these domains.

The Internet of Things (IoT) consists of a network of devices connected through the Internet in such a manner as to facilitate data transfer without human intervention. These devices are frequently integrated into everyday objects, allowing for interaction between the physical world and software. The Internet of Things spans a vast range of domains including transport, healthcare, military, commerce and home utilities, or ”smart homes”.

These ”things” devices are often constrained by minimal hardware and programmed for very specific tasks - such as operating a sensor and sending data to another part of the network. As such, traditional security protocols and mechanisms have been found to be unsuitable for IoT devices [1]. Consequently, developing secure systems that utilise IoT devices has become a challenge. This, in conjunction with the rapid adoption of the Internet of Things [2], has led to attempts to develop protocols, architectures and security frameworks for IoT integrated systems [3]–[9].

In this work we aim to develop a cost-effective physical intrusion detection and foot traffic monitoring system, utilising the Internet of Things to detect and relay foot traffic data. We first discuss current research and similar existing systems available to consumers and lay out our system requirements. We outline our design and implementation, which includes the use of motion and distance sensors controlled by ESP32 micro-controllers, an on-premise server and a cloud server, all communicating via REST APIs. We then discuss our findings in testing, as well as any limitations of these tests. Lastly, we compare and evaluate our design for overall effectiveness in accurately detecting intrusions and foot traffic, software security and financial cost.

CHAPTER 2

Research

2.1 Existing Systems

Many physical security and foot traffic monitoring systems are available on the market. These solutions often come at a high cost to consumers and, despite possessing the hardware capabilities and similar backend functionality; offerings are frequently divided between physical security systems and foot traffic monitoring systems, without much overlap. The high costs and lack of crossover in provided features of these systems may prove inhibiting to small businesses such as stores seeking to gain the benefits of both tracking customers during operating hours and intrusion detection technology.

A popular example of the former would be Ring [10] who offer camera doorbells and alarm systems, primarily marketed for use in the home. At the time of writing, these security systems have a retail price starting at £219.99 for a system including one motion sensor and one contact sensor, with the ability to add additional sensors at extra cost. These systems also include a companion mobile app where notifications are sent in the event of an alarm being triggered at no additional cost - except for extra features requiring a subscription. These systems, however, do not offer a platform for viewing foot traffic detected over time outside of when an alarm is activated. One large ethical issue relating to systems similar to Ring that is frequently discussed in the media, particularly in the US, is whether their wide-scale adoption constitutes an instance of participatory mass surveillance [11]. As such, it is important to tread carefully with implementing and installing systems such as these, ensuring all uses comply with local laws and any parties participating in testing are informed. Due to these privacy concerns and the higher hardware and storage costs associated with the use of cameras, we will avoid using photo and video surveillance.

Foot traffic monitoring systems, on the other hand, are generally marketed for industrial and business use, such as in warehouse buildings and stores. These tend to start at a higher cost for hardware and installation - likely due to additional hardware requirements for larger buildings, such as wiring and additional sensors. Such systems often operate subscription-based services, and offer features centred around data analytics for businesses from sensor data. As an example in the US, Dor [12] offer their services from $99 per year to retailers and utilise thermal sensors for generating data. Other proposed methods of gathering foot traffic data include taking advantage of the prevalence of smartphones to track WiFi and Bluetooth signals. This raises the issue of ensuring the data is anonymised to comply with data protection laws in the UK and EU. Additionally, this method is unsuitable for a security system; relying on an intruder to either have a smartphone, be connected to one's own WiFi network or both would be ludicrous. As such, sensors with physical inputs, such as motion and ultrasonic or laser trip sensors, will be preferred.

In this work, we aim to develop a system that can be used as both a physical security system and foot traffic monitor while minimising the cost of hardware, where possible.

2.2 Current Research

On the whole, the general state of security for IoT devices is oft observed to be unreliable and filled with vulnerabilities[6], [13], [14], possibly in part due to a lack of any standardised framework for implementing systems that utilise such devices [1]. In the USA, NIST have only recently published discussions and continue to work on recommendations for IoT security in their Cyber Security for the Internet of Things program [7], highlighting the relative infancy of this field.

