University of Sheffield

A Low-Energy Solution for IoT-Based Smart Farms.

Stephanie D Rudd

Supervisors: Prof H Cunningham, Prof J Clark

A report submitted in fulfilment of the requirements for the degree of PhD in Advanced Computer Science

in the

Department of Computer Science

July 23, 2023

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: ______________________________

Signature: ______________________________

Date: ______________________________

Abstract

This work proposes a novel configuration of the Transport Layer Security protocol (TLS), suitable for low energy Internet of Things (IoT), applications. The motivation behind the redesign of TLS is energy consumption minimisation and sustainable farming, as exemplified by an application domain of aquaponic smart farms. The work therefore considers decentralisation of a formerly centralised security model, with a focus on reducing energy consumption for battery powered devices. The research presents a four-part investigation into the security solution, composed of a risk assessment, energy analysis of authentication and data exchange functions, and finally the design and verification of a novel consensus authorisation mechanism. The first investigation considered traditional risk-driven threat assessment, but to include energy reduction, working towards device longevity within a content-oriented framework. Since the aquaponics environments include limited but specific data exchanges, a content-oriented approach produced valuable insights into security and privacy requirements that would later be tested by implementing a variety of mechanisms available on the ESP32.

The second and third investigations featured the energy analysis of authentication and data exchange functions respectively, where the results of the risk assessment were implemented to compare the re-configurations of TLS mechanisms and domain content. Results concluded that selective confidentiality and persistent secure sessions between paired devices enabled considerable improvements for energy consumptions, and were a good reflection of the possibilities suggested by the risk assessment.

The fourth and final investigation proposed a granular authorisation design to increase the safety of access control that would otherwise be binary in TLS. The motivation was for damage mitigation from inside attacks or network faults. The approach involved an automated, hierarchy-based, decentralised network topology to reduce data duplication whilst still providing robustness beyond the vulnerability of central governance. Formal verification using model-checking indicated a safe design model, using four automated back-ends.

The research concludes that lower energy IoT solutions for the smart farm application domain are possible.

Acknowledgements

I'd first like to acknowledge my fellow lab mates, thanks for the camaraderie and support along the way - it has been great riding alongside you all on the PhD journey. To Elif Kavun and Prosanta Gope for the academic support beyond SoAS sessions - and saving me from the occasional blonde moment.

My good friends Gareth Coleman and Richard Banks - I would have been in a pickle without your technical support.

Rudds B and P, K and N - for your understanding and patience throughout the whole event.

Professor John Clark - with your scrutiny of my writing and dry humour, every meeting was insightful.

And finally Professor Hamish Cunningham - for inviting me onto the experience in the first place, and your no-bull attitude. It has been a pleasure.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and motivation

The context of this work is the control and monitoring of sustainable aquaponic agriculture systems. These systems use Internet of Things (IoT) microcontrollers to gather data, push it to the cloud and then receive and act upon control instructions.

Aquaponics produces food using 90% less water than traditional farming, and contributes towards carbon-neutral food production. An aquaponics system can include certain levels of automation to reduce labour and waste whilst ensuring a consistent farming environment for native and non-indigenous crops. Systems are also environmentally-friendly, implementing a symbiotic relationship between plants and fish to allow the elimination of fertiliser in favour of fish effluent. Aquaponics systems can also employ remote operation and utilise green energy such as solar, wind, or piezoelectric harvesting to power the requisite microcontrollers supporting sensor-actuator devices for light, heat, pH levels and water monitoring. Finally, through the use of low-energy communications protocols such as Bluetooth Low Energy (BLE), Message Queuing Telemetry Transport (MQTT), Zigbee, or meshing such as ESP-NOW, aquaponics smart-farming can benefit from operations without the internet - allowing independent and low-power off-grid crop production.

1.2 Problem statement

As a potentially widespread sustainable food production technology, aquaponics ([7; 8; 9; 10; 11]) are vulnerable to cybersecurity threats due to their reliance on microcontrollers, sensors and actuators, and many other networked components. External threats are arguably the most pertinent to the landscape, as with the globally accessible internet - users intending to exploit vulnerabilities for unauthorised access to cause damage and downtime. Such vulnerabilities include tampering and theft of data, malware to compromise system functionality or data integrity, hardware failures particularly with lacking default protection, and issues relating to data privacy and protection. Specifically with regards to a microcontroller-oriented network, the challenge is to implement protection against

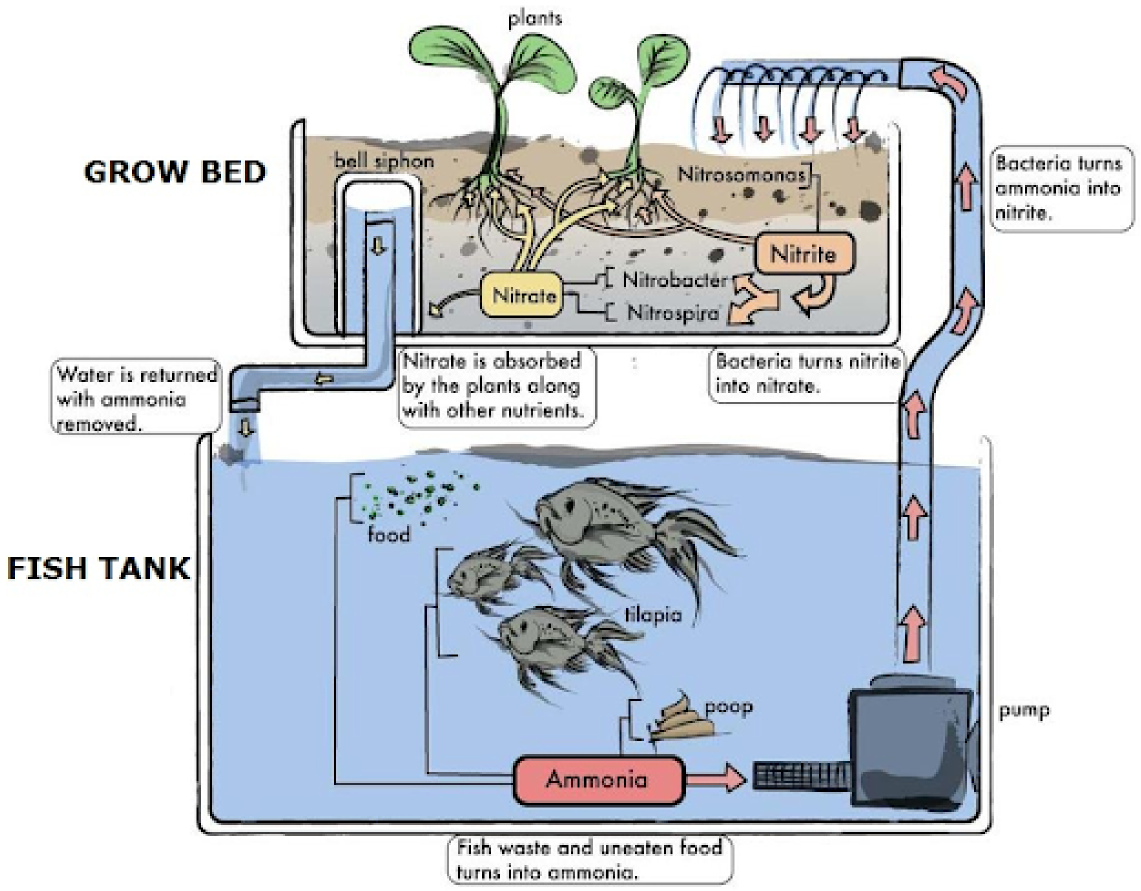
plants
GROW BED
bell siphon
Nitrosomonas
Bacteria turns ammonia into nitrite.
Nitrite
Nitrate
Nitrobacter
Nitrospira
Water is returned with ammonia removed.
Nitrate is absorbed by the plants along with other nutrients.
Bacteria turns nitrite into nitrate.
food
FISH TANK
tilapia
poop
pump
Ammonia
Fish waste and uneaten food turns into ammonia.

Figure 1.1: The Symbiosis of Aquaponics, image source [1]

such threats whilst maintaining efficient operational requirements with the relatively limited resources presented.

Whereas the general consensus refers to ‘adequate security’ as the appropriate measures to protect a system from unauthorised access, modification, or destruction, in the context of an energy-conscious security solution, this would include energy reservation. Consequently, we must ensure 'adequate' security - a trade-off between what the network requires to ensure operational continuity whilst ensuring a non-superfluous attitude, and which must be addressed as part of system design. When developing Internet of Things (IoT) applications, the resources of systems components must be taken into account. Devices in IoT systems are often highly constrained in terms of computer power, memory and available power. This affects what primary functionality can be hosted but also what security mechanisms can be deployed.

Microcontrollers are often designed to run for long periods of time on batteries or green energy - and security implementations must take account of the small amount of power and computational resources that they deploy. The de facto standard for secure networked communication, Transport Layer Security (TLS), has developed in parallel with traditional machines and their Operating Systems (OS): desktops, laptops, and more recently smart-phones - using Windows, Mac, Linux or Android, and all the complex architecture that those high powered technological developments have demanded. The more complex the computer architecture, the greater the deluge of data stored in cyberspace, and the threat landscape available, the higher the resource requirements of TLS.

IoT devices typically resemble machines of the 80s or 90s - Random Access Memory (RAM), flash memory, and processor abilities are all in short supply, and deliberately so in order to align with the minimal expectations of an embedded sensor-actuator system. These are not general-purpose devices but minimal systems optimised to their particular application. Detection of heat, light, chemicals, sound, presence, or whatever else from the sensors is only required to trigger actuator adjustment in order to condition environmental parameters using a series of short and simple messages. The question therefore arises as to why such rigorous security should be applied to these minimal devices, often operating on communications protocols away from the global threat landscape of the internet, and the answer to that is, because in its various forms, TLS is omnipresent to the point where its functions and compatibility have monopolised security application.

However, for this research, the following disadvantages associated with TLS should be considered:

1. **High latency:** TLS in itself does not necessarily introduce high latency, but the additional steps required for establishing secure connections and processing encrypted data can introduce some latency, especially in resource-constrained environments like IoT networks.

2. **Network complexity:** Implementing TLS can increase the complexity of network communications, as it requires additional protocol handshakes and data exchanges to

establish the secure channel. This added complexity may not be suitable for IoT devices, which often rely on lightweight communication protocols for seamless operation.

3. **Implementation cost:** Implementing TLS on IoT devices can be resource-intensive, as it demands additional processing power and memory. IoT devices are generally designed with low-power consumption and limited resources as part of their intended use, creating a challenge when integrating TLS without compromising the devices' efficiency and effectiveness.

4. **Man-In-The-Middle (MITM) attacks:** While TLS provides secure communication, it may still be vulnerable to certain types of attacks, such as MITM whereby an attacker intercepts and/or tampers with the communications between client and server - undermining the security provided by TLS.

However, TLS is widespread. TLS is available for almost every protocol - uTLS , BLE-TLS, MQTTS, even LoRaWAN. It is mature, strong, reliable, and has shown success in adaption to the chaotic and heterogeneous environment that is the IoT. Initially, TLS was designed to renew connections very frequently to safely communicate with multiple devices in a short period of time whilst communicating personal and sensitive data. Such frequent resets may not be necessary where relationships remain the same for long periods of time - and in the application domain of smart farming it may not handle personal or sensitive information.

Further towards the disadvantages pertaining to TLS is the concept of centralisation. The chaotic, ad-hoc, and heterogeneous nature of IoT networked devices renders any such IoT-oriented application domain an antithesis of centralised infrastructure. TLS is built on an architecture of certificates and their management. Traditional TLS relies on Certificate Authorities (CAs) to issue and manage digital certificates, creating a centralised trust model which can pose challenges in IoT environments, where devices are often widely dispersed and operate under diverse network conditions. Further details regarding the nature of CAs and the infrastructure belonging to TLS is discussed in section 2.6.

CAs can create bottlenecks and single points of failure, which may not be suitable for IoT systems that require more resilient and decentralised trust models. In this context, a distributed or decentralised architecture could be a more appropriate solution for IoT environments, as it can provide a more flexible and scalable trust model that is better suited to the unique requirements of IoT devices.

To reinvent TLS in its entirety would be impractical and would limit connectivity of any device not implementing that novel design, and so ideally, the use of existing TLS functions in a more flexible and lightweight manner would resolve the issues surrounding centralisation and energy consumption. The research challenge here involves challenging the stipulations of what TLS considers adequate security, and generating a new definition of that principle towards application-appropriate protection. Protection may include security or privacy functions - privacy functions can provide equally as effective protection and have demonstrated success in the long-standing Payment Card Industry Data Security Standard (PCI-DSS) regulations, and more recently, the General Data Protection Regulation (GDPR) directive.

The challenge of this research is to investigate, propose, and assess an alternative solution to TLS whilst retaining the mechanisms regarded as strong, reliable and ubiquitous. This research proposes a reconfiguration of existing TLS mechanisms towards a domain-specific solution, aimed at vastly reducing the energy consumption that can make TLS problematic in IoT use.

1.3 Research hypothesis

The research hypothesis investigated in this thesis is an energy-efficient IoT security solution can be created by combining a reduced variant of TLS with a low-overhead consensus-based access control.

The contribution of this thesis is to affirm this hypothesis by answering the four research questions given below. Firstly, a threshold protection framework was modelled to reduce superfluous security provisions. This energy reduction framework then formed the basis for developing and testing lightweight solutions for the three TLS mechanisms: authentication, data exchange, and an authorisation mechanism. In each case questions are framed within the chosen application domain of aquaponics smart farming, and subsequently point out more general implications for other IoT devices and networks.

The application of smart farming exemplifies a static environment in which relationships are mutually exclusive, persistent and alive for long periods of time - or however long the systems are supporting crop growth. Therefore, any IoT-oriented environment could benefit from this application, as long as it matches such criteria.

1.4 Research questions

1: How can security and energy metrics assess domain requirements?

This first question follows on from Chapter 3, Requirements Analysis, towards developing a content-driven framework for reducing energy consumption in Chapter 4. Risk-driven frameworks are used widely in the domains of security and privacy, and so referring to the impact assessment methods of well-known threat methodologies is adapted to ascertaining threshold protection. The outcome presents a framework by which security implementation is reduced as much as safely possible in order to make energy savings.

2: How much could energy consumption be reduced in device authentication?

This question is explored as the first half of Chapter 5, Energy-Conscious Authentication and Messaging, following on from the framework developed in Chapter 3. Authentication as a TLS mechanism is broken down into functions, processes and assets, and redesigned as a relatively lightweight alternative, removing the certificates required for systems using a web browser, and disregarding central governance (e.g. by a Certificate Authority (CA)) - whilst still assuring trust and authenticity of data. Results affirm that processing authentication on each microcontroller and generating additional keys considerably reduces energy consumption.

3: How much could energy consumption be reduced in payload security?

This question is explored as the second half of Chapter 5, which addresses mitigating the energy consumption of data exchange during secured sessions between devices. Message security in TLS is accomplished by a limited choice of complex and highly secure ciphersuites - ciphersuites available to the IoT domain but which would benefit the environment through reconfiguration of attributes depending on data protection requirements. By following the energy reduction framework provided in Chapter 3, the session security functions are separated and reduced to a 'threshold' protection level. Results indicate enormous energy savings whilst being application-appropriate and safe.

4: Can a lightweight, self-healing consensus mechanism be ascertained as safe?

This question is explored as Chapter 6, Energy-Conscious Consensus Mechanism, and proposes the design of a consensus mechanism similar to those used in blockchain technology, but of a novel, hierarchical and ephemeral form, with a limited data chain. This consensus mechanism is developed as an alternative to access control mechanisms which use lists, roles or rules, of user profiles. This is a decentralised model to challenge the vulnerabilities and scalability issues pertaining to centralisation, limited by small numbers of aquaponics devices, grouped using the example of allocation by BLE range.

It is offered as a granular alternative to the binary 'in or out' approach of TLS, using most of the cryptography attributes already implemented for authentication and session security. The design is model-checked under formal verification, and ascertained as safe.

1.5 Contributions to the field

The work has achieved (and published) the following results:

1. Energy-conscious risk-driven assessment considers each type of data exchange on its own merit, protection levels for benign and transparent data have been adjusted for energy reduction and without jeopardising the safety levels of the environment.

2. Through reconfiguration of the standard TLS ciphersuite to better suit IoT-oriented smart farm data exchange, it was discovered that the energy consumption attributed

to authentication and subsequent data exchange was effectively reduced to around 4% of the energy demand compared to conventional TLS practice.

3. The configuration of separate ciphersuite attributes demonstrated that domain-specific configuration of TLS could enable an energy saving of up to 81% in data transmission when compared to typical TLS practice.

4. A decentralised access control system with ephemeral data records was demonstrated as safe for the application domain, providing a scalable and limited model to mitigate use of resources.

1.6 Thesis Layout

The balance of this thesis is structured to guide the reader through the concepts and relevant literature pertaining to the study, requirements of the domain, and the design and testing of TLS-alternative solutions as follows:

Literature Review

The literature review covers the terms, definitions, existing solutions, and challenges pertinent to the research, to detect gaps in current solutions and opportunities for novel and beneficial developments.

The literature review begins with introducing the IoT landscape, smart-farming in general, and aquaponics specifically. Next the threat landscape is discussed, with reference to threats attending traditional architecture, constrained devices, and the power versus processing trade-off. Following this discussion, we consider risk-driven modelling practices by which to assess the risks to the domain application, and specifically the particular microcontroller employed.

Business modelling is also included, to widen the scope of considerations beyond but including standard security concerns, including a discussion on the differences between privacy and security to illuminate the possibilities of substituting traditional, heavy security functions with lighter, alternative privacy functions. There is overlap between the two definitions, particularly where confidentiality is concerned, and the recognised uses of such in financial data exchanges provides initial evidence of the safety of the substitution.

With knowledge of the threat landscape, we then proceed to consider existing solutions for securing constrained devices in similar environments, and a range of network topologies. This is where the problem of securing diverse network topologies arises, and in response we present examples of processing over a single node (centralisation), processing distributed across multiple nodes (decentralisation), and lastly peer-to-peer computation across equal interconnected nodes of evenly shared ownership and processing resources.

We are now in a position to investigate the security functions, common and contrasting, between our three distinct types of network topology, and under the fundamental security principles Confidentiality, Integrity and Availability (CIA). The CIA triad underlies all security practice, and includes a series of functions used from thousands of years ago to modern developments. This section presents the options available to the new solution design, with emphasis on power and processing requirements.

Finally, an analysis of TLS is presented as a benchmark for the security processes and energy consumption that the study aims to improve on. The previous sections of the literature review are included in this discussion to highlight how unsuitable the standard is for the aquaponics application, common functions available for use, and opportunities of flexibility, ultimately leading to improvement.

Aims and Objectives

This section undertakes an analysis of the security requirements to support the smart farming application domain. It uses a study of business processes and risk-driven security metrics to derive an energy reduction framework, factoring power and processing requirements in the same manner as threats. This can be undertaken by reducing, substituting, and removing security functions normally stipulated in TLS, and outputting metrics to correlate low to critical energy use based on implementation.

The requirements analysis draws from the literature review to propose research challenges and research questions. Firstly, research challenges are deduced by considering the problematic elements of TLS infrastructure when applied to IoT. Secondly, research questions are induced to resolve the research challenges to propose a four-part solution.

The resulting steps outline the experiment framework for answering the research questions over the following three chapters.

Methodology

This section draws from the aims and objectives to illustrate practical ways in which to derive measurable results from experiments one to four.

The first step of the methodology involves reducing, substituting, and removing the conventional security functions in the TLS infrastructure. Next, the study proposes to evaluate and quantify the resulting security measures' impact. It is critical to ensure that, while reducing the energy use, the security of the smart farming application domain is not compromised. For this, a carefully designed risk evaluation will be employed, which will be detailed in the next chapter.

Subsequently, the deductions of functions and proposals of the risk assessment will be used for experiments two and three - the idea is to measure the corresponding energy use based on the implemented changes and its subsequent impact on power and processing requirements.

Finally, the methodology for designing and model-checking a suitable authorisation mechanism are presented as an add-on for TLS as opposed to part of the re-configuration.

Energy-Driven Risk Assessment

This section covers research question one. Following the guidance illustrated in the literature review regarding privacy functions, we can then ascertain where appropriate function substitution could occur, and how to rank energy consumptions of each security and privacy function to produce metrics. Metrics are proposed at this stage to reflect a relative difference between attributes - strength of confidentiality (encryption), integrity (hashing), and availability (authentication). At this stage, the assessment seeks to demonstrate how trade-offs can be made between strength and energy consumptions for implementation later on. This is the reconfiguration of TLS ciphersuite parts that will ideally increase the lifespan of the application domain in terms of battery life - perhaps enabling green energy through things such as solar or wind.

The assessment begins by considering how to mitigate energy consumption by exercising impact scoring and Soft Systems Modelling (SSM). Utilising discovery techniques such as SSM, we can better understand the attributes available to quantify an assessment, including but in addition to, typical security factors.

Risk assessments operate on severity, and the aim of invoking a novel risk-driven framework is to challenge elements of the typical TLS strategy for the domain under implementation, so that the proposed solution exhibits a considerably reduced use of energy.

This constitutes a Data Protection Impact Assessment (DPIA), which occurs in the design phase. Security and privacy by design, as opposed to retrospective problem solving, enable a benchmark by which power analysis study later on can confirm or deny consumption savings reflected in the DPIA metrics. The functionality reduction process, and the results they obtain, will subsequently be challenged in Section 4, and it is argued will constitute an accurate reflection of the post-DPIA energy consumption differences.

Having considered the scope of the domain, security requirements, and the configurations proposed for the solution, we can now proceed with testing under power analysis.

Energy-Conscious Authentication and data exchange

This section includes research questions two and three, and describes the process of design and testing for energy-conscious device authentication and message protection. The research detailed in the literature review covers the standard TLS configurations for each mechanism, and our DPIA provided a framework for adjusting this configuration for energy reduction. This chapter describes the process by which the DPIA results were proven correct.

Device authentication is considered first, beginning with the challenges of authentication in general and how the design contributes. This includes optional session privacy to remove the need for superfluous encryption, decentralised processing to alleviate infrastructure requirements, persistent sessions to take advantage of the stable and long-term relationships between devices, and, finally, low energy consumption as a result of the collective contributions. It is low energy consumption that the study aims for, but previous contributions highlight a number of additional benefits. TLS authentication is then described, along with its problematic aspects, and the literature review is referred to, describing how the expectations of constrained devices, centralised governance and domain application render TLS inappropriate.

Existing solutions for energy-conscious authentication are then introduced as self-signed certificates and certificate-less security are considered towards developing a novel solution. Authentication functions available to the domain microcontroller are presented and proposed, and presented as an implementation proposal for power analysis.

Message payload security is now considered, similarly presenting topic challenges and contributions. As with authentication, privacy driven by content and domain requirement indicates potential for considerably lower energy consumption than TLS offers. The problem statement for messaging describes ciphersuites and how unnecessary the strength and encryption within them can often be, but also how the stringency of TLS offers little flexibility to ciphersuites.

Existing solutions to TLS alternatives are limited, but feature energy harvesting for increasing power resources, mitigating message attributes and edge processing, as opposed to using a server or hub. In response, our solution proposes a distributed infrastructure, additional keys to allow for selective encryption, and ciphersuites based on the requirement of delivery in order to assess threshold strength.

Power analysis as a testing methodology is presented through Ohm's law, and the techniques to measure time and power are illustrated for obtaining accurate energy consumptions. Experimental test results are then presented and discussed, demonstrating a positive outcome of the DPIA in Chapter 3, in which both novel designs consume significantly less energy than regular TLS employment.

Energy-Conscious Consensus Mechanism

This section presents research question four, as the design and safety check of a consensus mechanism - a decentralised and ephemeral authentication mechanism based on a limited data chain. This is for granular access control, a more selective approach than TLS based on hierarchy and roles, with the purpose of domain assurance regarding system integrity and device entry approved by the existing group. We introduce this chapter with reference to device capabilities in context of typical authorisation applications; centralised lists of correlating rules and roles amongst databases, and how this contradicts the dispersed, chaotic and heterogeneous nature of IoT. Our contribution is the proposal of a distributed resolution by a process to distinguish between genuine and false network devices.

TLS access control is binary and operates on an approach of 'either there is access to the network, or there is not'. Typical use of TLS would not present this as a problem, because the relationships between client and server are deliberately brief, and manifold. However, the nature of a smart farm application could benefit from long-term and persistent relationships, wherein insider or persistent threats may be more of a concern. As such, additional protection is considered in a decentralised, scalable manner, using the consensus of neighbouring devices.

With knowledge of what the domain would benefit from, the design provides a low-energy approach by recycling the functions existing already from device authentication and message security, and limiting these attributes into clusters of systems within a predefined range. Limiting the number of systems governed by a distributed mechanism ensures the chain is short, overlap is limited, and repetition of attributes between clusters is high enough to assure integrity and backup domain-wide. A shallow hierarchy with ephemeral roles is proposed, so that replacement devices do not extend the chain to prevent scalability issues. The mechanism intends to be agnostic, and the functions employed in the example are representative of any security application in which the same can be recycled from authentication and messaging. Finally, we propose that a consensus mechanism is the most appropriate because from periodic integrity checking, a number of existing devices can grant access, forming autonomous self-healing.

With emphasis on low-energy already covered, the model is then checked for safety and secrecy. Formal verification is used for model-checking, and demonstrates safety by secrecy.

Conclusion and Further Work

The research is concluded by referring back to the contributions provided by experiments one to four, and how they create an energy-conscious alternative to TLS as a whole. The limitations of the study are discussed, and future research topics are proposed beyond the scope for this research.

1.7 Prior publication

- Low-Energy Authentication with Selective Privacy for Heterogeneous IoT Devices in Smart-Farms [12].
- Selective Privacy in IoT Smart-Farms for Battery-Powered Device Longevity [13].
- Towards Lightweight Authorisation of IoT-Oriented Smart-Farms using a Self-Healing Consensus Mechanism [14].
- Threat Modelling with the GDPR towards a Security and Privacy Metrics Framework for IoT Smart-Farm Application [15].

Chapter 2

Literature Review

The literature review covers the background problems and related work pertaining to the proposed security solution, and begins with an introduction to the application domain, the IoT landscape, smart farming as an IoT application, aquaponics specifically, and the microcontroller we will be studying. We then move onto the threat landscape, security vulnerabilities in general, the trade-off between power and processing in IoT applications, and the significance of risk-driven threat modelling. Following on from the threat landscape is an exploration into design modelling for the proposal - from a business perspective, the differences between privacy and security, and the applications for each in implementation, and finally how metrics can be objectively scored as part of risk assessment. The next section presents existing security solutions to centralised, decentralised, and distributed security - invoking concepts for the proposal. The next section considers security mechanisms as the fundamentals of security practice and implementations for confidentiality, integrity and availability on the microcontroller. Finally, we discuss the de facto standard of internet security (TLS), including the mechanisms and differences between the first and second version of the same protocol.

2.1 Application domain

This section firstly presents the IoT landscape as a concept, the nature of constrained and heterogeneous devices, and general characteristics in application. Secondly, smart farming as an industrial application of IoT is discussed, summarising the benefits compared with traditional farming and discussing the networking topologies of existing smart agriculture. Aquaponics is then presented as the specific application for this research, with attributes of the domain that will affect the design of our security framework. Finally, the featured microcontroller is presented, since it is necessary to understand the nature of hardware and firmware before efficient, resource-centric security practices can be fulfilled.

2.1.1 The IoT landscape

Whilst the web was developed to share information generated by people, the IoT operates on data created by things [16], and to facilitate interactions with and between those things. The IoT is pervasive by nature [17], enabling device-to-device and human-to-device interaction. Devices will be sensors and actuators, usually hosted by microcontrollers and (gateway or hub) microcomputers.

Despite implementation in large-scale applications, technical specifications or references for architecture are far from being standardised [18]. IoT devices use the internet as fundamental infrastructure, but also may connect autonomously with each other [19]. This diverse networking nature of communication means IoT connections become unpredictable, spontaneous and difficult to anticipate [20]. A network nature that is chaotic and amongst myriad heterogeneous devices, perhaps dispersed over many miles by various protocols, presents quite a topological challenge - at times devices cannot be catalogued and controlled in a traditional sense.

Threat reduction and reassurance of safety is paramount to the future of IoT applications, which are increasingly supporting the monitoring and control of daily environments [21]. With concerns escalating over security, privacy and trust [22], industrial IoT employment amongst sectors such as healthcare, military and agriculture, is of concern - as security is neither straightforward nor standardised. Heterogeneous connected devices can appear fragile and vulnerable, and the IoT landscape is a relatively immature approach to data management and security. On the other hand there are benefits to this simple nature. Microcontrollers are different from other modern technologies such as smartphones or laptops, but very comparable to the standards of the 80s and 90s in terms of RAM and flash memory. They are not designed to undertake the processes of desktop machines, and the physical size and native protocols reflect that in an obvious way.

Constrained devices are often devoid of an Operating System (OS), at least in the sense of time sharing, process management and the like), accessible mass storage, user accounts, a browser and so on, and this absence of functionality removes numerous vulnerabilities. To offer further protection against global threats, microcontrollers operate on a range of protocols, such as Bluetooth Low Energy (BLE), Long Range Wireless Area Network (LoRaWAN), Message Queuing Telemetry Transport (MQTT), Radio Frequency (RF), as common examples. Since IoT systems are highly constrained in power, processing and storage, good practice suggests that the aforementioned protocols be used in preference to the much more power-demanding internet. By removing the internet as the core communication within an IoT environment, the threat landscape is considerably reduced.

Smart farming is now discussed as the domain of industrial application for the research. A range of smart farming technologies and practices are presented, and we consider network topologies and security strategies appropriate to accommodate the nature of semi-automatic crop production.

2.1.2 Smart farming

Envisioned as a clean alternative to resource-hungry traditional agriculture, smart agriculture has been highlighted as playing an important role in efficient food production [23]. With the population at eight billion now, and expected to increase to nearly ten billion by 2050 [24], food supplies have become a major concern [25]. Smart agriculture promotes the concept of semi-automated systems for monitoring and controlling farm environments - maximising farm productivity whilst reducing input resources wherever possible.

Green energy plays a key role in smart agriculture. In addition to microcontrollers and microcomputers as lightweight alternatives to desktop machines, IoT networked sensor-actuators can operate for years on battery operation or indefinitely on green energies. Notable progress has been contributed in the use of wind turbines for agriculture [26], solar [27], and piezoelectric utilising movement to charge super-capacitors. These all find application in self-powered IoT [28]. Furthermore, amplification of the seasonally and geographically restricted energy generation of solar power has undergone substantial development using fluorescent reflectors [29]. In summary, the reduction of carbon dioxide emissions by green efforts has recently been so significant, that indoor vertical farming has been able to reduce ventilation energy consumption by 58% on top of the carbon-neutral input for the plant production itself [30], demonstrating a step towards carbon-negativity.

The social benefits of smart agriculture have demonstrated local food production and lower gas emissions towards reduction of pollution, in addition to jobs creation [31]. Crops with specific characteristics may be particularly beneficial for some people. For example, low-potassium kale can benefit those suffering from renal dysfunction [32]. Tightly-controlled nutrient solutions can facilitate the development of such crops. There are also active campaigns to support the reduction in red meat consumption in favour of plant-based food and proteins, to reduce the land mass and other resources required for livestock farming [33].

Smart farming innovations have witnessed developments in Machine Learning (ML) [34], animal behaviour and analysis for dairy farming [35], Precision Agriculture (PA) in soil-less greenhouse recirculation of saline water [36], vertical hydroponics [37], and aquaponics.

Now we turn to aquaponics, the smart farm application of this study. We present the context and motivation for energy reduction in the security of this semi-automated crop production environment.

2.1.3 Aquaponics

Aquaponics is considered as one of the most notable smart farming solutions for environmentally-efficient food production, using green energy towards self-sustainability, a soil-less environment, and nutrient-rich water circulation [27]. Systems have been modelled using IoT microcontrollers, and lightweight protocols aligned with the nature of this type of networking [38], with many examples of Artificial Intelligence (AI), [39] ML [40], and wireless sensing [41]. In general, there is a large coverage of the aquaponics domain in terms of evidential benefit, operation, and viability towards efforts for carbon reduction. There is very little coverage, however, on appropriate security applications.

Aquaponics presents a symbiotic relationship between plants and fish, where fertiliser and other synthetic nutrients are substituted by fish effluent in a continuous pump cycle around fish tanks and soil-less crop growth beds [42]. Aquaponics is similar to hydroponics but with the inclusion of fish, and the exclusion of artificial growth hormones towards an environmentally beneficial alternative [43]. In addition to symbiosis, setups can incorporate aspect-specific lighting, humidity and heating, for non-indigenous crops, regardless of season or geography. With influences of green power such as solar or wind, and appropriate expectations of the system for power and processing of the IoT components, aquaponics could exemplify a carbon-neutral, or even carbon-negative model for the future of food production.

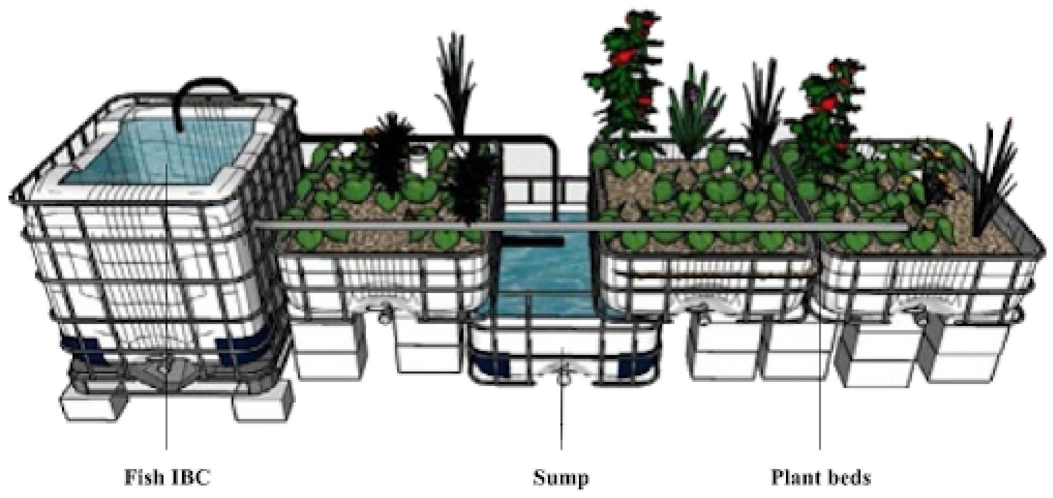

Figure 2.1: Typical Aquaponics System, image source [2]

The relevance of this study is for promoting a more viable use of aquaponics for smart farming, by establishing appropriate security. We understand that aquaponics is at the forefront of semi-automated food production, but we also believe that security is neither widely covered, nor optimised for this domain - yet it is imperative to the protection and continuity of any IoT network. Without security, failure and downtime cannot be assured against, yet implementation of inappropriate security can pose a great threat to longevity of battery and green-powered devices by draining resources. Typical security application is generic and restricted by configuration, rendering it heavy, and demanding of power, time, and processing - three things that IoT in general cannot supply. Should the aquaponics farm be moved off-grid to demonstrate further independence, use of the internet would also be disregarded, and along with it, the most common, mature applications of security. Implementers of network security therefore face a dilemma. They can either alleviate the farm of security, rendering it vulnerable to attack and downtime, or they can implement a series of unoptimised, generic security solutions for each protocol, jeopardising the performance of constrained devices.

In order to provide a comprehensive understanding of the broader context within which this study is situated, it is important to present a commonly used network topology in the field of IoT, which also forms the basis of this project - edge processing with a hub for cloud delivery. Despite this being a structural component of our project, it is included here in the literature review as it is a commonly accepted and widely used topology within the IoT domain. As a typical model for IoT networks, it is a solid foundation upon which the discussions and analyses of is study are built.

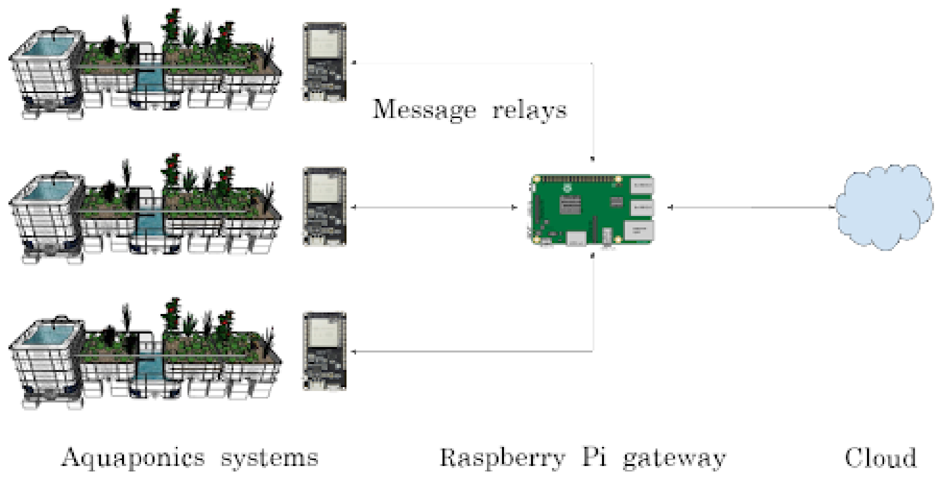

Figure 2.2: Networking the Systems, image sources [3; 4]

In summary, to optimise the lifespan of constrained devices, freedom of farm placement, and ultimately reliability and continuity of the farm, security is currently a paradox, both essential and detrimental. With these restrictions in mind, Espressif's ESP32 Huzzah! Feather board was selected to model the challenge and resolution of a prospective security solution. We now introduce the ESP32, and discuss its benefits, limitations, and how it will help address the challenge of an appropriate security solution.

2.1.4 ESP32

The ESP32 is a dual-core, low-power microcontroller with integrated WiFi, Bluetooth, and Bluetooth Low Energy (BLE). It also hosts radio transmission, several peripheral interfaces, and most importantly, four hardware modules dedicated to cryptographic hardware acceleration, making the ESP32 a natural candidate for exemplifying the ideal security solution in an IoT environment.

Below is a function block diagram for the ESP2:

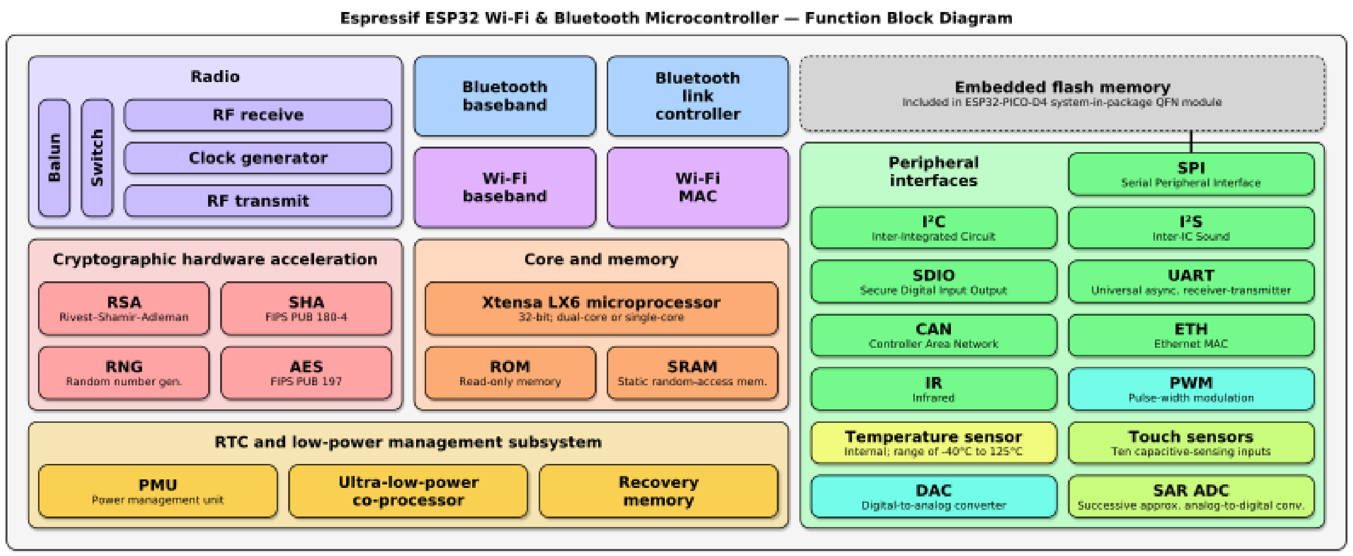
Espressif ESP32 Wi-Fi & Bluetooth Microcontroller — Function Block Diagram
Radio
Balun
Switch
RF receive
Clock generator
RF transmit
Bluetooth baseband
Bluetooth link controller
Wi-Fi baseband
Wi-Fi MAC
Embedded flash memory
Included in ESP32-PICO-D4 system-in-package QFN module
Peripheral interfaces
SPI
Serial Peripheral Interface
I²C
Inter-Integrated Circuit
I²S
Inter-IC Sound
SDIO
Secure Digital Input Output
UART
Universal async. receiver-transmitter
CAN
Controller Area Network
ETH
Ethernet MAC
IR
Infrared
PWM
Pulse-width modulation
Temperature sensor
Internal; range of -40°C to 125°C
Touch sensors
Ten capacitive-sensing inputs
DAC
Digital-to-analog converter
SAR ADC
Successive approx. analog-to-digital conv.
Cryptographic hardware acceleration
RSA
Rivest-Shamir-Adleman
SHA
FIPS PUB 180-4
RNG
Random number gen.
AES
FIPS PUB 197
Core and memory
Xtensa LX6 microprocessor
32-bit; dual-core or single-core
ROM
Read-only memory
SRAM
Static random-access mem.
RTC and low-power management subsystem
PMU
Power management unit
Ultra-low-power co-processor
Recovery memory

Figure 2.3: ESP32 Block Diagram, image source [5]

ESP32 specifications are detailed by Espressif [5], but the following are relevant for this research:

- Xtensa LX6 microprocessor: low-power, 32-bit, with C/C++ compiler, encouraging work at this low-level for most applications.
- Static Random Access Memory (SRAM): 520KB, very limited flash memory for data.
- Cryptography block: Dedicated hardware acceleration of four security-centric functions:
  - RSA: Asymmetric key generation, encryption and decryption.
  - AES: Symmetric key generation, encryption and decryption.
  - SHA: One-way hash function generation.
  - RNG: Generation of large, random numbers.
- Security features: For preventing remote access to the board:
- Secure boot: Unique key generation on startup to prevent untrusted firmware.
- Flash encryption: Encryption of the memory using the boot key.

We can see from the block diagram that a security solution should employ a small amount of instruction to reduce its occupation in the limited flash memory, built in C or C++, and founded on the four functions hosted on the cryptography block, since these are hardware-accelerated.

In this application, the ESP32 is hosted by Espressif's Huzzah! Feather board [44], a microcontroller supporting a variety of lightweight communications protocols; BLE, MQTT, ESP-MESH, RF, and TCP/IP for the internet. The board is also written to using a micro USB cable, and a 3.3V power input. The ESP32 has been used frequently in research of smart farming applications, including poultry [45], crop monitoring [46], and irrigation [47]. With a good success rate in farm operations and dedicated security functions, the ESP32 was considered a sensible choice to demonstrate a security model utilising distributed, edge-processed cryptography to reduce security infrastructure and energy consumption.

2.2 Not every cloud has a silver lining

With reference to the idiom "every cloud has a silver lining", or that every sad or unpleasant situation has a positive side to it - the case is not necessarily so with regards to cloud computing, and herein explores the attack landscape pertaining to that and IoT.

The last section described the hardware chosen for the domain, the ESP32, and the assets of the hardware - cryptography functions RSA, AES, SHA, and RNG, with storage and processing limitations briefly highlighted. This chapter explores the vulnerabilities of the IoT threat landscape in general, and the power versus processing trade-off, which in the IoT domain is a problem - enough reason to exclude security in order to preserve the lifespan of battery-powered devices.

By proposing a security solution oriented on low energy consumption, we intend to increase viability for inclusion of security at the design stage of smart farms. Where higher computer processing for stronger keys requires greater power and longer computation time, security must be reduced as much as safely possible, and so a risk-based approach is necessary to ascertain a level of security that is adequate, but not superfluous to, 'threshold' or adequate smart farm protection. This section explores the fine balance between necessary security measures, and reduction of computation within security mechanisms.

2.2.1 Security threats

IoT networks by nature are chaotic, constrained, and heterogeneous. Such characteristics indicate the strengths of such networks which are quite the opposite of desktops, laptops and modern smartphones, which can accommodate expansive processors, storage, and run on mains power, or at least are expected to have regular access to mains power. However, the natural limitations of IoT microcontrollers offer natural protection, because they have a relatively simple interface, and as a result, are much less accessible. Microcontrollers lack the vulnerability landscape of a desktop computer in a similar way that a calculator does compared against a smartphone - the less accessibility available, the more robust the device is.

While the ESP32 can optionally run a real-time Operating System (OS), like FreeRTOS, it's important to note that this is not a conventional OS, and certainly does not contain a web browser. Yet, despite this difference, security remains a critical concern.

It is widely recognised that any device connected to the internet can potentially be a target of various security threats. Even though traditional threats [48], such as SQL injection [49], CSRF, and XSS [50], are primarily associated with systems that have an OS and a web browser - the IoT devices powered by ESP32 are not immune to security vulnerabilities. The attack surface may be different, but it's equally crucial to implement security measures to protect these devices. In this sense, the type of OS available to the ESP32 refers to a call for attention towards potential vulnerabilities and the necessity of additional hardening to enhance the security posture of devices within the IoT application framework - such as vulnerabilities particular to IoT devices, for example, default passwords and usernames. Perhaps such reliance on any user's knowledge of poor default security is a little optimistic, and there is basic need for good device authentication and data transfer. There is no requirement for browser TLS when dedicated hardware acceleration and library calls provide the same cryptography functions pertaining to Confidentiality, Integrity and Availability (CIA) of data. These are, on the ESP32 crypto block, provided by encryption or confidentiality as AES, hashing or integrity as SHA, and authentication or availability as RSA - and a Random Number Generator (RNG), to assure of unpredictable computation behind the aforementioned functions. These functions provide processing straight from the ESP32 hardware - enabling the same configuration as browser TLS, in addition to many other options for configuration as well.

The Serial Peripheral Interface Flash File System (SPIFFS) [51], and Mongoose OS [52], for example, pose unreasonable expectations on the underlying architecture. Attempting to transform machines comparable with those of the 1990's towards modern expectations imposes a heavy toll on the storage and energy resources of such constrained devices.

In 2021 the top ten threats of IoT security according to the Open Web Application Security Project [53] were as follows:

1. **Weak, guessable, or hard-coded passwords:** including default passwords, regular words vulnerable to attacks that run through the contents of a dictionary, or with too few characters. Weak passwords are liable to brute-force attacks and render encryption useless. Devices need configuring individually [54], preferably disregarding the predictable default passwords and replacing them with random strings including special characters.

2. **Insecure network services:** unnecessary or insecure network services running on the device such as remote access, media sharing or unprotected visibility. Services should be run on the basis of necessity, to not challenge the CIA triad, or allow unauthorised remote control. Service manipulation of IoT smart city data has been exploited by malicious nodes utilising services [55], through open ports [56], and the Mirai attack could be engineered from a traditional DDoS to a service-oriented attack [57].

3. **Insecure Ecosystem Interfaces:** are not on the IoT devices but connected as separate entities within the greater environment, the network, the data processing

unit, lacking access control, and weak or no encryption. Often insecure web interfaces introduce traditional threats such as SQLi or CSRF and XSS . Segregation of network layers between IoT and IoT support could prevent a lot of the browser access, where global internet access could be read-only by authorisation system, for example. The general census once again advises that layers in the IoT ecosystem should be protected by CIA mechanisms, authentication and authorisation [58].

4. **Lack of Secure Update Mechanism:** is a lack of available backups, and the secure delivery of firmware updates and validation [59]. Routine backups for business continuity could partly resolve this, as well as encrypted updates, which again is a case of fulfilling the CIA triad at either point of transit. Without encryption, updates may reveal credentials (e.g. passwords broadcast in the clear over wifi) and network instructions (e.g. code specific to a smart farm sensor's operation). Without integrity, the material can be tampered with in transit, and without availability, there is no proof of where it came from, such as a malicious node.