Despite this, there has been greatly increasing research into the cyber security of IoT enabled systems and suggested architectures in recent years, with several suggested architectures and practices. Implementing secure systems involving IoT devices are limited by the relatively low computing power and storage of "things" devices. As such, usual protocols and practices, such as public key infrastructure and firewalls, become infeasible [5], [15]. Consequently, several protocols for authenticating IoT devices have been proposed.

The authors of [8] propose a two factor authentication scheme for wireless sensor networks that utilises elliptic curve cryptography; they also highlight flaws in a previously proposed scheme [9]. The authors show that this scheme is verifiable, provable and privacy preserving - satisfying authorisation, confidentiality and integrity properties of security. The authors of [3] propose a three factor authentication scheme for wireless sensor networks, providing user anonymity and claims resistance to several threats, including side channel, forgery, known session-specific temporary information and insider attacks. More recently, there have been attempts to solve the issue of IoT security using Machine Learning and Artificial Intelligence[16], however, the sheer nature of these approaches requires large amounts of computational power, which is often not present in IoT devices.

In [5], authors propose a middleware architecture to provide end-to-end security by isolating IoT devices from the wider internet; devices are only given access to an authentication gateway, which acts as an interface between the IoT layer, a middleware application and the internet. Their approach utilises REST APIs and traditional cryptography algorithms between the middleware and adjacent layers, removing the need to implement a new protocol to integrate IoT devices into a system. This approach, however, requires an additional element in the system architecture, as compared to allowing IoT devices with WiFi capabilities direct access to the internet.

In [4], the authors propose a lightweight authentication and data transmission scheme, highlighting flaws in previous attempts as being too computationally expensive for IoT. The authors claim this scheme is resistant to various common attacks including replay, eavesdropping, modification and masquerade. Upon analysis in a simulation based environment, the authors conclude that their method significantly outperforms existing schemes with lower energy consumption, storage costs, communication costs and up to 66% lower computational costs.

We implement a system architecture with greater similarity to [5], and attempt to implement strong security practices, such as least privilege, while minimising overall financial cost, as compared to existing systems on the market.

CHAPTER 3

System Requirements

The following requirements detail features and functionality intended for the system. These functional and non-functional requirements are categorised using the MoSCoW methodology, whereby features are prioritised based on perceived difficulty and business benefit, according to a predefined time frame. These categories are generally labelled, in descending order of priority; **must**, **should**, **could** and **would** (or *won't*). Those with lower perceived difficulty that provide higher, or more immediate, business benefits are given higher priority than others. This categorisation is often subjective, particularly regarding how difficulty and benefit are defined or measured. This results in the occasional tendency for developers to prioritise "quick wins" over features that require more effort but provide greater, but perhaps less immediate, long term benefit to users, stakeholders or both. We use this methodology in combination with iterative Agile development, aiming to produce a minimum viable product - or proof of concept - before implementing more complex features.

3.1 Functional Requirements

The system:

- **must** detect objects or persons that pass a designated physical space.
- **must** create a log and/or alert within a short time of a sensor being 'tripped'.
- **must** provide means of viewing all 'intrusions' or sensor traffic detected.
- **should** facilitate deactivation of intrusion alerts
- **could** facilitate activation of intrusion alerts for only particular areas.
- **could** create an alert if any registered device becomes disconnected - as this could be due to a device becoming physically compromised.

- **would** provide a data viewing platform for analysing foot traffic data - since this depends on an otherwise working system being implemented first and implementation may not be possible in the designated time frame.

3.2 Non-Functional Requirements

The system:

- **must** facilitate low latency communication between different elements in the architecture - In the event of intrusion detection, users should be alerted promptly, so that appropriate action can be taken to prevent or minimise damage caused.
- **must** store all detected sensor traffic in a structured database with device details and a timestamp - for a user and application to access and potentially review in future
- **must** allow new IoT devices to be registered at runtime - to initialise devices generating sensor data.
- **must** continuously monitor an area for foot traffic or other objects passing - breaks in monitoring could lead to undetected intrusions and would result in less foot traffic data being gathered
- **should** not store any login credentials in plain text - to prevent threat actors from gaining access to login credentials should a device be compromised.
- **should** use encryption for any communication over the internet - to prevent snooping login credentials or leaking user data.
- **should** prevent remote access to IoT devices from arbitrary devices - to prevent threat actors from gaining unauthorised access to the IoT device and any data it has access to.
- **would** periodically ”ping” other devices in the system to check they are still connected or active - this would allow us to check whether IoT devices are still operational without manually checking and help alert users if a device becomes physically compromised.

CHAPTER 4

Design and Implementation

4.1 Overview

For our design, we opted for an architecture divided into three main areas: IoT and sensors, an on-premise application to act as a device registration gateway, and a web application operating on a cloud server. All communication between each domain or layer is completed via REST APIs with endpoints that either require authentication via login or are not exposed to wider networks and devices.

4.2 Hardware

In terms of hardware, we require sensors that will detect foot traffic as well as a programmable microcontroller to communicate with these sensors and other elements in the system. Several microcontroller boards are available on the market with varying costs and features. For this project, we opted to use an ESP32-WROOM board, developed by Espressif Systems [17], for its compatibility with the widely supported Arduino IDE [18] and WiFi connectivity capabilities. In keeping with the theme of minimising financial cost; only infrared motion sensors and ultrasonic distance sensors were chosen for detecting foot traffic. An advantage of using these types of sensors is that the anonymity of monitored persons is preserved when used merely for foot traffic monitoring purposes. The motion sensors utilise pyroelectric detectors to detect any motion of bodies emitting infrared radiation. The distance sensors meanwhile, work by calculating the time between emitting an ultrasonic sound and receiving the reflected sound from surfaces - such as walls and humans - in front of it.

4.3 Cloud Server

The web application runs on a cloud server and serves as the primary platform for interpreting and storing sensor data. This application is implemented in Python 3 [19] and uses the Flask web framework [20], exposing a REST API for the on-premise application and IoT devices to query via HTTPS requests.

For security and privacy purposes; most endpoints are protected, requiring a user or device to authenticate with a username and password and receive a JSON Web Token (JWT) which can be included in future requests to access protected endpoints. This was implemented using the Flask-JWT-Extended python package [21]. All passwords are hashed using the flask-bcrypt python library [22] and stored in a SQL database. For an admin user, this permits access to all endpoints, including those that may alter the application configuration. Meanwhile, IoT devices are only granted access to an endpoint for sending sensor data to the server, maintaining the security practice of implementing least privilege. When an IoT device sends a request to this endpoint - upon a sensor being tripped - the application creates a log containing information corresponding to the device whose sensor was tripped, whether this constituted an 'intrusion' (if the alarm is activated) and a timestamp. If the alarm is activated, an alert is created. Additionally, a minimal front end is provided for a user to view all intrusions detected since start up.

4.4 on-premise Application

The on-premise application serves as an initialisation gateway between IoT devices and the web server. This application runs on a server on the local access network (LAN), exposing an endpoint only to the LAN for IoT devices to register and receive login credentials for the web application. When a new device attempts to register, the on-premise server receives a unique identifier in the request to serve as the device username and generates a random password for this device. The on-premise server, possessing admin login credentials, will authenticate to the web server and send the new IoT device's credentials to an admin-protected endpoint in order to register the new sensor device in the web application. Upon successful registration, the on-premise server then sends the generated password to the IoT device - ensuring it is not stored locally - allowing the IoT device to authenticate to the web application directly. This prevents storing login credentials on the IoT device in plain text, preventing credentials from being stolen should the physical security of the IoT device become compromised. This also saves on the computing power and storage needed from the IoT micro-controllers, since we will not have to generate a secure, random password on the IoT devices at runtime. In other potential implementations, this on-premise application could act more as a middleware, forcing IoT devices to only connect to it, preventing exposure to the wider internet. Such an implementation as in [5] would have IoT devices sit behind the middleware and an IoT gateway, where authentication would be performed to prevent unauthorised access between IoT devices and the web (including IoT applications).