5. **Use of Insecure or Outdated Components :** outdated components and libraries can provide the easiest routes to exploitation. The fewer libraries and components, the better in this respect, and a good example of a 'by necessity' approach. OS platforms and lots of third-party software can create a vulnerable network. Shellshock was an example in which the outdated version of SSL allowed the DDoS botnet [60], and the superseding Linux-based DDoS IoT botnet, Bashlite [61].

6. **Insufficient Privacy Protection:** the 2018 stipulation of the General Data Protection Regulation (GDPR) [62], accelerated the use of privacy functions to anonymise, pseudonymize, and tokenize data. The personal information of living subjects left unprotected in storage or in transit around IoT environments was becoming increasingly concerning, particularly with health data, and so functions were introduced that were lighter than encryption. The Payment Card Industry Data Security Standard (PCI-DSS) [63], has employed tokenization for years - GDPR regulations use the same standards.

7. **Insecure Data Transfer and Storage:** this issue is about lacking access control (authorisation), for sensitive data anywhere in the IoT ecosystem, and lacking encryption at rest, in transit, or during processing. With data in transit, TLS can be implemented for most protocols, but there is an absence of widely accepted protection for data at rest and countermeasures that would enhance security and privacy at the design phase [64]. A ubiquitous IoT security application for data at rest would be complex and perhaps jeopardise the lifespan of devices in the same way as regular TLS, as a high-strength, energy-hungry solution. This is where risk-driven threat modelling helps ascertain the sensitive data and which areas of the environment need the most protection - towards reducing energy consumption with a 'by necessity' protection approach.

8. **Lack of Device Management:** this is the lack of default security deployed in production, including asset and update management, monitoring and threat response. Manufacturers do not supply software for these processes, but there is a large open source community supporting the hobbyist and industrial IoT security domain. The ESP32 has an Over The Air (OTA), update tool, with several versions available in the Arduino IDE [65]. It is also possible to manually backup the firmware, or set these actions to timer - the ESP32 has had excellent impact in healthcare with such reliable features [66]. An advantage to the open source style of IoT programming is that even though proprietary solutions do exist, we do not have to enter into a contract with a manufacturer that would cost money and risk privacy.

9. **Insecure Default Settings:** many IoT devices are deployed with default passwords that are vulnerable to dictionary attacks, if there are security settings at all. The ESP32 however has a few security functions; secure boot, EPROM, flash encryption, and the dedicated, hardware-accelerated cryptography functions as mentioned in section 2.1, Application Domain. Default settings aside, this should not deter us from resetting credentials which may be weak, duplicated on many devices, or stored on a database somewhere.

10. **Lack of Physical Hardening:** is the weakness in physical systems for their contents to be gained by proximity, magnetic fields, USB ports, and so forth. Malicious nodes can be attached to devices for immediate access, or persistently for remote control at a later date. Although the ESP32 has been subject to physical injection attacks [67], or the Fatal Fury attack [68], the secure bootloader key generated on startup is unique to each device, so each victim device must be physically accessed via its own System-on-Chip (SoC). Physical attacks can be protected against by additional authentication and integrity onto the device, as demonstrated with 1-wire Electrical Erasable Programmable Read-Only Memory (EEPROM) against side channel attempts [69]. This method uses one of the functions hosted on the ESP32, SHA hash function for one-way integrity checking, and authentication function Hash-keyed Message Authentication Code (HMAC), available in the firmware library to support the hardware function. The functions are explored later on, in section 2.5, The CIA. Other countermeasures include Electro-Magnetic (EM) proof coatings, casing shields, and fault detection methods to report clock distortion.

2.2.2 Power versus processing

The topic of power versus processing as a trade-off renders security implementation quite a contradiction to the nature of IoT. Energy is the major strand here, but the style of traditional security is also a paradox to the heterogeneous and spontaneous nature that is the IoT. TLS is a good example of this problem:

1. **TLS is centralised:** server-based central networks limit the range of the environment,

or require multiple servers. IoT is very dispersed and often the sensors are placed in awkward places, like tall building facades.

2. **High-spec ciphersuites:** TLS is very strong, and IoT networks do not favour high computation. The key strengths are hundreds of thousands of years strong - unnecessary for purpose.

3. **Rigid in its requirements:** mechanisms and functions are present all of the time without option for configuration adjustment. Functions are stipulated when they are not necessary.

4. **Channel-based not payload-based:** protection is provided by protocol, not content. This means that a separate TLS configuration will require setting up for TCP/IP, BLE, MQTT, LoRaWAN and so forth. This means multiple libraries and duplication of instructions on the device.

5. **Uses certificates:** X.509 certificates hold a lot of data that IoT devices do not need because they largely do not use browsers. Certificates also require a lot of storage for their chains containing roots, branches and leaves - becoming a storage and server burden.

6. **Session timeouts:** The secure sessions reset almost immediately after data transmission, even if the relationship between two devices is persistent. This means that for every single message from one device to another, the session undertakes setup, transmission and closure, perhaps every five minutes. This is a hugely unnecessary waste of processing, power and time.

7. **Complex architecture:** The setup requires an underlying governance architecture known as Public Key Infrastructure (PKI), which is both centralised and requires separate processes for key and certificate management such as revocation and escrow. If a single entity within this structure fails, so does the entire certificate system.

Despite these incompatibilities, TLS is widely accepted, mature, regulated, strong, and most importantly, ubiquitous - which means the functions underpinning TLS (SHA, AES, RSA), are the same ones available on constrained devices, such as the ESP32. A suitable solution could use the same functions in a mechanism suitable for IoT, whilst addressing the seven problem areas outlined above. An in-depth analysis of TLS is discussed in section 2.6, Transport Layer Security (TLS). Meanwhile, we consider how to safely reduce the uage of security functions by assessing the threat landscape.

2.2.3 Risk-driven threat modelling

As part of a trade-off between risk mitigation and energy consumption, the purpose of risk-driven threat modelling is to scope the risk landscape, and address only what is necessary in order to reduce the security provisions down to a threshold safety level. As with reference to

OWASP Top Ten IoT Threats item '6. Insufficient Privacy Protection' above, and following the notion of PCI-DSS in general, substitution of security functions with privacy functions have been exercised for a long time in the realm of internet safety. Taking the example of established practice, the risk-driven threat assessment seeks to incorporate privacy functions as a lightweight alternative to encryption wherever possible. We begin with considering existing risk-driven methodologies, what they consider vulnerable, and how to quantify risk.

There are several risk-driven methodologies for security at the systems design phase:

- STRIDE is the most mature [70], and provides threat-specific variants to arrange property violations identifiable in Data Flow Diagrams (DFDs), of the system design. An immediate problem with STRIDE in the smart farm domain is the presumption of specific vulnerabilities according to the top ten, which in a heterogeneous environment, may not be applicable.

- PASTA [71], describes a seven-stage model of activities to combine business objectives and technical requirements. It is attacker-centric, considers governance, architecture and operations, and may reflect the concerns of the domain better than STRIDE.

- LINDDUN [72], largely concerns privacy, with a systematic approach influenced by DFD-illustrated entities, data stores, processes and data flows. With reference to the aforementioned overlap between privacy and security, LINDDUN will be a valuable approach.

- CVSS [73], provides a numerical score of a vulnerability based on its characteristics. The CVSS consists of three metrics groups (base, temporal and environmental), and is often used in conjunction with other threat-modelling methods. CVSS is not useful for the smart farm security application design, but could be as part of a penetration test following implementation towards remediation. It will not be used in this research as we should focus on design phase security and privacy.

- Persona non Grata (PnG) [74], considers the motivations and skills of human attackers by characterising attackers as archetypes for system misuse towards specific goals. Use case scenarios are generally helpful in system design, and so this will be a beneficial security application of such scenarios.

- TRIKE [75], was created as a security audit framework from a defensive perspective, understanding actors, assets, intended actions and rules based on a DFD, and then dividing each one into one of two categories - elevation of privilege or denial of service. Since this framework assumes that users are very active in the system, it is irrelevant - the aquaponics system aims for as few active users as possible towards semi automation.

As part of a larger design, STRIDE suggests the use of DFDs, but this is irrelevant to a TLS re-model using a limited set of security functions. PASTA supports the use of business objectives and technical requirements, which will be useful to investigate, and

LINDDUN demonstrates the significance of privacy in addition to security. The CVSS scoring system, although a useful reference, is a retrospective record of successful exploits, rather than preventative methods. PnG, similarly to PASTA, emphasises the motivations of human attackers and introduces social engineering as a profiling tool. The significance of human emotions means that attacks are driven by the content an attacker is pursuing, and this would suggest a risk assessment should be content-oriented. TRIKE again seeks to incorporate the varying actors and actions they must take against the system in pursuit of goals. In summary, the human aspect to security engineering is featured in the most popular risk-driven methodologies, as is privacy and metrics scoring. These properties will be considered in the next section.

2.3 Design modelling

The previous section discussed threats pertinent to IoT, the power and processing tradeoff, and risk-driven methodologies to assess risk severity towards implementing appropriate security solutions. The discussions were focused on security specifically, generally resolved by applying the CIA principles fundamental to cryptography. However, a universal application of the CIA would not resolve the challenges of employing TLS for IoT ecosystems, nor would they reduce energy consumption enough to make such a prospective solution a justifiable alternative to TLS. As a result, a suitable security solution should consider a wider scope, by reflecting on threats and influences including, but not limited to, security.

Scoping from a business perspective is critical to the performance of an energy-centric security solution. Where security refers to levels of hardiness of the system, it does not examine environmental impacts which may affect energy reduction, or at least not beyond the CIA principles. Designing a security solution towards a green aim, and viable alternative to TLS is both a business and security proposition.

We begin with an introduction of Soft Systems Methodology (SSM), and the available methods to ascertain the security objectives required for a design using techniques for scoping the project and from surrounding influences. We then consider the differences between privacy and security functions, and the impacts of each, on the solution design. Finally, we consider how to score protection and energy consumptions using objective metrics systems for a quantifiable assessment process addressing both risk and energy cost.

2.3.1 SSM from a business perspective

Soft Systems Methodology (SSM), [76], was developed to analyse problems where there may be divergent views about the definition of a problem. These are considered ‘soft’ problems, and focus on the wheres, whens hows and whys. SSM is a long-standing approach to systems modelling and has been included as a research consideration for IoT [77], multiple criteria decision making [78], analysis of information security in banking [79], cyber security defence [80], decentralised security design [81], analysis of security breaches [82], enterprise planning for security [83], and to exploit web application vulnerabilities, including the infamous Heartbleed [84].

1. **The development of the SSM technique**

   SSM has evolved from four previous practices into one inclusive technique. These were:

   (a) **Blocks and Arrows:** focus is implementing change rather than introducing or improving a system, through a sequence of stages which iterated to previous stages:
      i. Analysis
      ii. Root definition of relevant systems
      iii. Conceptualisation
      iv. Comparison and definition of changes
      v. Selection of change to implement
      vi. Design of change and implementation
      vii. Appraisal

   (b) **Seven Stages:** A cluster of seven activities in a circular process:
      i. Enter situation considered problematic
      ii. Express the problem situation
      iii. Formulate root definitions of relevant systems and in root definitions
      iv. Build conceptual models of the systems named in the root definitions
      v. Compare models with real-world situations
      vi. Define changes both possible and feasible
      vii. Take action to improve the problem situation

   (c) **Two Streams:** A recognition of the role of history in human affairs. This method includes logic by activity modelling, and culture and politics to enable judgements to be made between conflicting interests by people concerned, enabling actions to be taken.

   (d) **Four Main Activities:** Iconic rather than descriptive approach, invoking the cultural stream of analysis by four activities:
      i. Finding out a problem situation including cultural and political influence
      ii. Formulating some relevant and purposeful activity models
      iii. Debating the situation using the models, by firstly positive prospective changes both desirable and feasible, and secondly the compromises between conflicting interests to enable actionable changes.
      iv. Taking action to bring improvement.

2. **CATWOE**

   The latest development of the SSM approach is a technique which provides a framework for defining and analysing business stakeholder perspectives towards a change of current situation. Customer, Actor, Transformation, Worldview, Owner and Environment (CATWOE). Each part of the mnemonic is described below:

- **Customers:** Who are the beneficiaries of the business process, and how does the issue affect them?
- **Actors:** Who are the people involved in the situation who will undertake the work?
- **Transformation:** What is the transformation of better inputs to outputs that this analysis is based on?
- **Worldview:** What is the big picture, and what are the wider impacts of stakeholder beliefs?
- **Owner:** Who owns the situation under analysis, and what role, positive or negative, do they play?
- **Environmental:** What are the constraints, rules and policies that will impact solution success?

3. **BATWOVE**

   Beneficiaries, Actors, Worldview, Owners, Victims and Environmental (BATWOVE), is the extension of CATWOE to include victims; in which every intervention or transformation can be of benefit or detriment to ideas and people.

4. **SWOT and TOWS**

   Strengths, Weaknesses, Opportunities and Threats (SWOT), is a classic business strategy tool and TOWS is a variant of the same acronym to extend SWOT from an internal environment to an external one (Javaid, 2021). The aim of this model is to draw domain-specific attributes such as communication ranges and architectural details, which will impact the attributes of the metrics framework later on.

   SWOT and TOWS prompts the following attributes:

   - **SO:** Strengths to maximise opportunities, or “maxi-maxi”.
   - **ST:** Strengths to minimise threats, or “maxi-mini”.
   - **WO:** Minimise weaknesses by taking advantage of opportunities, or “mini-maxi”.
   - **WT:** Minimise weaknesses to avoid threats, or “mini-mini”.

2.3.2 Privacy and security

The opportunity to reduce energy consumption of security functions is bolstered by the possibility of some privacy functions being interchangeable.

Whereas security definitions CIA relate to encryption, attestation of non-tampering, and proof of origin respectively, privacy definitions refer to the rights of a living subject. Article 5 of the GDPR defines seven core principles for lawful handling of personal information:

1. **Lawfulness, fairness and transparency:** To identify valid grounds for collecting and using personal data - 'necessary processing' or a 'lawful basis'.
2. **Purpose limitation:** To be clear about what the processing purposes are, from the beginning.
3. **Data minimisation:** To ensure personal data is adequate, relevant and limited to only what is necessary.
4. **Accuracy:** That personal data is correct and not misleading.
5. **Storage limitation:** To retain personal data only as long as it is required, justified by purpose.
6. **Integrity and confidentiality (security):** Appropriate security measures must be in place to protect data.
7. **Accountability:** To take responsibility for what is done with the personal data.

With regards to reducing energy consumption, particular attention will be paid to these principles in due course when justifying substitution of varying security functionality with privacy application. Such privacy functions have mature proof of concept under the aforementioned financial regulation PCI-DSS, using tokenisation. Alternatively, there are the privacy functions of anonymity and pseudonymisation [85]. Anonymisation means the inability to identify a living person amongst a data set - a data set of a group's health, for example, where many subjects will belong to a group of age or gender, but none shall stand out as an individual. Anonymisation is often used in statistics for geography, health, demographics, and risk factors like insurance. Pseudonymisation, compared to anonymisation, does not alleviate data of all identifiable information - but reduces the link of a dataset with the original identity of an individual.

Pseudonymisation and anonymisation as privacy functions overlap with security, as they could mean an encryption scheme, or a nickname with meaning to operatives but not outsiders. They could also mean a reference code, an artificial identifier indexed against the real data in a central server. Tokenisation is also similar to encryption, but often operates on a simple substitution to disguise data such as credit card numbers. Tokenisation can therefore be 'unlocked' without a key, particularly with well known credit card tokenisation algorithms used as part of the Cardholder Data Environment (CDE) [86], of PCI DSS [87] - but it requires a series of substitution numbers to do so, which will be anonymised in the background.

The approval of PCI DSS for use of cryptographic alternatives such as pseudonymisation for sensitive data, indicates that avoidance of heavily mathematical functions is entirely safe, practical, and even encouraged under the governing bodies of financial regulation. This work investigates where it is safe to substitute security functions for privacy functions.

However, privacy functions should operate within the overall safety of traditional CIA security functions. As PCI-DSS is a mature example of how tokenisation has provided a secure substitution of security techniques, a useful comparison between the two approaches demonstrates several advantages a decentralised TLS model would offer over token-based authentication systems. Whilst token-based authentication systems are simpler to use than TLS which utilises complex underlying PKI, TLS offers numerous advantages. With reference to [88], such advantages are::

1. **End-to-end encryption:** Communications can be encrypted from client to server to ensure no interception or tampering. With token-based authentication, communication is typically, but not always, encrypted only between the client and authentication server, as opposed to the server holding the requested data. The authentication is responsible for providing resources based on a user's credentials - thus leaving a potential opportunity for data leak in part of the communication chain.

2. **No single point of failure:** It is possible to distribute certificates across multiple servers for client verification, so that if one server is compromised, the system as a whole retains security. In token-based authentication systems, the authentication server renders itself as a potential single-point of failure.

3. **Trust model:** Similarly to certificate distribution, the trust model of a decentralised TLS is the difference between a single and multiple servers. Where trust is distributed amongst many places in TLS, the authentication server is a single point of potential failure in token-based authentication systems.

Taking the advantages of a security model to act as an overall perimeter to promote privacy functions within it, the next part considers the role of GDPR within security functions.

2.3.3 GDPR and the CIA

Privacy by Design (PbD) has been proposed using a Quantitative Threat Modelling Methodology (QTMM) (Luna et al., 2012), by risk assessment based on privacy-related attacks. The difference between privacy-oriented assessment is that other forms of threat modelling are largely agnostic of involved data subjects (Sion et al., 2019). The GDPR sets 99 articles and 173 recitals towards protecting the privacy of personal and sensitive data for subjects - and processes to attain that protection, with and without security functions.

Although the aquaponics system is largely devoid of personal and sensitive data belong to human subjects (we shall assume the fish don't mind their data being studied publicly), then the usefulness of the GDPR pertains to assessing a 'data-oriented' or 'content-based' attack risk, and ideally will result in substitution of security functionality with the less mathematical privacy functionality. Following the GDPR principle of 'necessary processing of data and nothing more than absolutely necessary', the challenge is to reduce protected content, and processing wherever possible.

GDPR privacy functions can be realised in several ways:

- **Data masking:** (Larrucea et al., 2021) refers to disclosure of data with modified values. This could be character shuffling, tokenization, encryption, or character substitution - one of the oldest forms of cryptography. Substitution is where a value is exchanged for another value, to make identification or reverse engineering difficult. This character set will be the same throughout the policy, but may turn into a centralised process.

- **Pseudonymisation:** (Kangwa et al., 2021) is data de-identification that substitutes private identifiers with false identifiers or pseudonyms. "John Smith" could be changed to "Mark McConnell", but this would require some form of centralised database.

- **Generalisation or minimisation:** (Goldsteen et al., 2021) is the removal of data to prevent identification, such as a house number of an address, so accuracy of data remains whilst not being too specific.

- **Perturbation:** (Broen et al., 2021)is the use of round-numbering and adding random noise to modify a dataset. The set of values must be proportionate to noise to avoid disturbance, making this method quite complex.

- **Synthetic data:** (Slokom and Larson, 2021) is algorithmically-generated information with no relation to the real case. These datasets represent patterns from real data by invoking standard deviation, linear regression, medians and other statistical methods. It is of no use to our comparatively simple model. Permutation (Barati et al., 2021): data swapping or shuffling rearranges dataset attributes so that they do not fit original information, often by switching columns in databases.

2.3.4 Scoring metrics

Metrics are used to quantify security applications - to attach a value so that attributes are in someway comparable and quantifiable. They do not have to be an accurate representation at this stage, just relative to each other in order to propose energy consumptions relatively. A Data Protection Impact Assessment (DPIA), or risk-rating methodology can produce objective results using likelihood and impact factors for calculating risk severity, and a single numerical score ranging from low at one to critical at four.

The Open Web Application Security Project (OWASP), provides a risk rating template [89], to objectively evaluate risk severity by multiplying likelihood of attacks from threat agents and vulnerabilities, by impact on technical systems and business continuity. Factor scores are ranked 1-9, based on a 'worst case scenario', and are fulfilled by the following criteria:

1. **Estimating likelihood:** Threat agent factors
   (a) Skill level: How technically skilled the threat agents are.
   (b) Motive: How motivated the group is likely to be, based on the value of the reward.
   (c) Opportunity: Resources and opportunities required.

   (d) Size: The origin and position of threat agents of the system.

2. **Estimating likelihood:** Vulnerability factors
   (a) Ease of discovery: How easy it was to discover the vulnerability.
   (b) Ease of exploit: How easy it is to perform the exploit.
   (c) Awareness: The knowledge of the exploit the threat agents are assumed to have.
   (d) Intrusion detection: How likely it is that the exploit will be detected.

3. **Estimating impact:** Technical factors
   (a) Loss of confidentiality: How much data could be disclosed and how sensitive it is.
   (b) Loss of integrity: How much the data could become corrupted.
   (c) Loss of availability: How much of the service could be lost, and the vitality of that service.
   (d) Loss of accountability: The traceability of the threats to their origin.

4. **Estimating impact:** Business factors
   (a) Financial damage: Resulting financial loss from an exploit.
   (b) Reputation damage: The extent of reputation damage to the business.
   (c) Non-compliance: How much exposure non compliance would introduce.
   (d) Privacy violation: The amount of resulting personal and sensitive information disclosure.

Scores resulting in low risk severity are not a concern, and medium scores are generally considered tolerable with monitoring, but high and critical results require immediate attention. To gain similar results for IoT security and energy, a novel template can be composed with additional likelihood and impact factors derived from the results of the SSM testing; CATWOE, BATWOVE, SWOT and TOWS.

2.4 Existing solutions

The sections up to now have discussed the characteristics of domain, likely threats, and how to address those threats with risk-driven modelling techniques. Now we shall look at existing solutions, what others have done to address those same IoT security and privacy threats in smart farm applications.

This section is structured in three parts, where security solutions can be grouped into three topologies: centralised, decentralised, and distributed. Many security functions overlap between the topologies, and so this section serves as an introduction to application of the functions, before exploring their mechanics in depth in the next section.

2.4.1 Centralised security

Centralisation pertains to governance and processing using a central authority, usually for encryption key escrow, device identifiers, intrusion detection, and review and approval of new network devices. Organisations have typically used centralised services to provide X.509 certificates for banking and shopping gateways, and this has been a safe approach for a number of years - appropriate for servers that operate from a single or small number of locations. Centralised infrastructures are, however, energy inefficient, usually rely on internet connectivity, must be active and online in order to be used, and are rendered vulnerable by a single point of failure [90].

TLS employs X.509 certificates to bind keys with identities. The management of these certificates in undertaken by a Public Key Infrastructure (PKI), typically consisting of five parts; Certificate Authority (CA), to store and issue X.509 certificates, Registration Authority (RA), to verify entity identity, a central directory to securely store and index keys, a management system for access and distribution, and a certificate policy to state procedures and rules. This is a cumbersome and delicate series of operations criticised for codependency and vulnerabilities due to the centralised data stores [91]. PKI is also storage-intensive, populating lists for both current and revoked data, causing a scalability issue for IoT. The CA of every website is hosted on its server, available to view as current and sufficient by site visitors. Revoked certificates have displayed as current in some browsers, and have been identified as a security flaw [92] - further disproving the value of the X.509 system. Key escrow for certificates also undermines privacy [93], and has been described as placing "Keys under doormats" [94]. Official bodies are granted access to TLS communications for law enforcement using PKI facilities [95], with escrow hosted on servers and retaining key ownership. Although it is possible to operate TLS without certificates [96], there is no alternative fulfilment for verifying the identity of the server by which you, as the client, are contacting - and therefore without employing TLS there is no assurance of security. In summary, even with the heavy implementation and energy consumptions of encryption for privacy, there are still privacy issues present. In the case of IoT applications where the majority of data readings are benign, such a false perception of privacy only serves to drain battery life - in return for a questionable infrastructure.

Other centralised security solutions include Software Defined Networks (SDN), gateways [97], Certificate-Less Public Key Cryptography (CL-PKE), [98], and Certificate-Less Signature schemes (CLS), using Key Generation Centres (KGC) [99]. Multi-party protocols have been implemented using TLS's Diffie-Hellman Key Exchange (DHKE), [100], and broadcasting using Public Key Cryptography (PKC), has been successful with public keys [101]. However, key escrow remains an issue with centralised security, where binding identity and public keys still remains the burden of a single server.