4.5 Internet of Things

Each IoT device consists of an ESP32 microcontroller board connected to one or more sensors. The micro-controllers are programmed using the Arduino-ESP32 library [23]. These ESP32 devices operate by initially calling a setup function, and then repeatedly calling a loop function while powered on.

On startup, within the setup function, the IoT devices must register and authenticate with the web application before sending sensor data. To prevent the storage of login credentials in plain text, the devices send an HTTP request to the on-premise application REST API, which operates on the same local access network. The request will include a unique device identifier, to later act as a username for logging into the web application. From the perspective of the IoT device, it will then receive a randomised string in response, which is saved in a variable at runtime to act as a password for the web application. The device then uses these credentials to make an HTTPS request to a login endpoint on the web application's REST API, receiving a JWT in response which can be used thereafter to send requests to the protected /sense endpoint for recording sensor data. Storing the password and JWT in variables at runtime means they will not persist when the device is powered off.

The IoT device will then constantly monitor connected sensors in the loop function. In the case of the motion sensors, these will monitor for infrared movement whilst the distance sensors will check if, at any point, its distance to the nearest object drops below a particular threshold. The distance sensor measures the time (in microseconds) between emitting an ultrasonic sound and receiving the echoed sound back; using this time we can approximate the distance between the sensor and the object that reflected the sound, using the speed of sound in air (approximately 343m/s in air at 20°C [24]) as follows:

$$\begin{aligned} d &= 343\,m\,s^{-1} \times t\,\mu s \div 2 \\ &= 343 \times 100 \div 10^6 \times t \div 2\ cm \\ &= 0.0343 \times t \div 2\ cm \\ &\approx t \div 58.3\ cm \end{aligned}$$

Using this, we can program a threshold that is significantly less than the standard distance between the sensor and the wall or object it usually faces. When the detected distance drops below this threshold we know that something has passed in front of the sensor. For example, if mounted on a wall in a corridor that is 1m wide, we can set the threshold to less than 85cm. In this sense, the distance sensor can act like a trip wire.

When any IoT device determines that a body or object has passed it, it will then make an HTTPS request to the web application via a protected endpoint, including its previously obtained JWT in the request header for authentication and identification purposes. To save on storage and computation, as well as for simplicity, this operation is always performed whenever a sensor is tripped, leaving the interpretation of whether this constitutes an intrusion or not up to the web application logic.

CHAPTER 5

Evaluation

5.1 Testing

Upon implementation of the features detailed previously, we conducted end-to-end testing within a single occupancy studio and a house. Some difficulty arose in attempting to position the IoT devices in such a manner that they could be both powered and able to yield meaningful sensor data. This inhibited our ability to test the system in many different environments. Introducing a battery to the hardware design would allow us to work around this. However, this would increase the cost of hardware and create inconvenience in that batteries would periodically need to be replaced or recharged. For this reason, we believe it would be best to simply install the IoT devices in a way that allows them to be constantly powered by the main power supply in the building.

In testing the sensor behaviour alone, we found the motion sensor to work as intended, whilst our implementation of approximating distance with the ultrasonic sensors was found to be accurate within 2cm in the environments tested. We deemed this to be more than precise enough for our purposes. It remains to be seen if this would change significantly in environments with differing temperatures and pressures; however, it is unlikely to vary enough to compromise the validity of our approximations. An alternative implementation may be to keep track of the last distance value recorded and only create an alert if this value decreases by some significant amount between loops, as opposed to creating an alert when the value drops below a particular threshold.