2.4.2 Decentralised security

Decentralised security is a hybrid between centralised and distributed, where responsibility and workload is distributed to several nodes, rather than just relying on one server. In a distributed network, every node is responsible and has a workload, and the material is duplicated. Decentralised systems have limited duplication, and specific allocations, but are much more robust than centralisation.

The trust issues of decentralisation away from a central node is a key topic in the development of such a topology [102], where removal of trust is represented by removing the root certificate functions of a central server. Since the aim of decentralisation is to remove the key escrow burden, self-signed certificates have been developed, where a device can prove its own authenticity without a governing CA.

There are two ways by which to authenticate the client and server devices in the TLS connection using X.509 certificates. The first and most traditional way is to send the device's Public-key and a Certificate Signing Request (CSR), to a trusted CA [103], and the second is for the client, to act as a CA by generating a root certificate itself and signing it using a Digital Signature Algorithm (DSA) and without the other content of the certificate identifying the organisation. Servers without some form of genuine authorisation, such as from VeriSign are considered untrustworthy, since any machine can self-authenticate, and self-signed certificates are not to be trusted with data such as credit card details. Unfortunately, neither CA-authorised nor self-signed certificates have been effective on browsers, as many certificates continue to operate out of date [104].

However, in absence of a browser and data private to an individual, as is the case with many IoT applications - particularly environmental or agricultural, mutual authentication of client and server may not be such as issue compared with traditional security handshakes. IoT applications are far different from financial transactions in client-server relationships, and as long as the environment is assured that the correct client-server data exchange remains that way, the requirement of browser-type interrogation supporting a certificate chain is unnecessary. In practical terms, we are not entering financial or personal data, and authenticity can be established for hardware using network keys of even a series of LED blinks representing a sequence similar to Morse code. We know that we are communicating with the correct device - that device can then be 'locked in' to a relationship using authentication, and issued the instructions set necessary to operate an aquaponics environment. Away from traditional use of X.509 certificates, such as online shopping and banking, using self-signed certificates, and the DSA on its own, removes many problems of centralised TLS and PKI security.

Other solutions to decentralised security and trust include game theory [105], interrogating nodes based on self-interest to render inappropriate authorisation impossible. This method is computationally complex, and not well suited to battery powered devices. Artificial Intelligence (AI), [106], and Machine Learning (ML), [107] have also begun security applications. AI and ML are complex and often need datasets and algorithm 'training' to accomplish effective practice, whereas the aquaponics domain will ideally have a very objective approach to which devices can be trusted and which cannot, with immediate effect.

Decentralised Identifiers (DID), have also demonstrated trust [108], using unique hardware numbers used as tamper-proof authenticity. DIDs refer us back to the secure boot and flash memory encoding of the ESP32, where the EEPROM could provide such a secure facility. Other types of device identifiers have been used when deploying LoRaWAN, where re-keying could be performed without hard-wiring, using smartphone communication [109].

2.4.3 Distributed security

Distributed computing refers to a topology in which components of a software system are shared amongst a number of computers, or nodes. In a distributed model, multiple nodes can act as client or server, or change between roles to fulfil security functions such as authentication and authorisation. With scalability, this is a problem, as all transactions are processed by multiple parties, verified, and added as a data block onto an infinitely accumulating chain, known as a blockchain. In itself, this style of data storage is the opposite of what our domain needs, but the security mechanisms used within Distributed Ledger Technologies (DLTs), [110], such as the blockchain of Bitcoin [111], and the Directed Acyclic Graph DAG of IOTA [112], are very informative.

Whereas security key exchanges occur between two parties [113], or multiple parties [114; 115], a consensus mechanism is used within a DLT as automated review and approval of transactions. Consensus mechanisms are fault-tolerant processes used to achieve necessary agreement from a group for a single data transaction. The consent of multiple nodes then gives the permission for that transaction to reside as part of a data block on the chain. Although typically resource-hungry, the study of various distributed consensus mechanisms can be applied to platforms lighter than blockchain. Notably, IOTA utilises the graph DAG platform and with a limited number of consensus members, the graph concept could present a lightweight and granular alternative to traditional centralised approaches. Below are examples of popular mechanisms:

- **Proof of Work (PoW)**, [116] requires members of the network to solve mathematical puzzles to prevent anybody from occupying the system. This is known as mining, and widely used in cryptocurrencies to prevent attacks such as 51% [117] - but PoW is energy-intensive.

- **Proof of Stake (PoS)**, [118], select transaction validators based on proportion of ownership on the chain compared with other members, and although it does not use extreme amounts of energy like PoW, it is criticised for 'blockchain oligarchy', or majority ownership, and as such is vulnerable to the 51% attack.

- **Proof of Elapsed Time (PoET)**, [119], is where users wait a random period of time and the first to finish waiting attains leadership of the new block. PoET uses trusted code with trusted, known users on a permissioned, or non-anonymous, network.

- **Practical Byzantine Fault Tolerance (PBFT)**, [120], is based on the Byzantine Fault Tolerance algorithm in the late 1990's; a scenario in which Generals gather around an enemy city and must communicate without interception. They gather around the city and send one message a time through a messenger, relying on cooperation and coordination, ignoring unauthorised influence.

- **Proof of Importance (PoI)**, [121], introduced a rating system to combat the 'oligarchy' issue of PoS in which the rich become richer. The rating system requires a minimum vested stake, a minimum transferred rate, a recent transaction bound by time limit, and valid transaction partners identified as separate users. The algorithm does not require complex computation, but rather a series of experience proofs to read through quickly.

- **Proof of Capacity (PoC)**, [122], is where the mining node's hard drive space is used to determine the mining rights on the network. Considered an improvement on PoW, it takes 40% of the time to produce a block by dedicating storage and processing before mining begins, and removes the mining conflict of the same puzzles by allowing different 'plot' routes for working through puzzles - so there is both hard drive plotting, and block mining.

- **Proof of Authority (PoA)**, [123], uses identity as a stake for reputation-based consensus, which means validators are not staking currency, but their reputation instead. PoA blockchains are therefore secured by random validating nodes as trustworthy entities. In supply chains and hierarchies, PoA is considered an effective and reasonable solution.

- **Proof of Signature (PoSign)**, [124], is a new consensus mechanism which utilises a network of known, authorised nodes, registered and active under digital signatures. It is used for signature-verified transactions to include securitisation in addition to privatisation. Each node must sign off a transaction before the associated block can be appended to the chain, and if repeated, failed, attempts are detected, the node will be blacklisted.

- **Proof of Location (PoL)**, [125], enables a device's physical location coordinates to be broadcast to the blockchain without relying on that particular device. Radio, GPS and BLE-enabled devices can assess the physical location of nearby devices.

With knowledge of where authentication and authorisation are positioned by topology, we are now at an appropriate place to explore the technicalities of both through the fundamentals of cryptography, the CIA triad.

2.5 The CIA

The previous section considered the application of different applications of security in the context of network topologies. This exploration has informed us that a decentralised topology would be beneficial for both robustness and minimising the duplication of data. Each topology has been exemplified using security functions belonging to PKC, AI, ML, CL-PKC and CLS. Ideally, we will make use of the four functions provided in hardware acceleration on the ESP32, in the application of a decentralised aquaponics security framework.

Confidentiality, Integrity, and Availability (CIA), are the three fundamental concepts necessary to negotiate a channel for secure data exchange [126]. This section explores each third of the triad, with explanation of the functions relevant, including those hosted natively on the ESP32, and those additionally available within the library for function calls, MBEDTLS.

2.5.1 Confidentiality

Cryptography, or cryptology, is from the ancient Greek 'kryptos', meaning 'secret', and 'graphein', meaning 'to write'. Confidentiality underpins the nature of secret writing, requiring a key to unlock each message. Following the setup of a secure channel between two devices, or authentication, a message can be locked from one side, and unlocked from the other. This process keeps the message in confidence and exclusive to those two parties sharing the same key, known as symmetric encryption. Asymmetric encryption is the locking of a message with one key of a non-identical pair, and unlocking it with the counterpart key. Symmetric encryption is most commonly used in data exchanges between two parties, whereas asymmetric encryption is used as part of device authentication - discussed as part of Availability.

On the ESP32, symmetric encryption is provided by the Advanced Encryption Standard (AES), supported by one of the four hardware-accelerated functions, and available to call in several strengths using the MBEDTLS library. There are many forms of symmetric encryption algorithm, otherwise known as ciphers, with AES being the most widely used. Other ciphers include Data Encryption Standard (DES), Rivest Cipher 4, 5, and 6 (RC6), and Blowfish. Specifically for lightweight applications such as IoT, encryption ciphers include SIMON, PRIDE, and PRINCE.

Ciphers come in two forms - stream and block. Stream ciphers encrypt data by bits or bytes, using a shared key to generate a stream of bits (known as the keystream) which are then typically XOR-ed with the stream of plaintext bits to convert it to ciphertext. Block ciphers break data down into blocks of fixed sizes, before converting the block into cipher text. AES is a block cipher, encrypting data at a block size of 128 bits, or an array of 16 bytes. Knowledge of the block size is important for security hardening of certain cipher operation modes.

A cipher's operation mode is a variation of the implementation of the cipher's algorithm. AES is the most commonly used block cipher, and since the ESP32 supports it in hardware, these are good reasons to use it in this work. Some modes use an Initialization Vector (IV), otherwise known as a Starting Variable (SV), to enhance the randomness of encrypted data, or ciphertext. Often this will be the same length of an encryption block - 16 bytes in AES, and randomly reset for each secure session.

Below are explanations of the current modes available to AES in the MBEDTLS library:

1. **CTR** Counter (CTR) mode [127], turns a block cipher into a stream cipher, generating the next key-streamed block by encrypting successive values of an incremental '+1' integer counter. Counter mode used to be crticised for this predictability, but has now been widely agreed as safe, and that any issues with security are associated with the underlying cipher (AES), rather than the mode. CTR and CBC modes are recommended by Niels Ferguson [128], and Bruce Schneier [129].

2. **CBC** Cipher Block Chaining (CBC), [130], is a mode in which each block of plaintext is XORed with the previous ciphertext block, and then encrypted. The XOR (eXlusive OR), function denoted $\oplus$, is a Boolean logic operator [131], commonly used in cryptography by comparing two input bits and generating a single output bit. If the input bits are the same, the output is zero. Otherwise, the output can be one or zero, depending on the comparison. This mode requires a fresh and random IV in the first block to ensure ciphertext uniqueness.

3. **GCM** Galois Counter Mode (GCM), [132], is an all-in-one variation of AES, known as Authenticated Encryption with Associated Data (AEAD), and represents a complex mathematical approach based on Galois field multiplication, allowing a higher throughput of data than algorithms such as CBC.

   In GCM, blocks are sequentially numbered, the block number is then combined with an Initialization Vector (IV), encrypted using a block cipher (in our case, AES), the result of which subsequently XORed with plaintext, and this produces the ciphertext. Ciphertext blocks then proceed through a finite field arithmetic process, the result producing a data tag, used as an authenticity and integrity reference. GCM is a form of counter mode, in practice a stream cipher, employing AES for the encryption part. A unique IV is required for each stream so that blocks are encrypted to avoid repetition.

4. **CFB8 and CFB128** Cipher Feedback Mode (CFB), [133], uses the whole output of the AES block cipher, and generates a stream cipher. CFB has similarities with CBC encryption, in that it uses a chaining mechanism involving the cipertexts. In CFB the XOR-ing with plaintext occurs after the block encryption. In CBC it occurs before the block encryption. So, in full-block CFB mode a ciphertext block is encrypted and the result is XOR-ed with the next plaintext block to produce the next ciphertext block. (In a sense it also operates similarly to a stream cipher). It is also possible to have less than full block feedback. NIST (REF SP800-38A), defines CFB with a bit-width integer parameter, otherwise known as the 's' value. Each plaintext segment and ciphertext segment consist of 's' bits, referenced into the name of the mode.

2.5.2 Integrity

Integrity refers to the non-tampering of a data exchange, where the message can be compared to the receipt of its original, included in the same data exchange. A recipient can compare the receipt to the same value when creating a new receipt - if the values are identical, the message has not been tampered with. This receipt is a one-way function known as a hash, and although not strictly a security function (there is no key), it is irreversible and therefore undoubtedly true. The same input will always produce the same output, and we can use the analogy of a gnome in a box with a book and some dice:

A gnome works as a hash compiler. This gnome dwells in a large box, with an input window and an output window, some dice to create a string to act as the message receipt, and a book to record all the receipts. For every message he receives through the input window, he writes the message in his book, rolls the dice, and records the dice values next to the original message. He then hands the message and the dice value as a receipt, through the output window. For each message the gnome receives, he first looks in his book to make sure he has not generated a receipt for it before - and if he has, he outputs the same receipt. The receipts are always the same length because they act as a compression function rather than ciphertext, which is why they cannot be unlocked using a key.

Therefore using the MD2 compression function [134], for any length value will produce the same length string, but in an equally nonsensical fashion:

Hello:
b27af65e6a4096536dd1252e308c2427

I've got a brand new combine harvester and I'll give you the key:
0756e7e76c37759810e2422bbd17ccfc

The caveat to using an integrity function is timing. A receipt must be sent prior to its counterpart message, or with it - sending it after the message allows opportunity for any tampering to create a new receipt and send that following the intercepted message. The following hash functions are available in the MBEDTLS library:

1. **MD** The Message Digest (MD) [135], algorithm family is considered cryptographically broken now, as it suffers from collisions - or when two messages share the same hash value. Malicious users can take advantage of collisions, by accessing, imitating and altering data values. Although hash algorithms have been designed to be collision resistant, they can occasionally map two different inputs by the same output, known as the pigeonhole principle [136]. To avoid this, the longer and more complex the hash algorithm and output string, the less likely a collision is to occur - but the safer the hash, the higher the computation and thus energy consumption. MD operates on a 128-bit hash value, and where a byte represents a string character, MD2 produces a 16-character output, MD4 and MD5 produce 32 characters, and MD6 is available in 128,

256, and 512-bit hash values, providing character outputs of 32, 64 and 128 respectively.

2. **SHA** The Secure Hashing Algorithm (SHA) [137], is considered the current safest compression function, and is also available on the ESP32 as a dedicated hardware-accelerator in the crypto block. The SHA algorithms are divided by three families, outlined below:

   - **SHA-1:** Often referred to as SHA-0, produces a 160-bit, 20-byte hash value. Typically rendered as hexadecimal pairs, the output is 40 characters long. SHA-1 has been deprecated since 2005.
   - **SHA-2:** Provides a set of six functions; SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256. The structuring of each function is almost identical, with varying shift amounts, with SHA-224 and SHA-384 acting as truncated versions of SHA-256 and SHA-512 respectively, computed with different initial values. SHA-512/224 and SHA-512/256 are both truncated versions of SHA-512, also with different initial values generated by a method detailed in the Federal Information Processing Standards (FIPS), [138]. The SHA-2 family is still current.
   - **SHA-3:** Emerged in 2005, and are structured completely differently to both SHA-1 and SHA-2 families. This subset covers the broader cryptographic primitive family Keccak [139], and include AEAD ciphers Keyak [140], and Ketje [141]. The authors of Keccak have proposed additional uses for the function including a stream cipher, authenticated encryption, and tree hashing for faster performance - but have not yet been standardised by the National Institute of Standards and Technology (NIST). Keccak is based on sponge construction, a function based on random permutation for greater flexibility and collision-resistance. Although designed to replace SHA-2 in current applications when the time arrives, the NIST does not have any plans on deprecating the SHA-2 family.

2.5.3 Availability

Availability refers to consistent and reliable access to data for authorised parties. For simplicity, we can divide availability into two parts: authentication and authorisation.

1. **Authentication** The process of authentication is the proving that something is true and valid. A person can authenticate themselves using something they know, such as a password, something they do, such as a pattern, or something they have, an identity card [142]. Multi-Factor Authentication (MFA), is when more than one of these is required, modernly used in banking by inputting credentials into a web browser and receiving a One-Time Pad (OTP), ephemeral verification code by SMS, for example.

   In the context of the smart farming domain, authentication is the mutual agreement of trust between two devices. This is the ‘handshake’ of TLS (discussed in detail in the next section), and commonly uses the asymmetric encryption function RSA.

Rivest, Shamir and Adleman, developed an algorithm (RSA) featuring the generation of a public and private key pair, an example of what we now refer to as Public Key Cryptography (PKC). A public key is generated using two very large prime numbers and an auxiliary value. The prime numbers are kept secret, as is the private counterpart to the public key. The public key, as its name suggests, is designed to be freely advertised, and given to anyone who wishes to decrypt a message encrypted with the private key. Equally, a private key can be used to decrypt a message encrypted by a public key. The computational basis for RSA (modular exponentiation) is hugely expensive. RSA keys are very long (several thousand bits) and should therefore be generated and used on constrained devices sparingly. 1024 bit-strength is estimated to take around 15 million modern computers running for a year to crack - but the recommendation is now to use a 2048 bit-strength or higher.

Essentially, when two devices authenticate, a public key is offered from one device to the other with the request of a session password. The password is then encrypted using the public key and returned to the original device for decryption using the corresponding private key. All subsequent messages can then be decoded with this shared, short, symmetric AES key, aforementioned as part of confidentiality.

2. **Authorisation** Access control and authentication are two main challenges in security and privacy requiring an immediate solution in the heterogeneous environment of IoT [143], proposed a capability-based authorisation model in which a token provides trust and a session key is produced to authenticate using Elliptic Curve Cryptography (ECC). Tokenisation has been introduced as ephemeral data for cryptocurrency transactions alongside various key exchange systems such as Diffie Hellman, RSA and the lighter ECC. Whereas tokens are not strictly cryptographic functions and are typically referred to as advanced pseudonymisation as a privacy function [144], they do limit the spread of data beyond a “need to know” basis, a fundamental concept of the GDPR [145].

   Most mature AC systems are based on roles or rules, and utilise Access Control Lists (ACL), and are not dissimilar from the traditional Relational Database Management System (RDBMS) [146], mapping usernames to privilege levels throughout known sets of data. ACLs present a few problems to the IoT network. Firstly they are central, and have to be kept somewhere that can be accessed by every node or microcontroller, then there's the requirement to know all the data available, and somehow map it to people or devices that will read, write, or execute it - even during the 1990's before the big data we process now, this was becoming an issue. Relational data management is difficult with IoT since it restricts autonomy, and becomes increasingly complex as new developments are integrated. Finally there's the duplication of data - for example if this list covers all possible relationships (for which some M:N relationships will require additional tables for normalisation) [147], this is a big expectation for limited storage space. We consider other, mature AC systems:

   (a) **DACS: Distributed Access Control System**, [148], is rule-based, and operates on the identity of users credentials within the organisation. This type of account setup would apply to every microcontroller in a network, taking time and resources to create large uniqueness for what should be an automated, preferably self-healing, relatively generic node.

(b) **DAC: Discretionary Access Control**, [149], is also rule-based, and assigns access rights based on rules specified by users, utilising ACLs and capability tables - rows with subjects, and columns with objects.

(c) **MAC: Mandatory Access Control**, [150], is considered the most strict of authorisation systems, and uses hierarchy to apply granular levels of access within a varying number of tiers, or 'security labels' often under the classifications of low, medium or high. The MAC system also permits access on a 'need to know basis', reminiscent of the GDPR's 'necessary processing', and thus pertinent to the topic of privacy. MAC decides which labels are attached to the documents within a system depending on the context in which it was created. For example, in a military setting, a user may occasionally need access to data above their clearance level in a state of emergency. A standard multi-level security MAC system would not allow this, and so the issue is inflexibility.

(d) **RBAC: Role-Based Access Control**, [151], operates on roles, rather than individual user accounts, using the 'least privilege' possible for the operative to fulfil the job, similar to MAC but not as strict.

(e) **PBAC / ABAC: Policy, or Attribute-Based Access Control**, [152], is whereby access rights are granted using policies to combine resource, user, object or environment attributes. This is a Boolean-centric model using 'if' and 'then' statements about who the request is made by, the action they request and for what resource. This model does not use pre-defined roles or lists.

Other applications include capability certificates provided by policy engines, Elliptic Curve Cryptography (ECC), and based authorisation certificates [153], but utilising certificates requires some form of authority by which to negotiate terms such as signatures, origin, roots and leaves - a problem similar to ACLs and centralisation. Similarly, keys have been used to update access and permit certain users - but a centralised system must also be employed to generate and monitor fair usage [154]. [155] presented a hierarchical authorisation system in which users of similar privilege levels are placed in the same group. This type of trust-organised placement perhaps makes more sense in IoT, where greater trust correlates to older, probably original devices.

With reference to the previous section regarding decentralised and distributed security topologies, DLTs have demonstrated robust and accessible networking for cryptocurrencies [156], data storage [157], asset tokenisation [158], and e-commerce [159]. Although DLTs are generally constructed in graph and blockchain form, they all accumulate data to be operational - and this inevitably requires more storage and energy to operate.

Decentralised architectures introduce a hybrid design, in which there is no single point of failure, but a defined group of devices either shares the same data, or relies on its own central device. A decentralised infrastructure can be spread over a long range using any number of technologies with communicating central nodes being elected by voting systems using reputation, such as with P2P ranking such as for file sharing online [160].

CIA implementation can be composed of any combination of the above functions using the MBEDTLS library. Although the three CIA attributes need not be present all the time, integrity and availability are always necessary to prevent tampering or identity masquerade. In summary, it is the confidentiality, or encryption cipher, that can be safely removed from the set of three attributes. This flexibility is not available in TLS, and should be exercised with caution. The risk-driven DPIA proposal is intended to ascertain appropriate use of confidentiality so that it can be selectively removed to reduce energy use as part of a TLS overhaul. In the final section, we consider Transport Layer Security (TLS) - the components and mechanisms which have made the de facto standard so successful, whilst simultaneously inappropriate to IoT applications.

2.6 Transport Layer Security (TLS)

The previous section examined the functions supporting the fundamentals of security - CIA, authentication and authorisation. There was also a brief summary of how those functions contribute towards a set up of a secure channel using PKC - the handshake, or session setup, and data exchange. In this section, we present the TLS 1.2 and 1.3 handshakes using the X.509 structure, and the various issues surrounding the use of certificates. Below is a visual comparison between TLS 1.2 and 1.3:

2.6.1 TLS 1.2

A handshake is a negotiation between two parties; the client who initialises the connection through a web browser, and the server. TLS 1.2 supports interrogation of the server using the X.509 structure - the client is not interrogated for authenticity. To form a secure channel between any client and a genuine server, it is the server who needs to prove they are a genuine bank, shop etc. X.509 certificates demonstrate an excellent use case for supplying proof of server authenticity to any visitor. The handshake follows three steps; negotiates a ciphersuite, authenticates, creates a session.

1. **Negotiates a ciphersuite** Web browsers are the most common clients, and the TLS requirements can vary between popular browsers such as Google Chrome and Firefox. On the server side, there are variations between Operating Systems (OS), such as Linux Apache and Windows Server. This first negotiation step is to agree on the most suitable capabilities of both client and server - the mutually supported cryptographic functions.

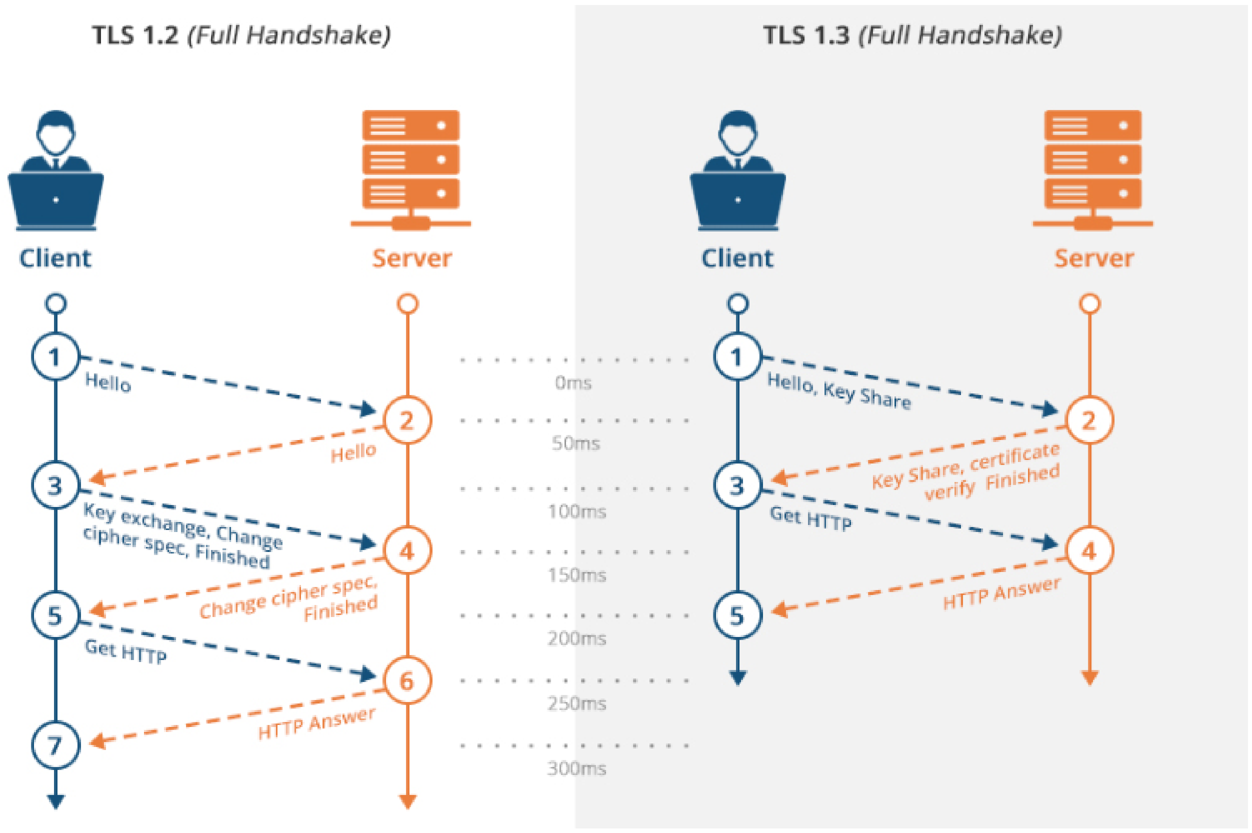

Figure 2.4: TLS 1.2 and 1.3 Handshakes, image source [6]

The methods agreed on are known as a cipher suite. A cipher suite contains the encryption, decryption, hashing and digital signature algorithms used to fulfil CIA for data exchange later. Once the cipher suite is agreed, the server will send its certificate to the client - the X.509 proof of ownership.

2. **Authenticates** Upon receipt of the X.509 certificate, the client checks the digital signature of the certificate to ascertain that the claimed origin of the certificate matches its truthful origin. This step is essential, and a mismatch between claimed origin and truthful origin will result in the client destroying the connection and starting again. All TLS certificates are issued by a CA. Many organisations have their own CA, or they will buy certificates from trusted authorities such as VeriSign. The digital signature proves the origin of the certificate, linking the certificate to its CA. The certificate is also associated with a private and public key pair owned by the server, where the private key is used to unlock messages encrypted by the public key.

   Although the public key is designed to be exposed to infinite clients, the digital signature is used to provide proof of origin - to prove that a message came from where it claims to have. This origin is attested by use of the sign and verify functions. The server will only be able to decrypt and use the data encrypted by its corresponding public key if it possesses the correct private key - so any attack spoofing the server will be to no avail. This is where TLS employs asymmetric encryption, traditionally RSA.

3. **Session setup** The last part of the TLS handshake requires creating the secure session, in which a shared key is used to encrypt and decrypt messages on both the client and server side. The client, now in possession of the server certificate, will generate an AES key (16 characters using AES-128, or 32 using AES-256), encrypts it using the public key on the certificate, and returns it to the server along with a combined integrity and authentication code (SHA-HMAC). The server will then decrypt the message, containing the AES key, and check it against the SHA-HMAC. This is the next step of interrogating the server - if the server can unlock the key, the session is set up securely. The server then responds with a confirmation that a secure channel is set up, and all subsequent data exchanges will be encrypted using the symmetric key, until the session is closed.

4. **Ten-step negotiation** The full handshake follows ten steps:

   (a) **Client to server: ClientHello:** "ClientHello" is the client greeting the server, with intentions to negotiate a secure HTTPS connection. The client lists their cryptographic capabilities so that the server can then offer a matching cipher suite from their capabilities. This hello message also includes a large, randomly generated random number known as the 'client random'. The significance of random numbers in TLS negotiation is discussed after the negotiation steps.

   (b) **Server to client: ServerHello:** The server responds to the client with its own polite "ServerHello" greeting, announcing the mutual cipher suite selected from the list previously provided by the client. The server returns a randomly generated large prime number, known as the 'server random'. If the server returns connection parameters (a ciphersuite), that the client cannot accept, the connection terminates, and the secure channel fails before messages can be exchanged. A ciphersuite must include an encryption function for confidentiality, such as the AES Cipher with a bit strength (AES-128), and a mode (CBC), a hash function for integrity such as SHA, also with a bit strength (SHA-256), and now an availability function - Hash-Keyed Message Authentication Code (MAC), for proving origin. This ciphersuite will read something like "AES-128_CBC_HMAC_SHA-256".

(c) **Server to client: Server certificate:** The server will now send its TLS certificate chain, including both root and intermediate certificates to the client. Definitions of these two certificates are covered after the TLS negotiation process. To provide server authentication, the TLS certificate is digitally signed by its organisation's CA, allowing the client to verify genuine origin. Whilst checking the certificate chain upto the CA, the client can detect other problems with certificate data such as a certificate which has expired, or an incorrect Domain Name Server (DNS). The client will also ensure that the server possesses the certificate's private key - of which the correlation with the public key is tested in step 6, key exchange.

(d) **Server to client: Server key exchange:** This step is relevant only for authenticating two parties. In the instance of DHKE, where the server provides a contribution to the secret key, and requires authentication from the client. For this example using RSA, we can ignore this step.

(e) **Server to client: ServerHello done:** The "ServerHelloDone" message politely informs the client they are finished negotiating parameters, are happy to proceed with setting up a secure channel, and awaits a response to agree to their side of the negotiation.

(f) **Client to server: Client key exchange:** The client now provides their contribution to the session key by testing the parameters provided earlier by the server, for its authenticity. The client will now generate a random string, now required to be of a 256-bit length, or 32 bytes, known as the Pre-Master Secret (PMS). This is not the actual key used for the session, but the master key, usually an AES key, from which each future session key will be derived, using a Key Derivation Function (KDF). Deriving each AES session key from a PMS ensures Perfect Forwarding Secrecy (PFS), and is discussed following the negotiation process.

(g) **Client to server: Client changes cipher specification:** This message informs the server that the client has generated a session key from the PMS using a KDF, and will now use encrypted communication. The client is changing the cipher specification from the context of key exchange to message encryption and decryption, or from RSA (asymmetric), to AES (symmetric).

(h) **Client to server: Client finished:** The client sends a message to the server to indicate they are done negotiating on the client side, and that their contributions for the handshake are complete. This "Finished" message is the first of the encrypted relays, using the shared session key, and will include the first Hash-Keyed Message Authentication Code (HMAC). A MAC is proof of message authentication to show the message has come from the correct server, and a hash is proof of integrity, or that the message has not been tampered with during transit.

(i) **Server to client: Server changes cipher specification:** This message is a confirmation that the server will now use encrypted communication. The server decrypts the PMS sent from the client, using the last RSA action, and computes the session key to match the same session key as the client. The server sends "ChangeCipherSpec" back to the client to confirm this change.

(j) **Server to client: Server finished:** The server sends its final message "Finished", using the session key computed from the decrypted PMS, and uses that session key for the first encrypted relay from the server to the client. At this point, both parties share the same session key, and have exchanged data. The secure session is now set up.

2.6.2 TLS 1.3

TLS 1.3 mitigated many TLS 1.2 vulnerabilities, predominantly by reducing the number of ciphersuites from 37, and support for over 300 legacy ciphersuites, to five. Naturally this eliminated many of the security holes from such a large range of ciphersuites, including outdated and broken options, but it also removed a lot of choice - choices which if implemented correctly, would have perhaps suited an IoT application better than the five left to choose from. TLS 1.3 ciphersuites are all very strong, but then again, TLS 1.2 was just as rigid in the enforcement of all CIA functions.

Within the TLS 1.3 handshake, it became possible for negotiation to take place using 'guesswork'. The client could ascertain at a probability of 1:5 which suite the server would use, reducing the handshake process from 10 steps and two rounds, to three steps over a single (or sometimes zero) round. The new Zero Round-Trip Resumption (0-RTT), (Tange et al. 2020), approach was regarded to have 'a whiff of future regret' - allowing the resumption of previous connections using pre-established cryptographic agreements leaves potential for replay attacks (Fischlin and Günther 2017). Apart from session resumption, TLS 1.3 follows a simpler procedure than TLS 1.2:

1. **Client to server: ClientHello:** Client sends supported cipher suites, client guesses key agreement protocol, client shares key.
   As with TLS 1.2, the negotiation begins with ClientHello, a request to instigate a secure channel. TLS 1.3 had reduced the number of supported ciphersuites significantly, from 37 to five. This means that the client can guess the key exchange protocol, and send

its contribution to the key share, assuming the Diffie-Hellman Key Exchange (DHKE), or similar mutual KE protocol, over the public channel at the same time.

2. **Server to client: ServerHello:** Server sends key agreement protocol, server shares key, server finished.
   The server will respond with ServerHello, the same as step two in TLS 1.2, and with a certificate to demonstrate server authenticity. Assuming that the client guessed the cipher suite correctly, both client and server will share the Authenticated Encryption with Associated Data (AEAD) protocol. This will most likely be the GCM cipher. If both parties agree on the cipher suite, the server will now send its contribution of the shared key as the server share of the DHKE, or other mutual KE protocol. The server computes the shared session key and will then declare it has finished the server-side of the negotiation.

3. **Client to server: client checks certificate:** Client generates keys, client finished.
   The client has all the required information to authenticate the X.509 TLS certificate and use the two shared key contributions to compute the session key at the client-side. This computation should match the server-side. On completion, the client declares it has finished the negotiation with a ClientFinished message. The secure channel is now set up and message relays can be safely undertaken.

2.6.3 Certificate Issues

In addition to the rigidity of TLS, there are several issues when implementing the X.509 infrastructure into IoT applications:

- **No browser:** Without a browser, the signature element of the certificate can be verified to check data authenticity without the bulk of certificate data.
- **Self-signed certificates:** Contain private and public keys within the same entity and cannot be revoked, making it difficult to detect security compromises.
- **Certificate tree:** Long, storage intensive, and unjustified without a browser.
- **Centralisation:** Vulnerabilities of singular authorities, particularly without internet connectivity for remote maintenance.
- **Numerous protocols:** Each protocol may need its own TLS library.
- **Session timeouts:** The nature of persistent relationships for the smart farm domain operates very differently to the short-term sessions of typical TLS applications.

Although TLS 1.3 sought to reduce the authentication process by incorporating 0-RTT, storage requirements of part keys and certificates still contradict the underlying nature of constrained devices. These part keys are in addition to the certificates of underlying X.509

infrastructure, and as such, are arguably more of a burden than the requirements of TLS 1.2. TLS 1.3 also uses fewer ciphersuites than 1.2, but these are more demanding and less flexible, and so are also headed in the wrong direction when considering energy consumptions.

Ideally, the new design will eliminate the centralised PKI, X.509 certificates, not introduce the part key notion of the 0-RTT, and reduce the security of the domain to reflect a largely non-sensitive series of data exchanges in which transparency of water temperatures is not an issue. Since the aquaponics systems require persistent relationships with a single entity of a data hub for monitoring and control, rapid succession of authentication is no longer a requirement, and will thus naturally aid in reducing the demands of building new secure relationships with multiple entities.

This now brings us to a close of the mechanics of TLS, and the literature review. We are now in a good position to present a requirements analysis for the four parts of a suitable security solution.

Chapter 3

Aims and Objectives

The previous chapter examined existing solutions, problems and gaps pertinent to developing a security solution for the IoT aquaponics domain. This chapter considers the research challenges and research questions used to structure the empirical chapters of the research, towards confirming the hypothesis:

- An energy-efficient IoT security solution can be created by combining a reduced variant of TLS with a low-overhead consensus-based access control.

Within the scope of aquaponics, the featured IoT framework plays a pivotal role in facilitating real-time monitoring, control, and efficient communication among devices. It employs a range of technologies tailored to the aquaponics context, such as specific sensor-actuators for water quality monitoring, pH levels, and so forth.

The primary objective of our IoT framework is to extend the lifespan of IoT devices in this aquaponics setting. The focus of our research centres on the reconfiguration of Transport Layer Security (TLS) with a view to reducing energy consumption. This approach is based on the understanding that constrained IoT devices are often limited by their battery life, and reducing their energy consumption can significantly extend their operational lifespan.

By re-configuring TLS within our IoT framework, we aim to enhance the energy efficiency of these constrained devices, consequently improving the overall sustainability and cost-effectiveness of the IoT framework for aquaponics. The amalgamation of these optimised technologies and techniques provides a more effective, efficient, and secure IoT framework, specifically designed to serve the needs of the aquaponics domain.

Therefore, the primary focus of this work is the reconfiguration of Transport Layer Security (TLS). However, this reconfiguration is not an end in itself, but a means to an end. The ultimate objective is the implementation and optimisation of a secure IoT framework, specifically tailored for the aquaponics application domain. In this context, TLS reconfiguration is a key element, as it allows for the enhanced security and functionality required in a potentially vulnerable and complex IoT environment. Therefore, while a significant part of this work involves the technical details of TLS reconfiguration, this is all done with the aim of creating a more effective, efficient, and secure IoT framework for aquaponics.

We begin with considering the research challenges, elements of the TLS infrastructure which cause problems in an IoT setting, and what possible alternatives there could be to those mechanisms relying on the centralised infrastructure. We then propose research questions with aims and objectives to structure the research into four investigations; a risk-driven energy assessment, authentication, data exchange, and authorisation.

3.1 The Challenge of TLS Infrastructure

This section references section 2.6 of the literature review, 'Transport Layer Security (TLS)'. We first address the challenges presented with applying TLS to IoT, providing alternative possibilities that the ESP32 and MBEDTLS library support. We shall then present the three mechanisms of TLS, and categorise the solution aims into each. Finally, we present the research questions, their aims, and how objective results can be quantified. In the context of the aquaponics IoT domain, six challenges of TLS infrastructure have been deduced: no browser, self-signed certificates, the certificate tree, centralisation, additional libraries, and session timeouts.

3.1.1 No browser

Certificates cannot be practically read or challenged without browser view. IoT devices often do not utilise a browser to communicate with each other. A browser may be used after data collection at a hub, accessible by a mains-powered and very capable machine, beyond the remit of the device network, so certificates do not demonstrate authenticity to any interrogator through the intended browser medium.

3.1.2 Self-signed certificates

Do not offer any 'real' authentication. Self-signed certificates can be generated on devices rather than buying in certificates from a third party, but they offer no more authentication than a digital signature alone - the rest of the content is not visible to any user without the browser and accessible file system of a traditional client machine such as a desktop.

3.1.3 Certificate tree

Long, storage intensive, and unjustified without a browser. Root, intermediate and leaf certificates are parts of a long chain, requiring storage and a full system to govern. This is not practical on limited storage with multiple relationships over possibly hundreds of devices.

3.1.4 Centralisation

Typical TLS security with underpinning PKI architecture can be difficult to manage on a hugely dispersed and constrained network - particularly if WiFi is absent and the network is based on lighter protocols such as BLE or meshing. As the network scales, the limited storage and processing of constrained devices will become unrealistic without extended storage, which contradicts the purpose of a constrained network.

3.1.5 Multiple libraries

Each protocol may need its own TLS library. TLS traditionally operates on the internet, but most (if not all) protocols have a library for their own version. If each protocol requires its own certificate, uTLS, MQTTS, BLE-TLS and so forth, devices could quickly run out of space depending on how many protocols are in operation. However, when the built-in hardware acceleration is called on from the native MBEDTLS library, this processing can be undertaken before departure, requiring no security library for any protocol.

3.1.6 Session timeouts

TLS sessions reset client-server authentication very frequently. TLS sessions are very short, whilst the ciphersuites are very strong. Devices will quickly run out of battery if negotiations are invoked and closed for every data exchanged.

3.2 Re-configuring TLS Mechanisms

There are three mechanisms employed by TLS: authentication, data exchange, and authorisation. Ordering is in such a way to primarily establish a secure channel between two entities without the risk of an external entity discovering the means by which to set up the channel, using a large, resource-heavy process and keys. Following the initial setup, the smaller and less resource-hungry session key can be generated for each data exchange within that session. Finally, as a result of having a secure session, TLS presents a binary ‘in-or-out’ access control system.

Each mechanism is a topic of study towards a full solution, and will address the challenges that IoT faces when incorporating TLS, by considering a variety of alternatives available to the ESP32 and MBEDTLS library. Below we correlate the challenges and propose the alternative solutions:

3.2.1 Authentication

Forming exclusive trust between two devices. Authentication is where the X.509 certificates are required for client interrogation of a server before enough trust is built to provide sensitive financial or personal data. Since there is no sensitive or personal data in aquaponics systems relating to a living subject other than the fish (we’ll assume the fish don’t mind their data being shared), then there is no need for an X.509. Losing the X.509 rids the network of centralisation, the certificate tree, and the cumbersome X.509 tree.

However, there is still a requirement to know that the devices communicating have not been, and will not be, spoofed or intercepted at any time during communication. The ESP32 provides on-board asymmetric RSA for this purpose, where the public and private key pairs can be generated and delivered using exactly the same format as client-server integration. The difference is that they will not use an intermediary as a trust verifier. To authenticate origin of each device during this, a digital signature can be used to sign origin, and verify by the recipient as a counterpart of the public key. The digital signature is the same as included on a certificate, but without the superfluous content, or infrastructure.

3.2.2 Data exchange

Securing data exchange between two devices. Data exchange first needs securing, by passing an AES key from one device to another, initially encrypted using RSA, and subsequently all data to be encrypted by AES. Or at least that's the TLS way of data exchange. Depending on the nature of the content, most data exchanges in the aquaponics network from sensor-actuator to hub or cloud, will not require confidentiality. Environmental readings and control are the main purpose of the IoT involvement in a smart farm, and those readings are perfectly benign when read by the public - as long as they are not tampered with or spoofed. To this end, enabling transparency of data exchange content wherever practical bolsters robustness, because it lowers interest. However, all exchanges must contain a SHA-HMAC to ensure proof of origin (against spoofing), and integrity attestation (against tampering). Given this 'two out of three' CIA approach to data exchange, we hope to reduce energy consumption by at least 30% by default. To achieve this, the regular session setup must include a SHA-HMAC key as well as the regular AES, kept just as private.

The 'edge processing' made available by the ESP32's cryptography provisions allow flexibility away from burdening a centrally governing single server. Exemplified by payload security, all data can be encrypted, attested and verified at source and destination, regardless of protocol used. Since a channel is not being employed, numerous libraries to cater for each protocol channel need not be employed either.

Finally, the constant session timeout can be resolved. By resetting authentication every day, or every week rather than every data exchange, energy consumption can be reduced significantly. It is reasonable to assume that RSA and AES processing is far more energy intensive than SHA-HMAC alone for most of the time, but this investigation will be undertaken in depth later on.

3.2.3 Authorisation

Segregation of data for authorised access. By default, TLS will either allow access (via a shared, asymmetric or device password to access such), or it will not. It is very binary, and this does not protect against credential theft or human folly to the same end. However, other authorisation schemes do exist, ranging from one extreme to another by the medium of either centralisation (ACLs), or decentralisation (duplicated data on each node). Neither of which are suitable, yet, with reference to chapter 2.4 'Existing solutions', there have been many instances of consensus mechanism success to come degree - save the scalability issue. Blockchain applications in IoT are cumulative, regardless of consensus mechanism style. Aside from blockchain, consensus by PoW as an example is resource-hungry as puzzles must get harder, and whilst PoS offers some kind of hierarchy-based mechanism, it is threatened by oligarchy and of course, the 51% attack. In addition, segregated rights protecting entities from being written to within the network have not yet been explored in the consensus mechanism domain - a backup mechanism that could prove invaluable in times of disaster and recovery. In summary, a distributed or decentralised mechanism with the benefits of tiered safety pertaining to centralised systems could be a lightweight and scalable solution with backup and restoration as safeguarding.

This mechanism concerns the induction of a hybrid scheme, where a consensus mechanism could support IoT authorisation in a limited, ephemeral, and decentralised manner. AC in this sense would be an alternative to the TLS provision in the way that there will be three, four or even five levels of access, as opposed to the current two.

Research question one to four are now presented following the order of re-configuring TLS mechanisms, preceded by a risk assessment to inform the research of a testing methodology and expectations of results specific to the IoT domain application.

3.3 Energy-driven risk assessment

What are the security and privacy requirements of the aquaponics smart-farm when traded-off against energy consumption savings?

3.3.1 Risk assessment aim

The overall aim is to produce a method of identifying which components of a IoT-oriented security solution consume the most energy, and how to reduce those consumptions as much as possible without jeopardising the safety of the network. The experimental results from subsequent research questions should validate the DPIA by demonstrating a positive difference between TLS and the novel solution. Like with all risk assessments, this is a challenge of weighing up advantages and disadvantages, but with the aspect of energy being so poignant, we should reduce protection wherever reasonably possible rather than bolster it - this will address a resource-sensitive issue as a modification to typical risk-driven assessments found in the security domain.

To illustrate, in the context of our aquaponics application domain, the all-in-one typical TLS cipher suite is applied in such a way that prioritises energy efficiency over confidentiality in some cases. This approach will be informed by an energy-driven risk assessment, recognising that the high energy demand of data encryption could potentially deplete the battery life of IoT devices more quickly, undermining the overall operational efficiency and sustainability of the system.

In this scenario, while confidentiality remains a component of the CIA triad in the application framework, it is not given the same level of priority as the integrity and availability aspects. This is because, in the aquaponics setting, the nature of data typically collected and transmitted by the IoT devices, such as water temperature or pH levels, doesn't require high-level confidentiality as it doesn't involve personally identifiable or sensitive information. Thus, energy savings obtained by reducing the emphasis on confidentiality outweighed potential risks.

However, this doesn't imply that confidentiality is disregarded entirely in the application framework. Rather, its level of importance is calibrated according to the specific needs and constraints of our application domain, as explored in detail in chapters 4 and 5.

3.3.2 Risk assessment objectives

The objectives below outline the process towards invoking a Data Protection Impact Assessment (DPIA), driven by energy consumptions and security adequacy, based on the data exchanged in regular IoT-networked aquaponic systems use.

1. **Scope the domain:** Referring to section 2.3.1, 'SSM from a Business Perspective', apply the principles from the business domain to interpret attack motives, protection requirements, and the robustness and vulnerabilities of the aquaponics environment are likely to be. Typical of high-level business use casing, this is a conceptual and interpretive.

2. **Key findings:** From the broad, high-level scope, irrelevant threats can now be eliminated and replaced by domain-specific attributes specific to the application domain. The objective here is to list those elements to the benefit or detriment of an IoT-networked smart farm.

3. **Data mapping:** This is content discovery necessary to ascertain the data exchanges necessary for setup and operation. The solution is exemplary of optimisation by a domain-driven approach - by provision of the networked components, ie the ESP32 in the context of libraries, functions and protocols, and required data exchanges. Invitation of privacy principles in general will serve as an asset to this phase.

4. **Reduction and substitution:** To propose lighter and simpler functions to standard security ciphersuites used in TLS to optimise energy savings. Alternatives pertaining to the domain of privacy are widely available within the field of cryptography - the cryptography material native to the ESP32 correlates with standard security suites.

5. **Domain-Specific criteria:** To identify, with influence from the literature review and related work, those criteria suitable for an energy-driven risk assessment based

on assessing environment content specific to the aquaponics application domain, and assign quantifiable metrics to them.

6. **DPIA comparison:** The final outcome of the DPIA stage is to present the differences in TLS configuration. Whereas typical TLS configuration includes a full and standard ciphersuite necessary for temporary and multiple connections for accessing personal and sensitive information, a re-configured ciphersuite pertaining to the requirements of a persistent IoT connection will differ. The comparison between these two scenarios may demonstrate differences in energy consumption significant enough to justify considerably extended device battery life.