Two unexpected test subjects, Prince and Tinker (the author's cats), highlighted a flaw in the design when utilised in certain environments; sensors can be tripped by animals, such as pets, resulting in potential false positives. Such false positives may skew foot traffic data in a greatly expanded system if utilised in larger public areas such as shopping centres, where wildlife will almost certainly trip sensors. For use in the home, a possible workaround for the distance sensors involved placing them at a height that would vertically clear the height of a cat and most dogs, aiming the sensor horizontally across walkways. However, this raised the issue that an intruder could simply crawl under the plane the distance sensor looks along - avoiding detection. No solution could be found for the motion sensors. For these reasons, it is easy to see why many large scale foot traffic detection systems opt to gather data through WiFi, Bluetooth or mobile network signals. As such, the system in its current state is better suited to controlled indoor environments, where false positives will either not occur or are highly unlikely to cause significant changes in data trends. As for intrusion detection, this highlighted that the types of sensors used may not be entirely suitable unless the environment is expected to be completely static whilst an alarm is active.

Apart from the above, we can reasonably say that the system performed as expected; having implemented all of the **must** and most of the **should** have system requirements within the time frame.

5.2 Limitations

One limitation and potential security flaw highlighted in our design was storing WiFi credentials in plain text in the code of IoT devices. Should these devices be physically compromised or stolen, threat actors may be able to extract the network credentials from the device. A solution to this could be to create a gateway accessible via WiFi for a user to enter the permanent WiFi credentials from another device at runtime such as in [25]. This temporary gateway could be password protected and have traffic encrypted between it and other devices, preventing third party threat actors from viewing the permanent LAN credentials in the open. In some implementations, this may be difficult with device storage constraints.

It may also be good practice to keep IoT devices on a local access network that is segregated from normal users when integrated into larger networks, so as to minimise the potential attack surface if LAN credentials are somehow compromised via an IoT device. Alternatively, microcontrollers could be connected to the on-premise server through wire alone to prevent theft of WiFi credentials. However, this would drastically increase the cost of wiring and installation in many cases and remove the portability aspect of the IoT devices. Additionally, it would then make sense to perform all communication between the cloud server and IoT layer via the on-premise application, removing the need for WiFi connectivity in the microcontroller boards. Using boards without WiFi capabilities may reduce the cost of hardware slightly, but ultimately would result in a completely different type of system, demanding far more set-up time and with very little ability to adapt to small changes. The portability aspect of these devices would be completely lost, requiring far more effort to simply relocate a device from one part of a building to another.

As discussed above, much of the security of this system partially depends on keeping these devices physically secure, as well as determining how to secure our data and software should a device be physically compromised. This may come across as circular logic; the overall security of our physical security system depends on whether the devices in the system are physically secure themselves. This, in turn, highlights the true difficulty of the issues faced in providing physical security through electronic hardware and software. Much of this will depend on how the devices are installed and the ability of the security provider to detect tampering. At current this system does not provide any means of detecting malicious tampering.

5.3 Cost analysis

At the time of purchase (November 2023), three ESP32 micro-controller units were obtained for £14.99, whilst both types of sensors used were obtained at a cost of £9.99 for 5 units, at the time of purchase (November 2023). An additional £12.73 was also spent on additional wiring and breadboards for prototyping, bringing the total cost for prototype hardware at the end of 2023 up to £47.70, with wires to spare. It can be safely assumed that production value hardware would likely incur additional costs in design, housing, installation and labour. Thus, our choice to minimise the hardware required for cost reasons follows logically. Other systems on the market such as Ring [10] frequently utilise cameras, which can offer greater features particularly when marketed as a security system. However, cameras can often come in at much higher cost for hardware, computation and data storage - given photos and video have relatively large file sizes. For utilisation in a home or small business, this may not be perceived as a necessary feature when keeping costs low plays a large factor in decision-making. Lastly, as discussed above, providing power supply to our IoT devices by introducing a battery into the design may increase the cost, and we believe the potential benefit may not be worth the cost of additional hardware. As such, it can be reasonably believed that the cost of hardware and installation for a system such as ours could potentially be done at a slightly lower cost per unit than that of existing products.