3.4 Energy reduction in authentication and data exchange

How much could energy consumption be reduced during authentication, and subsequent data exchange?

3.4.1 Authentication and data exchange aim

The aim is to determine the lightest method of authentication and subsequent data exchange available to the application domain. This will be the discovery and response to the challenges presented by TLS authentication specifically towards the application domain.

3.4.2 Authentication and data exchange objectives

Objectives pertaining to research question two and three feature the same power analysis testing methodology, producing measured results in energy for secure setup of channel session between two devices.

1. **TLS challenges:** Explore the problem backgrounds and challenges of key exchange and session invocation mechanisms within the de facto standard, TLS.

2. **Related work:** Discuss lightweight authentication and data exchange methods within the IoT domain, particularly those decentralised and privacy-oriented solutions, operating on decentralised or distributed topologies.

3. **Design attributes:** Respond to the challenges presented by TLS by proposing a series of authentication design attributes towards a novel and domain-specific security solution.

4. **Experimental results and discussion:** Prove or disprove the effectiveness of the new configurations by comparing the results of the experiments against regular TLS implementation and in the context of the earlier DPIA predictions.

5. **Contributions:** Summarise the novelties of the research as contributions towards the field, with details of design limitations and notions of future work available for further research.

3.5 Lightweight consensus mechanism for authorisation

Can a lightweight, self-healing consensus mechanism, be ascertained as safe?

3.5.1 Authorisation mechanism aim

This question cannot use TLS as a benchmark for comparison, because the nature of the existing authorisation system is a default, binary provision that does exist, but cannot be increased by role or rule. It either allows access, or it does not. With all other authorisation systems, there is a degree of granularity - role-centric or rule-centric.

Therefore, this is a completely inductive investigation of a novel authorisation system based on a series of requirements specifically for the aquaponics network. Although the theme of energy consumption is the driving force of the project, we test the model for safety, in the knowledge that the design principles towards energy efficiency will already be satisfied.

3.5.2 Authorisation mechanism objectives

This is a study into applying a granular access control model for additional business continuity and downtime detection by safeguarding and presence-checking devices respectively. It is an inductive design, modelled as an addition to the default TLS approach of 'in or out the network', as opposed to a replacement.

1. **Authorisation mechanism challenges:** Explore the problem backgrounds and challenges of authorisation in constrained devices, traditional applications, and heterogeneity.

2. **Related work:** Discuss the existing solutions for authorisation with regards to privacy and security, various network topologies, and popular consensus mechanisms.

3. **Design and model-checking:** Propose the design model of the new authorisation system, and present the results of formal verification for proving or disproving the safety of the model.

4. **Contributions:** Summarise the contributions of the authorisation proposal with application in the aquaponics smart farm domain plus any other potential use.

Chapter 4

Testing Methodology

The previous chapter proposed the research challenges to address, the three TLS mechanism alternatives in which to incorporate solutions, and the aims and objectives as the means by which to achieve these solutions. This chapter presents the methodology elements for each research question, in preparation for the results chapters.

4.1 Research question one

- What are the security and privacy requirements of the aquaponics smart-farm when traded-off against energy consumption savings?

Responding to question one firstly requires consideration of a risk rating methodology, metrics scoring, estimating likelihood and estimating impact. Secondly, for the DPIA to be domain-application specific, the functions available to the ESP32 edge device must be included as attributes within an energy-driven Data Protection Impact Assessment (DPIA), influenced by the original risk rating methodology. Functions available to the ESP32 are accessible via the mbedtls library native to the microcontroller, and call upon the hardware-accelerated cryptography block. Results are presented in Chapter 5, 'Energy-Driven Risk Assessment'.

4.1.1 Risk rating methodology

The methodology for testing the first research question is the metrics behind a typical risk rating assessment template used in industry, and exemplified by the Open Web Security Project (OWASP) [89]. Identifying risk and correlating the severity of that risk is the first step of being able to resolve it, and being able to resolve it appropriately given resources and priorities. By ascertaining severity by a scoring system, a Data Protection Impact Assessment (DPIA) can be produced to highlight where protective resources are proposed to be implemented - offering a great control of resource allocation.

Impact factors interpreted by the OWASP risk rating methodology are grouped into two categories: estimating likelihood by threat agent and vulnerability factors, and estimating impact by technical and business impact factors. Based on the scoring system of the OWASP factors, attributes more relevant to the application domain can be invoked to produce a bespoke, energy-driven DPIA variation - towards satisfying the aim of the research question.

4.1.2 Metrics scoring

The OWASP risk rating methodology calculates risk by multiplying likelihood by impact. As each attribute is scored by the user from a scale of least likely to critically likely on a scale of 0-9, the results are combined and divided by the total number of attributes to provide an overall score, correlating to overall risk severity of low, medium, high and critical. The value of such a methodology is that during the design phase, these projected attributes are subject to unlimited changes until optimum design is attained prior to implementation. In the case of an energy-driven DPIA proposed as the novel outcome and contribution of research question one, such a methodology will provide the designer with adequate security and the lowest consumption of energy to extend the life of battery or solar-powered microcontrollers. By ascertaining the data exchanges required of the system, the DPIA will exemplify how each piece of content can be numerically quantified in terms of energy consumption. This outcome may be adjusted unlimited times until a 'low' risk severity is produced by values of combined attributes specific to the application domain. Such a DPIA could then be used in conjunction with the design of any energy-conscious IoT driven application domain based on the specific data exchanges of that project for optimisation prior to implementation.

The OWASP risk rating factors relevant for interpretation into an energy-driven DPIA are presented below:

4.1.3 Estimating likelihood

Likelihood refers to how likely a system will get exploited based on threat agents and vulnerability.

1. **Threat agent factors:** 'Threat agent' is a reference to malicious attempts of an attacker, rather than system failure or user accident. Threat agents will exist for all systems, and so this set of factors is an important set for incorporating into the energy-driven DPIA. The aim for these factors is to estimate the likelihood of successful attack, based on the worst-case scenario. Ratings have a scale of 1-9 where 1 represents lowest impact and 9 the highest:

   (a) **Skill level:** How technically skilled the threat agents are:

      i. **1:** No technical skills.
      ii. **3:** Some technical skills.
      iii. **5:** Advanced computer user.
      iv. **6**: Network and programming skills.
      v. **9:** Security penetration skills.

   (b) **Motive:** How motivated the group is likely to be, on desirability of reward:
      i. **1:** Low or no reward.
      ii. **4:** Possible reward.
      iii. **9:** High reward.
   (c) **Opportunity:** Resources and opportunities required to find and exploit the vulnerability:
      i. **0:** Full access or unrealistic resources required.
      ii. **4:** Special access or some resources.
      iii. **7:** Some access.
      iv. **9:** No access or resources required.
   (d) **Privilege:** The privilege level of the threat agent group against the system:
      i. **1:** Developers.
      ii. **2:** System administrators.
      iii. **5:** Partners.
      iv. **6:** Authenticated users.
      v. **9:** Anonymous internet users.

2. **Vulnerability factors:** This set of factors is related to the vulnerability involved, with an aim to estimate the likelihood of this vulnerability becoming discovered and subsequently exploited. There is very little vulnerability in the data at rest or in storage since the domain will not hold financial, personal or sensitive information belonging to living subjects (fish do not count as they are apparently unperturbed by the publicity of their data), and so the vulnerability landscape is atypically insensitive. The energy-driven DPIA will not employ vulnerability factors as they are beyond scope - and so they have been omitted.

4.1.4 Estimating impact

Impact refers to the either technical, or system impact, or business impact.

1. **Technical impact factors:** Technical factors correlate well with the fundamental Confidentiality, Integrity and Availability (CIA) triad, and offer an excellent basis to propose the differences in key strengths and sessions, etc. The energy-driven DPIA will segregate the strengths of these CIA attributes into a suitably tiered metrics system as part of the energy and security trade-off:

   (a) **Loss of confidentiality:** How much data could be disclosed and how sensitive it is:
      i. **2:** Minimal, non-sensitive data disclosed.
      ii. **4:** Minimal, critical, or extensive non-sensitive data disclosed.

      iii. **6:** Extensive, critical data disclosed.
      iv. **9:** All data disclosed.

   (b) **Loss of integrity:** How much data could become corrupted and how damaged it would be:

      i. **1:** Minimal, slightly corrupt data.
      ii. **3:** Minimal but seriously corrupted data.
      iii. **5:** Extensive but slightly corrupted data.
      iv. **7:** Extensive and seriously corrupted data.
      v. **9:** All data is totally corrupted.

   (c) **Loss of availability:** How much of the service could be lost, and the vitality of that service:

      i. **1:** Minimal secondary services interrupted.
      ii. **5:** Minimal primary services, or extensive secondary services interrupted.
      iii. **7:** Extensive primary services interrupted.
      iv. **9:** All services lost.

   (d) **Loss of accountability:** The traceability of the threat to their origin:

      i. **1:** Fully traceable.
      ii. **7:** Possibly traceable.
      iii. **9:** Completely anonymous.

2. **Business impact factors:** Business impact factors consider the consequences of financial, reputation, non-compliance and privacy damage, often recorded as part of a company's audit of ISO 27001 Information Security Management System (ISMS). For the energy-driven DPIA, this material is useful for interpreting into intentional exposure of data and graded privacy of particular data exchanges:

   (a) **Financial damage:** Resulting financial loss from an exploit:

      i. **1:** Less than the cost to fix the vulnerability.
      ii. **3:** Minor impact on annual profit.
      iii. **7:** Significant impact on annual profit.
      iv. **9:** Bankruptcy.

   (b) **Reputation damage:** The extent to which reputation damage would cause the business:

      i. **1:** Minimal damage.
      ii. **4:** Loss of major accounts.
      iii. **5:** Loss of goodwill.
      iv. **9:** Brand damage.

   (c) **Non-compliance:** How much exposure non-compliance would introduce:

    i. **2:** Minor violation.
    ii. **5:** Clear violation.
    iii. **7:** High profile violation.

(d) **Privacy violation:** The amount of resulting personal and sensitive information disclosure:

    i. **3:** One individual.
    ii. **5:** Hundreds of people.
    iii. **7:** Thousands of people.
    iv. **9:** Millions of people.

The risk rating methodology attributes have illustrated the formula behind risk from likelihood and impact. Chapter 5 will proceed with this approach to invoke attributes suitable for the domain application, specifically under the influences of energy consumption, IoT protocols and function substitution within the realm of privacy.

4.1.5 TLS functions through the mbedtls library

This section presents the TLS security functions available to the ESP32 by calling the native mbedtls library. These functions access the hardware-accelerated cryptography block functions RSA, AES, SHA, and RNG introduced in Section 2.1, 'Application domain'. Functions used for device authentication (research question two), and subsequent data exchange (research question three), are required for incorporation into the DPIA for energy consumption assessment (research question one).

As the comparison is between typical TLS and re-configured TLS for energy optimisation, the functions used for all experiments have to feature the same attributes. The experiments will be set out to demonstrate the difference in application of the same functions, strengths, ciphersuites and mechanisms.

1. **Authentication Functions and Attributes**

   Authentication functions include key generation, signing and verification, and key exchange methods, whilst authentication attributes include the key lengths and strengths, detailed below:

   (a) **Key Generation**

   Authentication keys are required for both client and server. These can be two very different asymmetric keys, or they can be a symmetric shared key built from a contribution of data from both client and server. Either way, authentication keys are very different to session keys for encrypting and decrypting readings. The session key is invoked following authentication. The authentication key types available for this study through mbedtls are:

i. **Rivest, Shamir and Adleman (RSA):** Creators of an asymmetric key pair used in PKC, where the device's publicly-available key is used to encrypt a message, and upon receipt, the private key is used to decrypt the message. RSA has been used as a long-term standard due to its strength, but is notoriously heavy and the key bit-sizes are very large. The ESP32 hosts dedicated hardware for RSA acceleration. Energy testing of RSA key pair generation will demonstrate the energy consumptions of the following key-bit strengths, with 1024 considered insecure, and deprecated.

   A. **2048:** 2048-bit key strength.
   B. **3072:** 3072-bit key strength.
   C. **4096:** 4096-bit key strength.
   D. **7680:** 7680-bit key strength.
   E. **15360:** 15,360-bit key strength.

ii. **Elliptic Curve Cryptography (ECC):** Is a another approach to asymmetric encryption, based on the algebraic structure of elliptic curves over finite fields - allowing the key sizes to be around a third of RSA sizes, but with equal security. There are 13 curves available to the mbedtls library on the ESP32, but SECP192R1 and SEC192K1 are the approximate strength equivalents of RSA-1024, considered unsafe and have been deprecated. Energy testing of the ECC key pair generation will demonstrate the energy consumptions of the following 11 curves:

   A. **SECP224R1:** 2048-bit key strength.
   B. **SECP256R1:** 3072-bit key strength.
   C. **SECP384R1:** 4096-bit key strength.
   D. **SECP521R1:** 7680-bit key strength.
   E. **SECP224K1:** 15,360-bit key strength.
   F. **SECP256K1:** 15,360-bit key strength.
   G. **BP256R1:** 15,360-bit key strength.
   H. **BP384R1:** 15,360-bit key strength.
   I. **BP512R1:** 15,360-bit key strength.
   J. **CURVE448:** 15,360-bit key strength.
   K. **CURVE25519:** 15,360-bit key strength.

(b) **Signing and Verification**

The Digital Signature Algorithm (DSA), was described earlier as the only valuable component of the traditional X.509 certificate. Using this signature, both server and client devices can authenticate their own key pair immediately after generation. This means that instead of using a certificate and associated infrastructure to prove the origin of a public-key, it can be undertaken in conjunction with generation using one of two methods:

i. **RSA sign and verify:** The message is signed with an RSA computation using the sender's private-key, and is verified by the recipient by using the RSA computation using the sender's public-key.

ii. **ECC sign and verify:** The message is signed with an ECC computation using the sender's private-key, and is verified by the recipient by using the ECC computation with the sender's public-key. Energy testing of sign and verify functions for energy consumptions can be recorded at the key generation stage, since the key sizes and curves will influence time and power. Therefore, key generation, sign and verify functions will be tested together.

(c) **Key Exchange Methods**

The key exchange method sets up a secure channel in a hostile environment using either an asymmetrical method, where one device does the majority of the processing, or mutual authentication, where contributions are equal between client and server device. Key exchange can be undertaken using:

i. **RSA-PKC:** RSA-Public Key Cryptography is an asymmetric method in which the session key (typically a 128-bit AES key for IoT), is encrypted using the RSA private-key, and decrypted on the recipient side using the sender's public-key. The sender would have to generate the RSA key pair first, and sign the message (AES key), in addition to encrypting it, to demonstrate authenticity as a unique and genuine sender. Fortunately, the authentication of the client is already satisfied with it generating and sending the session key to the server, and only the server needs to prove its identity. Energy testing will consist of RSA encryption and decryption firstly for an AES128-bit key, and secondly for keys AES128, IV (also 128-bit), and HMAC (256-bit), to represent the additional work for the design. It also includes one signature and verification to represent interrogation of the server, using the lightest RSA key pair.

ii. **DHM:** Diffie-Hellman-Merkle is the mutual authentication key exchange for both client and server to contribute equal work under the key generation of RSA 2048 and above. Energy testing of DHM requires the RSA key list as with key pair generation, sign and verify. As with RSA-PKC, this shall be performed firstly with generating an AES128, and secondly with the AES128, IV, and HMAC to represent the new design.

iii. **EC-PKC:** Elliptic Curve Public Key Cryptography would have been the lightweight equivalent to RSA-PKC by encrypting and decrypting the AES session key using the elliptic curves. Unfortunately, elliptic curve encryption and decryption is not part of the standard TLS ciphersuite [161], and so it has not been ported to mbedtls for use on the ESP32. This cannot be tested.

iv. **ECDHE:** Elliptic Curve Diffie-Hellman Ephemeral, is a key exchange method in which both client and server create a secure channel by contributing equal work. Each party generates and keeps their own secret whilst exchanging a factor of that number to derive the same shared secret key. This is mutual authentication, and requires signing and verification for both parties. Energy testing is undertaken by recording the group of 11 suitable curves under the ECDHE protocol, with a sign and verify function to represent each party, based on the lightest curve drawn from the previous result. As previously, this shall be performed firstly with generating an AES128, and secondly with the AES128, IV, and HMAC to represent the new design.

(d) **Data Exchange Functions and Attributes**
Data exchange functions and attributes include those mechanisms corresponding to the CIA triad - where ciphers enable confidentiality, hash functions provide integrity, and Message Authentication Codes (MAC), prove authenticity of origin. Strengths and CIA separations are described below:

i. **Ciphersuites for Confidentiality**
Details of AES operation is in section 2.5.1, 'Confidentiality'. AES is available to the mbedtls in three recommended strengths, and five modes in which to operate. AES is both widely adapted in practice, and available on its own hardware-dedicated acceleration crypto block on the ESP32 itself, so it is a natural choice for the domain. AES operating in its various strengths and modes will be tested for the confidentiality aspect of the experiments. AES must use one of three recommended strengths:

A. **128:** 128-bit key strength.
B. **192:** 192-bit key strength.
C. **256:** 256-bit key strength.

AES must operate in one of five modes:

A. **CTR:** CounTeR mode, a stream cipher.
B. **CBC:** Cipher Block Chaining mode, a block cipher requiring an IV (Initialisation Vector).
C. **CFB8:** Cipher Feed Back mode with 8-bit integer parameter.
D. **CFB128:** Cipher Feed Back mode with 128-bit integer parameter.
E. **GCM:** Galois Counter Mode, including in-built integrity and message authentication.

ii. **Hash Functions for Integrity**
The hash function is a one-way mathematical process ciphered in a similar sense to encryption, but irreversible. The literature review details the value of hash functions for integrity under the analogy of a gnome in a box with a book and some dice. The ESP32 provides the Message Digest (MD), family but these have been deprecated as too high risk for hash collisions [162]. The Secure Hash Algorithm (SHA) family of functions is widely accepted, designed by the National Security Agency (NSA), and also employs its own dedicated crypto acceleration block, similar to AES. Integrity functions, ie, types of

SHA, are necessary to ensure non-tampering, and must be included. The SHA-2 family functions considered safe and available to the mbedtls library are as follows:

A. **SHA224:** Truncated version of SHA256, using different initial values.
B. **SHA256:** Computed with eight 32-bit words.
C. **SHA384:** Truncated version of SHA512, using different initial values.
D. **SHA512:** Computed with eight 64-bit words.

iii. **MAC for Authenticity**
A Message Authentication Code (MAC), sometimes known as a tag, is a short piece of information used to confirm the authenticity of the message from the claimed origin. Since this MAC can also act as an integrity check, ie, to attest non-tampering, it is commonly used in conjunction with the SHA as a HMAC (Hash-keyed Message Authentication Code) and hash, or HMAC-SHA. Application of the MAC and SHA in conjunction provide a sound integrity and authentication checking process, which can also incorporate an additional key during the device authentication, or key exchange process. By implementing an additional key, the integrity and message authentication can run as a single process completely separately to the confidentiality process - with high security of their own. The HMAC-SHA functions are considered necessary in every data exchange, and will be analysed as a pair, in order to find the safest but most energy-conserving configuration. The HMAC-SHA is available to the ESP32 through mbedtls in all the aforementioned SHA functions, as the HMAC can be a SHA auxiliary.

The function calls presented above will also be brought forward to be tested using power analysis for responding to research questions two and three.

4.2 Research questions two and three

- How much could energy consumption be reduced during device authentication?
- How much could energy consumption be reduced during subsequent data exchange?

Responding to questions two and three require all the available TLS functions belonging to the mbedtls library to undertake energy sampling. Questions two and three have been grouped together in methodology and empirical chapters since they will both employ energy sampling under Ohm's law, in the contexts of authentication and data exchange, to compare the energy consumptions of typical TLS and re-configuration. Firstly, energy sampling under Ohm's law within the ESP development environment is introduced, followed by authentication using entropy and key exchange variables. The ciphersuites are then introduced as segregated CIA attributes, as a measurable variations for data exchange - again by typical TLS and re-configured TLS for the application domain. Finally the experiments are outlined for the process of measuring each function and corresponding variables.

4.2.1 Energy sampling under Ohm's law within ESP-IDF

Whereas in typical TLS application authentication is generally mutual, the edge devices will be responsible for the majority of the processing as part of the robust network design in order to prevent bottlenecks - and so energy analysis will take place on the ESP32 edge devices only. As a result, keys are invoked on the ESP32 and passed over to the server, or hub, to add to a 'keyring' in an asymmetric approach.

The ESP IoT Development Framework (ESP-IDF) [163], employed the Arduino library as a component. This allowed high control of the ESP board through the IDF's 'menu-config', in order to switch off the WatchDog Timer (WDT) [164]. Menuconfig → Component Config → Common ESP related → Disable Interrupt WatchDog, Disable Initialise Task Watchdog Timer on Startup. This was important to ensure that the board did not enter 'boot loop', or interruption, during testing in high repetition cycles - preventing the board from completing each task. The micros() function from the Arduino library returned the timing of any part of the sketch. Micros() was therefore used to time each function; key generation, signing, verification, the CIA functions. Time can be printed to the ESP32 serial port, displayed within the development environment and connected by a USB data cable.

An example of the micros() timing function is demonstrated below:

```
unsigned long time; void setup() {
   Serial.begin(115200);
}

void loop() {
   Serial.print("Time:");
   time = micros(); //prints time since program started
   Serial.println(time); // wait a second so as not to send massive amounts of data
   delay(1000);
}
```

Power can be sampled using an oscilloscope [165], to accurate levels appropriate for protocols such as BLE [166], and has demonstrated previous results in a similar domain of vertical farming [167]. Voltage, measured by microvolts, µV, multiplied by current, represented by I, and measured by Ohms, Ω, gives power - represented by milliwatts, mW. A 5V power bank was used to ensure consistency, in conjunction with the same resistor. By multiplying power by time, measured by microseconds, µS, we can ascertain energy in Joules, J.

4.2.2 Measuring Entropy and Key Exchange Protocols

When undertaking key generation, the ESP32 will employ the Random Number Generator (RNG), hosted as part of its dedicated hardware-accelerated cryptography block. This generator is called by 'ctr drbg' in the mbedtls library - a deterministic random number generator function. To gain a fair sample and eliminate anomalies, ten readings for each protocol will be recorded, from which an average time will be deduced as an example of 'normal' consumption time, whilst disregarding the first reading due to boot load. The results sets for testing authentication protocols are presented in section 6.4.

An example of measuring a sample of entropy using the micros() function in the context of ECDHE key exchange is demonstrated below in pseudocode:

```
1. Begin contexts
2. Seed ECDHE RNG
3. Set up client context for key pair
4. Generate ECDSA Signature
5. Compute ECDSA message signature hash
6. Set up server context for key pair
7. Server reads client key and computes secret, signs Server ECDSA
8. Client reads server key and computes secret, signs Client ECDSA
9. Check if both computations are equal, verifies Server signature,
    verifies Client signature
10. Free all contexts
```

Pseudocode for functions tested under power analysis is attached as Appendix A.

4.2.3 Measuring CIA Configurations in Data Exchange

No entropy generation is required for the CIA. The functions are also much less computationally-demanding than authentication functions, thus allowing a larger number of loop iterations. These experiments are outlined below and the results are also presented in section 6.4.

4.2.4 Comparing typical TLS with re-configured TLS by Authentication and Data Exchange

Below the experiments for authentication and subsequent data exchange are presented for results and comparison in Chapter 6. Hopefully, results will demonstrate how the DPIA outlined substantial areas in which energy consumption severity ratings could be reduced from typical TLS implementation to the proposals of re-configuration. Ideally these will be severity categories - ie, from critical down to high or medium, or from high or medium to low.

Reference to single and multiple keys below are in reference to segregating ciphersuites into three individual CIA components, thus enabling selective use. Whereas typical TLS will use a single key as part of an all-in-one ciphersuite, the re-configuration exemplifies the use of three keys, maximising selectivity of the CIA attributes. All-in-one modes are explained in section 2.5, The CIA.

Below the set of energy analysis attributes are presented, where full code is attached as Appendix A:

1. **Key pair generation, sign and verify:**
   (a) **RSA:** 5 key strengths.
   (b) **ECDSA:** 11 curves.
2. **Key exchange:**
   (a) **RSA-PKC:** Single key (typical TLS), and multiple keys (re-configured TLS).
   (b) **ECDHE:** Single key (typical TLS), and multiple keys (re-configured TLS).
3. **Integrity and message authentication functions:**
   (a) **HMAC:** 4 variables in 64 bytes payload.
   (b) **HMAC:** 4 variables in 512 bytes payload.
4. **Full CIA by AEAD and collective components:**
   (a) **AES128:** 5 modes in 64 bytes payload.
   (b) **AES128:** 5 modes in 512 bytes payload.
5. **Data exchange:**
   (a) **From server to client:** Fully protected GCM and collective components, typical TLS.
   (b) **From client to server:** Selective components and excluding privacy, the re-configuration.
6. **Energy Consumption Comparison Over 24-hours in Authentication and Data Exchange:**
   (a) **Readings:** 16 character readings comparing typical TLS and the re-configuration.
   (b) **Instructions:** 256 character instructions comparing typical TLS and the re-configuration.

4.3 Research Question Four

- Can a lightweight, self-healing consensus mechanism, be proven to be safe?

Responding to question four requires formal verification for the safety of a tiered authorisation model. Unlike research questions one two and three, answering question four does not include any type of comparison exercise between typical and re-configured TLS. This is because TLS has a binary approach towards access control or authorisation - as soon as the secure channel is set up, a participant either has access or they do not and there is no tiering between in or out. As the proposed re-configuration of TLS is domain application driven, relationships between edge devices (the ESP32 microcontroller), and the hub (Raspberry Pi or similar microcomputer), are persistent - greater emphasis is placed on business continuity and the possibility of insider attacks than typical TLS. As a result, a tiered model with a series of contingencies will be modelled, with varying privilege levels to prevent complete downtime.

Firstly the design is modelled around the domain requirements, then it is run through formal verification.

4.3.1 Designing the authorisation model

Following the findings from research questions one to three, a design will be deduced to reuse existing cryptographic functions. This is important to not duplicate data that will already exist after security setup, and also to not add unnecessary additional cryptographic functions that perform the same tasks. The results for these findings are presented in section 7.3.

4.3.2 Notating the model

The model design will be formally specified in terms of messages exchanged between cryptographic operations performed between parties and their roles. Appropriate high-level notation is explained below, as the Cryptographic Key (CK) model and adversary, flow charts, and employing Alice and Bob. Once modelled, appropriate analysis can be performed using automated tools.

1. **CK, Dolev-Yao, and the Adversary**

   The attack model, or threat model, defines the capabilities of potential adversaries. In the case of the Cryptographic Key (CK), or the Dolev-Yao model, the adversary is assumed to have complete control over the network with abilities of interception, modification and message injection. Therefore, the Dolev-Yao also assumes the perfect state of cryptographic primitives - without the correct key, such an adversary cannot decrypt or encrypt messages or form a signature. This CK model is fundamental in the realm of formal verification for cryptographic protocols, and is used to examine the robustness of a mechanism under 'worst-case' scenarios, using manipulation of honest parties and injecting data exchanges.

Specifically for a granular authorisation mechanism, it is crucial to ensure that an adversary cannot escalate privileges or access protected resources without permission. Formalising the protocol for analysis under the Dolev-Yao model can gain confidence in the longevity of protocol security, and provide properties such as freshness and integrity.

2. **Flow charts**

   Cryptographic protocols often involve complex interactions between parties which can be difficult to understand without simplistic, high-level illustration. The design will first be illustrated by flow charts representing how devices are authorised into the network using the symbols deduced from the cryptographic protocols in the results of chapters 5 and 6. Following the flow charts for each segment of the consensus mechanism life cycle, Alice and Bob notation can outline the formal notation.

3. **Alice-Bob notation**

   Alice and Bob notation, or message sequence charts, represent a high-level yet intuitive method for representing cryptographic protocols, predominantly using the names of sender and recipient Alice and Bob respectively. There are many other characters including Charlie, Eve, Heidi and so forth but the majority of scenarios utilise the primary couple A and B, with arrows indicating the direction of communications or actions taken between them. Where Alice sends Bob an encrypted message, this would read:

   ```
   A → B : {M}K
   ```

   Where A is Alice, B is Bob, the parenthesis indicate encryption of the M for Message under the K for Key.

4.3.3 Formally verifying the model

The next stage of the process is to convert the Alice-Bob notation into language required for two verification tools presented below - AVISPA + SPAN, and ProVerif, where full code is attached as Appendix B:

1. **AVISPA + SPAN**

   The Automated Validation of Internet Security Protocols and Applications (AVISPA) platform was used in conjunction with the Security Protocol ANimator (SPAN) tool for formal verification of the algorithms described in the activity diagrams. The aim was to ensure secrecy in certain attributes, whilst using as few security functions as possible in exchange for identifiers or privacy functions.

The Dolev-Yao threat model was used to test the secrecy and freshness of values sent between devices, guarding against eavesdroppers and intruders seeking to gain knowledge over a deliberately insecure channel. The protocol was written in High Level Security Programming Language (HLSPL), and loaded into the SPAN interface within AVISPA. SPAN provides four backends for verification:

(a) **OFMC:** [168], On-the-Fly Model Checking performs protocol falsification and bounded session verification. OFMC is demand-driven and uses a number of symbolic, constraint-based techniques such as the lazy intruder, and constraint differentiation, for typed and untyped protocol models.

(b) **CL-ATSE:** [169], Constraint-Logic-based ATtack SEarcher takes a protocol input using the AVISPA compiler language Intermediate Format (IF), and models all reachable states to determine whether an attack is possible under the Dolev-Yao intruder model.

(c) **SATMC:** [170], SAT-based Model-Checker considers typed protocol models, providing falsification and bounded session verification using a SAT solver. This is a satisfiability solver which takes a Boolean input, and outputs results based on whether the variables can confirm it is true. SATMC are not ideally used for trees.

(d) **TA4SP:** [171], Tree-Automata-based Protocol Analyser performs unbounded verification by approximating intruder knowledge using tree languages as opposed to strings, based on tree automata [172]. Secrecy properties in the model can be revealed as flawed by under-approximation, or safe throughout any number of sessions by over-approximation.

SPAN also provides three types of animation to simulate the protocol itself, the methods of an intruder, and a successful attack. If the proposed authorisation model is determined as safe by all, if not most the verification backends, then an attack simulation will not be possible.

Exemplified below is the conversion of Alice-Bob notation to CAS+, which AVISPA + SPAN then converts to HLPSL:

(a) `A → B : Ka`

(b) `B → A : {{SSID, Kas, Kb}Kb-1}Ka`

(c) `A → B : {{Na}Kb}Ka-1`

(d) `B → A : {{Na}Ka}Kb-1`

```
protocol DA;
identifiers
A, B : user;
KPa KPb : public_key;
SSID, Kas, Na : number;

messages
1. A -> B : KPa
2. B -> A : {{A, B, SSID, Kas, KPb}KPb'}KPa
3. A -> B : {{A, B, Na}KPb}KPa'
4. B -> A : {{Na}KPa}KPb'

knowledge
A : A,B,KPa,KPb;
B : A,B,KPa,KPb;

session_instances
    [A:alice,B:bob,KPa:kpa,KPb:kpb];

goal
secrecy_of Kas [A,B];
secrecy_of Na [A,B];
```

Alice-Bob notation and CAS+ is attached as Appendix B.

The safety results of the four formal models are presented in section 7.3.

2. **ProVerif**

   ProVerif is a formal verification tool which employs the applied pi calculus process to model and analyse security protocols. A ProVerif model of a protocol is written in the tool's input language, typed pi calculus, divided into three parts of declarations, process macros, and a main process.

   In terms of its back-end verification process, ProVerif represents protocols by Horn clauses and applies resolution to these clauses to deduce information, automatic for a large class of protocols.

   The process comprises of:

   (a) **Input reading and parsing:** ProVerif reads an input file which contains the protocol description, queries to be answered, and the definition of functions and predicates. This file is parsed and converted into an internal format that can be processed further.

(b) **Generation of Horn clauses:** The protocol's behaviour is represented as a set of Horn clauses. These Horn clauses describe all possible actions of the protocol's principals and the attacker.

(c) **Resolution:** Horn clauses are processed using a resolution method. This is an inference method which can deduce new information from the given set of Horn clauses.

(d) **Query evaluation:** The queries specified in the input file are evaluated. These queries could ask if a certain property holds for all executions of the protocol, such as secret confidentiality or whether a specific authentication property is satisfied.

(e) **Reporting:** The results of the query evaluations. If a property is found to be true, it means that the property holds for the protocol under the assumptions of the Dolev-Yao model. If the property is found to be false, this means that an attack was found and a sequence of actions has led to a violation of the property.

Presented below is an example of translating Alice-Bob notation to ProVerif inputs:

(a) `A → B : Ka`

```
(* Declarations *)
free c:channel.
free skA:bitstring.
fun pk(bitstring):bitstring.

(* Protocol *)
let agentA(skA:bitstring) =
    let pkA = pk(skA) in
    out(c, pkA).

let agentB() =
    in(c, pkA:bitstring);
    0.

(* Process *)
process
    ( !agentA(skA) | !agentB() )
```

ProVerif inputs for the balance of the formal models is attached as Appendix B. The safety results of the four formal models are presented in section 7.3.

Chapter 5

Energy-Driven Risk Assessment

The previous chapter proposed a series of aims and objectives to follow for four experiments, the collective answers of which will confirm or deny the hypothesis. This chapter considers the answer for research question one:

- What are the security and privacy requirements of the aquaponics smart-farm?

This chapter is ordered by the eight objectives outlined in Chapter 3, 'Aims and Objectives': scoping, key findings, data mapping, function impact and severity, reduction and substitution, objective scoring, assessment and comparison.

For each objective, reference is made back to the literature review. With an energy-centric focus, this is an exercise to ascertain threshold protection in order to reduce time, power, processing and storage consumptions as much as safely possible. The resulting Data Protection Impact Assessment (DPIA), aims for a hybrid between protection, both security and privacy, and energy-conscious application.

This metrics framework is designed to predict the consumptions of energy based on data exchange protection levels, and is useful at both the design phase of a security solution to dynamically trade-off attributes prior to implementation for optimum performance. In this research, it is this DPIA which acts as the solution's security reconfiguration of TLS - the ciphersuite components optimised in order to reduce energy consumption as much as possible whilst maintaining adequate protection. The DPIA scoring system is finally applied to TLS as it exists, and against the new solution as a prediction of how much energy can be relatively saved. If the scoring system demonstrates clear differences, the test results in subsequent empirical chapters will reflect valuable difference as predicted in the DPIA results.

5.1 Scoping the Domain

This section refers to the first section of literature review part 3, 'SSM from a Business Perspective'. We consider three modelling methods, BATWOVE, TOWS, and inductive elements from the most common risk-driven methodologies used in security design.

5.1.1 CATWOVE (CATWOE plus BATWOVE)

This is a top-level approach for scoping - an application from the principles of business studies not dissimilar to the Business Model Canvas (BMC), [173], which has been used for ‘building blocks’ for IoT business cases [174], and enterprise ecosystems, architecture and services [175]. CATWOE is often used in conjunction with the Unified Modeling Language (UML), [176; 177; 178], including design applications of cloud services [179], and information systems for farming [180; 181]. Whereas BMC justifies stakeholder and financial viability of IoT modelling in a business context, CATWOE as a top-level methodology, and UML as a process-defining methodology, produces the basis of functionality. Therefore a CATWOE and BATWOVE approach for redesigning the TLS process would be suitable, and UML for modelling for the authorisation system also seems sensible. For simplicity, CATWOE and BATWOVE have been combined to produce an encompassing ‘CATWOVE’. Below is the CATWOVE analysis:

1. **Customers: Who are the beneficiaries of the business process, how and why would they be affected by the security design?**

   Beneficiaries include consumers and their green interests, investors, farmers and food retailers. End consumers of farmed crops purchased from supermarkets will have an interest in purchase prices and sourcing. As consumers increasingly become aware of green farming and factors such as water use, fertilisers, fuel and other environmental impacts of traditional farming, the preferences are moving further towards responsible and local practices. The influencing aspects of organic and responsibly-sourced crops are closely related trust in the provider for nutritional benefits and environmental kindness [182]. This is a strong business case for aquaponics practice to serve the trust and willingness of the end-consumer - where systems can benefit from green energies and symbiosis without backup electricity or fertilisers, reducing the overheads of a security solution is pertinent to optimise consumer interests.

   Carrying this example through to investors, many businesses have recognised the development of environmental concerns, and ‘green marketing’ has accelerated to attract the financial interest and sales of this emerging market [183]. Advertising promotes local, organic and sustainable farming with few fertilisers and as little waste wherever possible, and is increasingly pushing plant-based diets. In addition, as part of the data storage behind agricultural supply chain records [184], particularly concerning traceability [185], Distributed Ledger Technologies (DLT), are starting to emerge with greener consensus mechanisms [186]. Where Proof-of-Work (PoW), has been the mining choice for Bitcoin’s success both for investors and ownership [187], greener alternatives for the same PoW are emerging [188], in addition to more energy-efficient models such as hybrid Proof-of-Stake (PoS) [189].

   Trust from green practice and accountability is also of great benefit to local producers and supermarket supply. By engaging in green practice as an emerging strategy and away from traditional agricultural resources, farmers can produce more crops per acre and require less input materials [190; 191; 192]. Aquaponics also provides a controllable and predictable environment to optimise crop yields and does not suffer from unpredictable weather or fire risk.

In summary, the consumer desire for more efficient plant farming is increasing, and optimising the energy consumptions, such as with a greener security solution, benefits the whole supply chain financially. Lowering energy consumptions of the security required for smart farming contributes towards the feasibility of the model - and much more so if it were to rely on green energy. If a farm could balance solar and wind energy by minimising energy consumptions, this would be very advantageous.

2. **Actors: Who are the people involved in the situation who will undertake the work?**

   Those involved in the implementation of the aquaponics environment, and those impacted by the success of the security solution includes the person setting up the system and farm workers harvesting the crops. This could be the same person(s), as in traditional farming, and what the impact of an energy-efficient security solution could be on them. For farm business owners and operatives, the financial benefits are the same as the business processes for their customers: consistent environments, higher yield, non-indigenous crops, semi-automation, predictable harvest dates, less waste and fewer resources such as fuel, plant and land.

3. **Transformation: What processes of inputs becoming outputs does the security fulfil?**

   The aquaponics network is data-centric. A series of ESP32s report environmental data to a central hub which is used for monitoring, issuing device updates, and configuring actuators remotely. This hub may well be the only part of the network that connects to the internet, whether hard-wired or by mobile data to upload content to a cloud-hosted database.

   This networking arrangement should be centralised, as the one-to-one nature of the BLE range of nominally 100 metres promotes the clustering of systems by range. If the farm is large and dispersed, central hubs such as the Raspberry Pi microcomputer can operate as 'key rings' for the secure collection of data beyond the range of a single, centralised hub or WiFi connection. Given the nature of clustering in this respect, the range of a large farm built of hundreds of systems will never run out of range given the lack of restrictions offered by BLE.

   Regardless of the network topology, security should be set up to be completely decentralised, in order to rid the network of X.509s, multiple TLS libraries, and the other research challenges discussed in the previous chapter. Therefore, each device must undertake the security processing for its own aquaponics system and send readings to the Pi over a protocol suitable for an IoT network such as LoRaWAN or BLE. Using this approach, security undertakings are governed by many devices surrounding a hub, or edge processing, and not the hub itself. This approach for outsourcing security to edge-processing both alleviates issues related to scaling, and central vulnerability.

The challenge in outsourcing security to edge devices is to mitigate the protection of all edge devices wherever safely possible - so that the battery life of each ESP32 is not put at risk by unnecessary protection whilst processing at the network edge. The challenge in creating an authorisation system is to ascertain the trustworthiness of a device belonging to a network, with assurance of group validation from previously authorised devices.

4. **Worldview: What is the big picture, and what are the wider impacts of stakeholder beliefs?**

   Although this research has a focus on security, it could inform a business case for aquaponics, based wherever possible, on reliability and consistently high crop yields. A good security solution will promote long battery life and business continuity, whereas a poor security solution will challenge the strengths of a constrained network, and bring about issues with scalability with downtime and latency. The solution, although in this research bespoke for an aquaponics network, exemplifies a design for universal application in the IoT domain, and all constrained, battery-powered devices.

   The 'big picture', therefore, is to present a measurable application of energy and risk in the application domain for benchmarking optimum practice. Ideally, energy consumptions proposed by the model should bolster the viability of the full aquaponics system, ie, to reduce consumptions as much as possible whilst maintaining confidentiality, integrity and availability od data. In addition, preventing unauthorised device access to the network given a strict granular access control system should bring confidence to stakeholders for continuity throughout malicious attack and device failures.

5. **Owner: Who owns the process or situation being investigated, and what role will they play in the transformation, positive or negative?**

   The Internet Engineering Task Force (IETF), operating under the auspices of the Internet Society (ISOC), are those responsible for developing and promoting internet standards including those related to TLS. Additionally, the roles of the National Institute of Standards and Technology (NIST), and the Federal Information Processing Standards (FIPS), whilst neither directly govern TLS, provide guidelines and recommendations for cryptographic standards, which are often used in conjunction with TLS to assure secure communications.

   Fortunately, the cryptography block of the ESP32 allows the stipulations of typical TLS regulations to be bypassed when implemented for edge processing whilst protecting payloads. As a result, the main concern of the implementation should not be how secure encryption is, but how long the device can live whilst operating an optimised TLS configuration. If this is considerably longer than typical TLS implementation, then there is a viable business proposition, and the farm owner, or investor, should utilise it over standard TLS. If the difference in device battery lifespan is negligible, then there would be no viable business case nor use to any network owner.

6. **Victims: What or who could suffer detriment as a result of intervention or transformation?**

   In contrast to the benefits of a novel solution to the Owners, a poorly designed, or inappropriate transformation of TLS into a lighter form could also be to their detriment. This is where the content-driven risk assessment justifies how appropriate the solution must be in terms of security and energy use, to prevent downtime as a result of attack (poor security), or drained devices (no energy). Ultimately this downtime leads to poor crop yield, unwell fish, financial issues, reputation damage and loss of customer assurance in the domain as a model. If the Victim becomes the customer, the aquaponics model as a driver towards green and sustainable farming fails at a conceptual level - and invites back the issues associated with more traditional farming as a carbon-positive footprint.

7. **Environment: What are the constraints, rules and policies that will impact solution success?**

   There are notable criticisms towards TLS and PKI [193], which encourages the development and approval of edge processing on a payload basis - particularly where energy is such a pertinent consideration, and processing power is so scarce. The aim of our project is to adjust the configuration of TLS to more appropriately support an IoT network - taking advantage of the fact that IoT standards are generally not enforced [194]. Working on the basis of lacking standardisation, where Grace Hopper's famous idiom "forgiveness is easier acquired than permission", we can circumvent most of the environmental constraints by proof of concept.

To conclude the CATWOVE study, an IoT-centric, energy-conscious security solution is very much a purposeful activity for the viability of an aquaponics domain and green farming. A model developed to justify the carbon-neutral, and perhaps the carbon-negative, movement is pertinent at this time for the IoT in general, and sustainable farming specifically to exemplify such a model as part of this research. We will now proceed to investigate the areas in which vulnerabilities may occur in theory, before an objective study into metrics.

5.1.2 SWOT and TOWS Analysis

Strengths, Weaknesses, Opportunities and Threats (SWOT), is a classic business strategy tool and TOWS is a variant of the same acronym to extend SWOT from an internal environment to an external one [195]. The aim of this model is to draw domain-specific attributes such as communication ranges and architectural details, which will impact the attributes of the metrics framework later on.

The SWOT analysis presents the following attributes:

1. **Strengths:** The things that the domain performs well at, unique resources, and what others view as advantageous.

   Internal strengths include the lack of a browser, reducing the threat landscape, independence from the internet due to alternative protocols, and hardware-accelerated cryptography functions promoting dispersed processing away from a single central unit. External strengths include a range of communications available from the outside world, including but not limited to the internet, and a general awareness of the significance of green farming in general.

2. **Weaknesses:** The things that could be improved on, the areas in which resources are fewer than rival domains, and the areas perceived as vulnerable.

   Internally to the network, battery life, particularly when compared against data exchange frequency, is the main issue. When compared against the demands made by the TLS standard, this is a considerable network weakness. Externally, a lack of consistent and reliable solar and wind resources do not promote a self-sustaining battery life based on purely green energy.

3. **Opportunities:** Opportunities available to the application, trends which could be taken advantage of, and opportunities from which strengths could arise.

   Internally, the selective and appropriate use of cryptography, data exchange, privacy functions and protocols provide a broad range of configurations. Externally, technologies such as solar, wind, and piezoelectric generation can be used in conjunction with each other simultaneously.

4. **Threats:** Things that could harm the domain, undertakings of rival projects, and threats which expose weaknesses of the domain.

   Internally, sabotage by authorised users with ulterior motives present a concern that cannot be resolved using the authorisation system under any TLS configuration. The network may also suffer downtime from mechanical or energy failure. Externally, malicious attempts and downtime by attacks aimed towards the various communications protocols are concerns.

The TOWS analysis brings these attributes together to form strategies outlined in Table 5.1.

The outcomes of which are proposed below:

1. **SO, Maxi-Maxi:** Protocols used by the IoT such as BLE, MQTT and LoRaWAN, are lightweight. As a result, a combination of reducing computation by selective cryptography, utilising green energy generators, could sustain the network without mains electricity. In conjunction with the carbon dioxide absorbed by crop production, this could become a carbon-negative environment.

Table 5.1: TOWS Matrix

-	**External Opportunities(O)**	**External Threats(T)**
Internal Strengths(S)	**SO "Maxi-Maxi Strategy"** Strategies which use strengths to maximise opportunities.	**ST "Maxi-Mini Strategy"** Strategies which use strengths to minimise threats.
Internal Weaknesses(W)	**WO "Mini-Maxi Strategy"** Strategies which minimise weaknesses by taking advantage of opportunities.	**WT "Mini-Mini Strategy"** Strategies which minimise weaknesses and avoid threats.

2. **ST, Maxi-Mini:** The use of short-range communications protocols reduces the threat landscape to that limited range. Use of the internet opens the network to threats of a global scale, whereas the use of BLE for example, restricts the threat to anyone outside of a 100m range at most.

3. **WO, Mini-Maxi:** With the exclusive one-to-one device relationships supported by BLE, the threat landscape becomes even smaller. Each device owns its own public-private key pair and long-term relationships promote the notion of infrequent authentication, as relationships are static until the system fails. As a result, the relationships stay persistent and use little power in re-authenticating.

4. **WT, Mini-Mini:** This is about defensive strategies to minimise loss rather than promote success. The notion of 'adequate' or 'threshold' security is emphasised here to maintain a good balance between power and computation, where security is lowered as much as safely possible, whilst battery life benefits as much as possible by incurring the least feasible computation.

From here we can present a series of key findings, to later on interpret into design specifications.

5.2 Key Findings

From the literature review, requirements analysis, and SSM study, we have deduced:

1. **BLE is a reasonably short range (100m), protocol and reduces the attack vector to local scale.**

   Since BLE operates within a very short range, connections are not globally accessible as they are with mains infrastructure such as the internet.

2. **There is no need for a browser, nor an OS on the ESP32.**

   Where the microcontrollers will undertake the majority of the processing on-board using native hardware acceleration, ie, processing at the edge of the network, data can be delivered free of an operating system and browser following initial key agreement.

3. **X.509 certificates can be substituted by signing and verification functions.**

   The key component to certificates in the signature, and correlating verification by which to establish authentication. Without the bulk of the certificate, authenticity can still be verified whilst also removing the large and centralised PKI underneath.

4. **Payload-based encryption allows for TLS configuration and processing at source.**

   Processing at the network 'edge' promotes the use of hardware acceleration native to the ESP32 - offloading most of the responsibility from the cluster hub such as a Raspberry Pi. This means energy consumption is dispersed amongst the majority of devices within the network rather than relying on a hub, whilst utilising all the mechanisms belonging to TLS in a manner more suitable to IoT - such as selective confidentiality.

5. **A content-driven DPIA will help alleviate unnecessary encryption.**

   The reconfiguration of typical TLS strengths and inclusions can be challenged under the analysis of a design-phase DPIA. During this analysis, a proposed energy-saving version of normal TLS will demonstrate a more suitable version for IoT, based on specific data exchanges of the aquaponics domain.