Apart from hardware, much of the true cost of these systems arises from the software; development and testing plus hosting and maintaining one web application alone can be extremely costly in labour and server, or even energy costs. Due to this, it would be possible to expand the on-premise application to include all the features and functionality of the cloud web application, allowing users to host the application themselves on existing hardware. This does, however, mean greater susceptibility to data loss in the event of, say, a burglary. As such, the system would likely need to facilitate notification of external devices, in the event of detecting an intrusion. Although the security features in such an implementation may become less practical or useful, it may allow foot traffic monitoring features to become more accessible. Much of this argument would depend upon whether users prefer a cloud or on-premise solution [26]. For these reasons, it may be difficult to save on software costs relative to similar products on the market without compromising on feature offerings.

Overall, the software cost and pricing of this system may possibly be kept slightly lower than existing systems at the expense of additional features.

5.4 Future work

In future iterations of this system, it would be desirable to implement additional features for foot traffic monitoring. Currently, the web-based front end provided in the system is extremely minimal; this could be expanded to include a more comprehensive platform for users to view or visualise sensor data. For example, a heat map could be generated across the floor plan of a building to visualise how foot traffic within it varies according to the area of the building, time of day, etc. Some users may desire more advanced analytics utilising sensor data as well as other data sources Such features would require the creation of a protected endpoint granting the admin user the ability to switch between passive monitoring of foot traffic for data gathering purposes and actively creating alerts at every instance of sensor data being received; functions for this are already implemented in the back end of the application, just not exposed at any endpoint. This could also be adapted to only trigger an intrusion alert when particular sensors are tripped by maintaining a data structure containing identifiers for devices that should (or should not) trigger an alarm if their sensor is tripped. An expanded form of this system may alter the current use of alerts to include sending a message to an on-premise admin or other devices.

In terms of better security practices, it would be beneficial for a system such as this to implement protection against physical security breaches and tampering. Part of this may include a central "authority" application keeping track of which devices are currently connected and active, such that alerts can be created and login credentials revoked if any device disconnects unexpectedly. This would help prevent threat actors from obtaining access to protected endpoints in the API should an IoT device be stolen, provided login credentials for the LAN are not stored in plain text. This could potentially be implemented by forcing each IoT device to send a ping to the central authority periodically - perhaps within each loop - which would reset a timer of sorts in the central application. Lastly, more advanced Identity and Access Management practices such as multi-factor authentication, login credential rotation and role-based access control would provide further protection against attacks.

The main drawback to implementing these more advanced features is that they incur additional financial costs, which may inhibit adoption by some users. Thus, keeping future versions of this system minimal and as lightweight as possible would help provide at least some protection and insights for those inhibited by the prices of more complex systems.

CHAPTER 6

Conclusion

In this work, we have discussed issues and potential designs for implementing both physical security and foot traffic monitoring systems using IoT devices. We discussed some of the general problems faced in IoT security and potential solutions proposed in current research. We then outlined our desire to create a system that combines features from both physical security and foot traffic monitoring systems, whilst minimising the financial cost compared to similar products on the market. We developed and implemented a design comprising three layers, an IoT layer with sensors and micro-controllers, an on-premise server running a device registration gateway and a cloud server running the main web application. Having tested and evaluated this design against others, we determined that compromises could be made in both software and hardware offerings to keep costs low. Understandably, these solutions may prove far less effective in providing security and data insights. Consequently, it can be more easily observed why similar systems with more advanced features start at high prices, with businesses possibly offering greater features when subsidised with subscription fees to turn a profit. Future work in expanding this system was discussed, including the potential addition of a visualisation and analysis platform for sensor data, as well as cyber security features to improve the effectiveness and practicality of the system. Despite the drawbacks and limitations, our solution operated as intended, implementing system requirements that constitute an effective proof of concept for a minimal, low-cost, security and foot traffic monitoring system.

References

[1] Y. Lu and L. D. Xu, "Internet of things (iot) cybersecurity research: A review of current research topics," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2103–2115, 2019. DOI: 10.1109/JIOT.2018.2869847.

[2] A. Kumar, S. Dhingra, and H. Falwadiya, "Adoption of internet of things: A systematic literature review and future research agenda," *International Journal of Consumer Studies*, vol. 47, no. 6, pp. 2553–2582, 2023. DOI: https://doi.org/10.1111/ijcs.12964. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/ijcs.12964. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/ijcs.12964.

[3] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, vol. 103, pp. 194–204, 2018, ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2017.07.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804517302254.

[4] K. T. S and P. P. Nair, "Aslads: A secure lightweight authentication and data transmission scheme for smart iot devices," in *2024 16th International Conference on Communication Systems & Networks (COMSNETS)*, 2024, pp. 927–935. DOI: 10.1109/COMSNETS59351.2024.10427390.

[5] H. Garg and M. Dave, "Securing iot devices and securely connecting the dots using rest api and middleware," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2019, pp. 1–6. DOI: 10.1109/IoT-SIU.2019.8777334.

[6] A. Gupta, D. Campos, P. Ganeriwala, S. Bhattacharyya, T. Oconnor, and A. Dcosta, "Modeling internet-of-things (iot) behavior for enforcing security and privacy policies," Jun. 2023.

[7] N. I. of Standards and Technology, "Nist cyber secuity for the internet of things program," Tech. Rep., 2016. [Online]. Available: https://www.nist.gov/itl/applied-cybersecurity/nist-cybersecurity-iot-program.

[8] F. Wu, L. Xu, S. Kumari, and X. Li, "A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, Feb. 2017. DOI: `10.1007/s12652-016-0345-8`.

[9] J.-S. L. Wen-Bin Hsieh, "A robust user authentication scheme using dynamic identity in wireless sensor networks," *Wireless Personal Communications*, vol. 77, pp. 979–989, 2014, ISSN: 1572-834X. DOI: `10.1007/s11277-013-1547-4`.

[10] Ring. [Online]. Available: `https://en-uk.ring.com/`.

[11] D. Calacci, J. J. Shen, and A. Pentland, "The cop in your neighbor's doorbell: Amazon ring and the spread of participatory mass surveillance," *Proc. ACM Hum.-Comput. Interact.*, vol. 6, no. CSCW2, Nov. 2022. DOI: `10.1145/3555125`. [Online]. Available: `https://doi.org/10.1145/3555125`.

[12] Dor. [Online]. Available: `https://www.getdor.com/`.

[13] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017. DOI: `10.1109/JIOT.2017.2703172`.

[14] R. Sivapriyan, S. V. Sushmitha, K. Pooja, and N. Sakshi, "Analysis of security challenges and issues in iot enabled smart homes," in *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, 2021, pp. 1–6. DOI: `10.1109/CSITSS54238.2021.9683324`.

[15] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017. DOI: `10.1109/MIC.2017.37`.

[16] R. Venkatesh and N. Malarvizhi, "Iot security and machine learning algorithms: Survey and analysis," in *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2023, pp. 444–451. DOI: `10.1109/ICOEI56765.2023.10125626`.

[17] E. Systems. [Online]. Available: `https://www.espressif.com/en/products/modules/esp32`.

[18] Arduino. [Online]. Available: `https://www.arduino.cc/en/software`.

[19] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[20] Pallets, *Flask.* [Online]. Available: `https://github.com/pallets/flask/`.

[21] vimalloc, *Flask-jwt-extended.* [Online]. Available: `https://github.com/pallets/flask/`.

[22] M. Countryman, *Flask-bcrypt.* [Online]. Available: `https://github.com/maxcountryman/flask-bcrypt`.

[23] E. Systems, *Arduino-esp32.* [Online]. Available: `https://github.com/espressif/arduino-esp32`.

[24] *Speed of sound*. DOI: 10.1093/oi/authority.20110803100522606. [Online]. Available: https://www.oxfordreference.com/view/10.1093/oi/authority.20110803100522606.

[25] tzapu, *Wifi manager*. [Online]. Available: https://github.com/tzapu/WiFiManager.

[26] C. Fisher, “Cloud versus on-premise computing,” *American Journal of Industrial and Business Management*, vol. 08, pp. 1991–2006, Jan. 2018. DOI: 10.4236/ajibm.2018.89133.