6. **One-to-one relationships using symmetric encryption can use shorter keys than asymmetric.**

   Following initial key setup, the ESP32 edge devices can maintain long and persistent relationships lasting months or years over the BLE protocol. During these relationships, the smaller shared symmetric key can be used to protect data - as opposed to resetting the key agreement for every few data exchanges as with standard TLS use.

7. **All processing can take place on edge devices to further reduce and complicate attack target.**

   A further advantage to edge processing is the distribution of keys amongst the majority of network devices. Should a cluster hub experience downtime, edge-processing from the ESP32s are proposed to establish a new, secure connection with the reset or replacement hub. This means that during times of downtime the edge devices will continue to operate with the instructional code already available, and to pause data reports to the hub until remedied. This further reduces the attack vector from one or a few cluster hubs to many edge devices.

8. **A tiered authorisation system is important to eliminate internal threats.**

   A granular access control system offers levels of protection beyond the binary 'in-or-out' method used in TLS. By segregating responsibilities and read-write rules between devices based on their histories of trust, a hierarchical approach can protect against insider attacks and downtime by utilising inaccessible backup devices.

5.3 Data Mapping

Here we identify the data that must be the main content of the design, given the mechanisms that can be kept and those which can be eliminated, and the actual data exchanges that the mechanisms are designed to protect.

By order of most sensitive to least sensitive, below is all the environment content:

1. **Private Keys:** Private keys of the Public-Private key pair of device authentication should be a long-standing, impenetrable set for delivering data with guaranteed confidentiality. This is part of key exchange for device authentication and operates under either RSA or ECC.

2. **AES Keys:** These keys are not as long-standing as private keys and are used to fulfil confidentiality as part of symmetric encryption of data exchanges such as readings and instructions between devices. AES keys are considerably shorter than public or private keys, require more frequent reset, and should remain secret. The key is kept on both devices belonging to a one-to-one relationship, and so there are two copies of every AES key which are both replaced on session refresh.

3. **HMAC-SHA Keys:** These keys ensure integrity and authentication of each message. Generally when a TLS session is negotiated, the ciphersuite automatically includes a SHA attribute for a one-way hash function as a receipt of non-tampering of the message, and a HMAC for authenticity check of origin. This key performs the same operations, but separately, as it can therefore operate independently of an AES key. By separating the functions, security is configurable at a lower level. As with AES, this key is placed on both devices within a one-to-one relationship and must also be refreshed with new sessions.

4. **Initialisation Vector (IV):** This key is for hardening session ciphers. It is used in a particular AES mode and promotes an element of randomness in encryption so that ciphertext outputs will always be different, even when the input is the same. The IV is a starting variable, typically random or pseudorandom, but at least unpredictable or unique for each session.

5. **Instruction Code:** This is a chunk of content to be hosted by each ESP32 controlling an aquaponics system, specific to the requirements of crop type for controlling lights, pH level, humidity, flood and drain timing and so forth. It will be a reasonably sized chunk of code and will likely vary between systems.

6. **Readings:** Environmental readings, such as water temperature, air humidity, light levels and so forth do not need confidentiality, because they are feeding back single or multiple digits assuring the data control centre that it is within a predetermined error margin. Readings themselves do very little other than inform us of a consistent environment in which fish will thrive and crops will yield to their full potential - but to

prevent the actuators from disrupting the environment, the readings must be true. This means that they do not need confidentiality, but do require an integrity and authenticity function - assuring the system that readings are correct, not tampered with, and from the expected origin. Readings therefore require SHA and HMAC functions.

7. **MAC Address:** This is an identifier unique to each device but available to any public BLE GATT [196] scanner, and which can easily be spoofed. The MAC address is useful to scan a device for functionality and location, but for delivering messages, should be used in conjunction with other cryptographic attributes such as asymmetric public and private keys, or symmetric data exchanges to ensure authenticity. The MAC address is a requirement in BLE data exchange in the same way that IP addresses enable TCP/IP, and so it must be included. For quick scans and network checks, they can be a convenient alternative to cryptography key identifiers.

8. **IP Address:** Similarly to the MAC address, IP addresses can be scanned and spoofed easily, but are useful in the same way for the TCP/IP stack as with the BLE/GATT stack - to ascertain device presence quickly. Following network scanning, devices can be communicated with using much more private means, such as the private keys. If the network is completely free of the internet, an IP address will not be required.

9. **Public Keys:** Are the counterparts of the public-private RSA and ECC key pairs, designed to be given out in order to encrypt messages meant only for the decryption of the recipient device. However, public keys should be accompanied with authenticity functions such as digital signatures, to make sure the sender device is part of the network, rather than a malicious node masquerading as one.

10. **Digital Signatures:** In conjunction with signing and verification functions, the Digital Signature Algorithm (DSA), ascertains authenticity during the key exchange phase between devices. It is included in the X.509 certificate as the proof of certificate source, and can therefore be used independently of certificates for data accountability and authentication.

With reference to Chapter 4, ’Research Methodology’, the scoring system from the OWASP risk rating methodology [89], was used as a basis by which to deduce a series of domain-appropriate factors. Whereas the OWASP system gauged on a scale of 0-9, this was adapted into a 1-4 criteria from low, medium, high and critical impact respectively.

With regards to supplier trust and transparency as outlined in the CATWOE and TOWS study, business impacts are very important. Even though financial data and so forth is not at risk, the well being of the fish and perceived quality of the operation would be considered as important to the green movement and the welfare-conscientious as animal rights are to those promoting free range, healthy livestock. However, the significance of business impact factors will always output as maximum damage, at the highest score, and bias the differences between DPIA results.

5.4 Reduction and Substitution

This is a further addition to confidentiality, where encryption functions can be substituted by privacy functions. The idea is to investigate whether privacy is a lightweight alternative to encryption, and if so, introduce privacy functions to reduce power consumption whilst retaining confidentiality.

Although the aquaponics system is largely devoid of personal and sensitive data belong to human subjects (we shall assume the fish don't mind), then the usefulness of the GDPR pertains to assessing a 'data-oriented' or 'content-based' attack risk, where, in circumstances of zero risk, then there may also be zero privacy. Following the GDPR principle of 'necessary processing of data and nothing more than absolutely necessary', the challenge is to reduce protection of content, and subsequent processing, wherever possible.

GDPR privacy functions can be realised in several ways:

- **Data masking:** [197], Refers to disclosure of data with modified values. This could be character shuffling, tokenisation, encryption, or character substitution - one of the oldest forms of cryptography. Substitution is where a value is exchanged for another value, to make identification or reverse engineering difficult. This character set will be the same throughout the policy, but may turn into a centralised process.

- **Pseudonymisation:** [198], is data de-identification that substitutes private identifiers with false identifiers or pseudonyms. "John Smith" could be changed to "Mark McConnell", but this would require some form of centralised database.

- **Generalisation or minimisation:** [199], The removal of data to prevent identification, such as a house number of an address, so accuracy of data remains whilst not being too specific.

- **Perturbation:** [200], The use of round-numbering and adding random noise to modify a dataset. The set of values must be proportionate to noise to avoid disturbance, making this method quite complex.

- **Synthetic data:** [201], uses algorithm-generated information with no relation to the real case. These datasets represent patterns from real data by invoking standard deviation, linear regression, medians and other statistical methods. It is of no use to our comparatively simple model.

- **Permutation:** [202], refers to data swapping or shuffling rearranges dataset attributes so that they do not fit original information, often by switching columns in databases.

These privacy functions will provide a scale of alternative functions as part of the confidentiality element of the CIA triad, and demonstrate the greatest energy savings in the design solution.

5.5 Domain-Specific Criteria

With influence from the OWASP risk rating methodology presented in section 4.1, the criteria specific for the aquaponics application domain are discussed below. Whereas OWASP operates a scale of 0-9, the new sets will use 1-4, attributing low, medium, high and critical risk severity levels respectively. This removes any ambiguity when allocating scores from an input perspective, to return consistent results. There are three sets of domain-specific criteria supporting a total of nine variables for assessment: security application, energy consumption, and privacy substitution. When applied to the ten environment content items listed earlier in section 5.3, the DPIA will produce risk severity of each one.

5.5.1 Security application factors

This set considers what kind of basic protection the data attributes require compared with what they are worth as a reward from an exploit. It is a good way of comparing typical TLS and internet applications against IoT by applying TLS-standards to deduce a result and compare it against the IoT environment, to determine which properties could be lightened or removed given the flexibility.

1. **Protection:** How much protection is required for the content?

   This attribute describes protection requirements within data exchanges such as instructional code and fish tank readings - the environment data content listed in section 5.3. Wherever possible, security functionality will be substituted by privacy functions as with the tokenisation mechanisms pertaining to PCI-DSS, exemplifying lightweight alternatives to cryptography computation.

   From an IoT perspective, fish tank readings are insensitive and need not be fully protected, because it makes no difference if threat agents can see them or not. However, readings must be assured of integrity for non-tampering, and authenticity as proof of origin. Digital signatures can be delivered in plaintext with zero protection, as they represent proof of origin. In these examples, the two most common data transmissions require no encryption - a huge energy saving to the IoT system, and a benefit to re-configuration that is not available when employing typical TLS.

   (a) **1:** Plaintext, no CIA or GDPR required.
   (b) **2:** Integrity and availability required.
   (c) **3:** Integrity, availability and some privacy required.
   (d) **4:** Integrity, availability, and guaranteed confidentiality required.

2. **Value:** How much value is placed on success of an exploit?

   This attribute represents the value of the exploit reward, compared with the level of security it should have to protect it. Low value material need not employ heavyweight protection, and ideally all data that can be transparent should be, to reduce computational costs.

   (a) **1:** No value at all.
   (b) **2:** Value in conjunction with other data exploits.
   (c) **3:** Reputation and business disruption.
   (d) **4:** Full downtime, financial reward, personal kudos.

3. **Range:** What is the communications range of the protocol?

   This is unique to the application domain in terms of IoT protocol communications range, deduced from the SSM earlier. Since most of the communications will take place within a small and restricted environment, the threat landscape is not representative of global applications, and this further reduces the need for heavy security.

   (a) **1:** 1:1 relationships in a small range ¡100m.
   (b) **2:** 1:1 relationships in a large range ¡10km.
   (c) **3:** M:N relationships in a medium to large range ¡1000 nodes.
   (d) **4:** M:N relationships in a global range and unlimited nodes.

5.5.2 Energy consumption factors

This set considers the frequencies of content processing, security strength and sensitivity of environment content. As a rule of thumb, the more ephemeral the data, the less heavy the protection need be - but TLS contradicts this by applying top security for all transactions.

1. **Frequency:** How frequently is data processed?

   Frequencies of data processing is rated 1-4, from lowest energy consumption to highest respectively.

   (a) **1:** Weeks and months.
   (b) **2:** Days and weeks.
   (c) **3:** Hours and days.
   (d) **4:** Seconds and hours.

2. **Security strength:** What security strength will be applied, according to typical or re-configured TLS?

   This variable describes the symmetric, asymmetric, and Elliptic Curve Cryptography (ECC), security tiers by currently acceptable strengths.

   (a) **1:** None.

   (b) **2:** AES-128, RSA-3072, ECC-256, less strong and less costly.
   (c) **3:** AES-192, RSA-7680, ECC-384, quite strong and less costly.
   (d) **4:** AES-256, RSA-15,360, ECC-521, very strong but very costly.

3. **Sensitivity:** What is the intended exposure to the public?

   This attribute describes the sensitivity of data against business impact, and should be assessed on the assumption that data has been attained for malicious intent.

   (a) **1:** Designed to be public.
   (b) **2:** Not designed to be public, but of no consequence.
   (c) **3:** Designed to be private, and could be critical with other data.
   (d) **4:** Designed to be secret, and will be critical.

5.5.3 Privacy substitution factors

This set is intended to provide potential differences in energy consumption by re-configuring typical TLS security functions to privacy functions wherever practically possible, without jeopardising the safety of the domain. Low outcomes demonstrate little difference in energy use, but attention should be paid to medium, high and critical results, where a large amount of energy can be saved.

1. **Encryption:** How much strength reduction is possible?

   This attribute represents energy savings based on reducing encryption key strengths. At the time of writing, an AES-112 key is safe for 15 years, and AES-128 for considerably longer [203]. In this instance AES-112 has been included as a factor for reduction, but following good practice for longevity, AES-128 will be used for testing as the smaller key throughout this research.

   (a) **1:** Reduce to bit-strength AES-112, RSA-2048, ECC-224.
   (b) **2:** Reduce to bit-strength AES-128, RSA-3072, ECC-256.
   (c) **3:** Reduce to bit-strength AES-192, RSA-7680, ECC-384.
   (d) **4:** None.

2. **Privacy:** How can computation be reduced by strength of privacy function?

   With reference to section 5.4, presented below are the privacy functions used as lighter alternatives to security functions as demonstrated in the tokenisation of PCI-DSS.

   (a) **1:** Data anonymised by removing attributes.
   (b) **2:** Plain text without privacy.
   (c) **3:** Data masking, substitution, or permutation.

   (d) **4:** Pseudonymisation or tokenisation.

3. **Computation time:** How much computation could be reduced by changing frequency of processing environment content?

   This represents changes in data frequency, where attributes such as device authentication can be extended given the permanent nature of device relationships. This is a different scoring system to protection of data, where rather than short-lived data being safer and cheaper, the reduction is placed on using it for longer periods of time in order to process it less.

   (a) **1:** Weeks and months.
   (b) **2:** Days and weeks.
   (c) **3:** Hours and days.
   (d) **4:** Seconds and hours.

5.6 DPIA Comparison

Ideally scores of low will be shown throughout the re-configuration outcomes. Medium outcomes are acceptable, high requires attention and critical equates to running the system towards downtime given poor energy management.

Table 5.2 shows energy consumption using typical TLS:

Table 5.2: Typical TLS results.

Data Attribute	**Ref**	**Security Application**	**Energy Consumption**	**Privacy Substitution**
Private keys	1	4	4	4
AES keys	2	4	4	4
HMAC-SHA keys	3	4	4	4
IV	4	3.33	3.50	3.50
Instruction code	5	3.33	3.50	3.50
Readings	6	3	3.25	3.25
MAC address	7	3	2.5	2.5
IP address	8	3	2.5	2.5
Digital signatures	9	4	3.25	3.25
Public keys	10	4	3.25	3.25
Average Score	-	3.567	3.375	3.375
Severity	-	**Critical**	**High**	**High**

5.6.1 Security application

With typical TLS, security application is what it is, and there is no flexibility - the data is either sent using TLS and is under full protection, or no security is applied at all. As a result, every result is under full protection. The values vary, as some data ranges from public to secret, and the range of technologies can change such as MAC used for BLE. Range does not make a difference when using TLS, because the assumption is a global scale network of unlimited nodes. All scores return as high or critical.

5.6.2 Energy consumption

Critical scoring has been saved by the nature of the data, and not the actions of TLS. Readings and MAC addresses owning public status lower the score of energy consumption, but it is still a high score, and that means a short lifespan for battery or green-powered networks.

5.6.3 Privacy substitution

The keys could all be reduced to recomputing every few months to reflect the strength of AES. Reducing the cipher to 112-bit would be beneficial; despite the preemptive warnings from TLS that 112-bit will soon be inadequate, it is still safe for several years, the threat landscape is so small without a browser or global access, and atypical communications protocols such as BLE. In summary, TLS is very high in security and energy consumption, but also very high in mitigation options given the ability to re-configure attributes.

Table 5.3 shows energy consumption using the re-configured attributes for the aquaponics domain application:

Table 5.3: Re-configured TLS results.

Data Attribute	**Ref**	**Security Application**	**Energy Consumption**	**Privacy Substitution**
Private keys	1	3	2.75	2.75
AES keys	2	3	2.75	2.75
HMAC-SHA keys	3	2.33	2.25	2.25
IV	4	2.33	2.25	2.25
Instruction code	5	2	2	2
Readings	6	1.33	1.25	1.25
MAC address	7	1	1	1
IP address	8	1	1	1
Digital signatures	9	1	1	1
Public keys	10	1	1	1
Average Score	-	1.8	1.725	1.725
Severity	-	**Critical**	**High**	**High**

5.6.4 Security application

The beauty of energy consumption, security and privacy by design is that the consumption can be assessed before implementation. Here security application is the highest consumption of energy because the keys require full CIA, and without it there would be full downtime. However, the nature of other data has changed from full protection to optional privacy (instruction code), and no confidentiality or privacy (readings). The range of device communications has also changed from a presumption of global threat to a small network less than 100m between pairs of devices.

5.6.5 Energy consumption

Severity scores requiring encryption are higher than others because encryption is computationally expensive. The score is still an overall low, and leaves little room for improvement.

5.6.6 Privacy substitution

The keys could all be substituted with forms of pseudonymisation or tokenisation, and with an extended lifespan given the strength of AES. Reduction was not available for any attribute since re-configuration indicated a value and sensitivity assessment pointing to the many attributes not requiring full protection. This consistency is a good indication that security and privacy by threat-base modelling offers benefits to the IoT environment.

In summary, the scores were low throughout, and there was little energy reduction ability given the benefits offered by the framework earlier on, whereas TLS returned critical and high, the IoT network returned low.

Chapter 6

Energy-Conscious Authentication and Exchange

Discussions and results for research questions two and three are presented together here because they both require an analysis of the energy consumptions of functions belonging to two separate mechanisms, working together to fulfil the authenticity of two devices and establish safe communications between them. Both mechanisms must be studied in a measurable objective way, and within the limitations of the mbedtls library calls to address:

- How much could energy consumption be reduced during device authentication?
- How much could energy consumption be reduced during subsequent data exchange?

The aim of question two is to determine the lightest method of authentication available to the ESP32, using the mbedtls library, without the support of X.509 certificates. The aim of question three is to determine the most suitable method of data exchange available to the application domain. The difference with this question is the intricate nature of incorporating payload volume and selective functions according to the DPIA results for each type of data. This is where the DPIA will also be tested for accuracy - how the predicted consumption differences realise themselves in practice, between TLS and the novel solution.

The mbedtls library provides a C/C++ library based on the function available to TLS. Fortunately, where TLS has strictly enforced the use of strengths and techniques deemed safe, mbedtls allows many legacy functions such as shorter key strengths. This provides a dimension of freedom unavailable when following regular TLS stipulations.

This chapter presents discussions surrounding challenges pertaining to typical TLS in IoT application, including expectations of the domain, centralisation in regards to certificates, and application-appropriate security. Related work is then presented in the contexts of similar solutions including self-signed certificates, and certificate-less security. From these discussions, attributes can be deduced to outline to goals of the re-configuration design. Finally, following the testing methodology outlined in section 4.2, results are presented and discussed to compare typical TLS against re-configuration.

6.1 TLS Challenges

The successes of smart-farming rely heavily on IoT infrastructure. For sensors and actuators to operate for a long period of time correctly and continuously, downtime and interruptions have to be as close as possible to zero, and this is where security takes the stage. Security infrastructure can be draining on energy resources and leave networks vulnerable to a single point of failure, but security in general ensures reliable communications between pairs of devices.

The issues pertaining to security are that of a 'balancing act' [204], where integration is unattractive for two main reasons. Firstly, the value of security is not perceived until loss [205], and so if the domain is considered to be of little value to attackers in general, or beyond the chances of being targeted because they are considered 'low impact', then security can be seen as adding costs without benefits. Secondly, the predefined ciphersuites available to replicate the de facto security standard TLS, are tremendously draining on constrained devices. For this reason security is considered a challenge [206], as TLS needs infrastructure, energy and processing - three things that contradict the nature of constrained IoT devices and networks.

TLS is strong and universally accessible to both constrained and traditional devices and their protocols - but the energy consumption and centralisation are problems. Established TLS security practice has revolved around firstly utilising a negotiation process known as a 'handshake', authentication to create a client and server relationship, and secondly, invoking a secure session by which to exchange data secured by a symmetric key.

Here we consider how to address the issues surrounding TLS and PKI [207]. Such issues include reducing the high energy consumption of TLS client and server authentication for subsequent data exchange, and removing the central vulnerabilities of this model, whilst providing security for any communications protocol the network will use. We consider the problem background as three parts; expectations, centralisation, and application-appropriate security.

6.1.1 Expectations

The biggest problem of securing constrained devices is the unrealistic expectations of what a microcontroller should be able to process, compared with a perception of adequate security. Now, since TLS traditionally secures machines supplied with constant power, high-bandwidth telecoms and excellent processing capabilities, it has developed in parallel with those provisions. However, it is more appropriate to compare microcontroller capabilities with machines typical of the 90's, and their applications to suit. Constrained devices are equipped with the protocols and power supplies to support the operations of a smart-farm environment - the ESP32 provides suitably small packet sizes and BLE to communicate them. Therefore security considerations should not limit choices between a power and processing trade-off, but how TLS can be adapted to function effectively within an environment reflecting 'obsolete technologies' [208].

6.1.2 Centralisation

TLS employs X.509 certificates to bind keys with identities. The management of these certificates in undertaken by a Public Key Infrastructure (PKI), typically consisting of five parts: Certificate Authority (CA), to store and issue X.509 certificates; Registration Authority (RA), to verify entity identity; a central directory to securely store and index keys; a management system for access and distribution; and a certificate policy to state procedures and rules. This is a cumbersome and delicate series of operations criticised for codependency and vulnerabilities due to the centralised data stores [91]. PKI is also storage-intensive, populating lists for both current and revoked data, causing a scalability issue for IoT. Certificates are available to view by visitors within the browser, however, revoked certificates have been displayed as current in some browsers, and have been identified as a security flaw [92] - further disproving the value of the X.509 system. Key escrow for certificates also undermines privacy [93], and has been described as placing “keys under doormats” [94]. Official bodies are granted access to TLS communications for law enforcement using PKI facilities [95], with escrow hosted on servers and retaining key ownership. Although it is possible to operate TLS without certificates [96], there is no alternative fulfilment for verifying the identity of the server by which you, as the client, are contacting - and therefore without employing TLS there is no assurance of security.

In summary, even with the heavy implementation and energy consumptions of encryption for privacy, there are still privacy issues present. In the case of IoT applications where the majority of data readings are benign, unnecessary protection need not be applied, and greater savings can be made.

6.1.3 Application-Appropriate Security

Applying TLS to any protocol comes with the same caveats as traditional internet regulations. Although DTLS [209], MQTTS [210], BLE-TLS [211], and µTLS [212], are lightweight IoT applications, the restrictive and impractical stipulations are still the same. In addition to the X.509 infrastructure, this includes expensive privacy enforcement, and a library for each protocol. Confidentiality, integrity and message payload authentication are enforced by the secure channel TLS creates between the client and server. In many ways, this is positive. Confidentiality guards against eavesdropping, integrity ensures non-tampering, and authentication proves the message is from the intended origin - but privacy is not always necessary. Using TLS for any protocol has this same issue. Following authentication, TLS uses a shared symmetric key to encrypt and decrypt data transferred between the two devices. This key is used by a cipher, most commonly the Advanced Encryption Standard (AES), enforcing unnecessary protection over insensitive data.

In addition, even the smallest recommended AES key strength, 128-bits, cannot currently be broken in less than around a million years. A final thought towards the threat landscape of the application is use of protocols for communications between the edge devices and the server or central gateway [213] - beyond the 100 metre range of BLE, it is inaccessible. Compared against the global accessibility of traditional internet communications, the risk likelihood is very low. For a decentralised or even distributed heterogeneous network of IoT devices, a scheme providing a 1:M, or M:N relationship between devices and their respective keys [214], would have been ideal. However, the mbedtls library native to the ESP32 does not currently provide such freedom, and so we must proceed this study with cryptographic primitives based on a 1:1 client-server relationship.

6.2 Related Work

Self-signed certificates and certificate-less security are considered as related work, where they enable the elimination of a central PKI.

6.2.1 Self-Signed Certificates

There are two ways by which to authenticate the client and server devices in the TLS connection using X.509 certificates. The first and most traditional way is to send the device's Public-key and a Certificate Signing Request (CSR), to a trusted CA, and the second is for the client (ESP32), to act as a CA by generating a root certificate itself and signing it using a Digital Signature Algorithm (DSA); providing verifiable ownership and thus, authenticity. Self-signed certificates have not typically been effective on browsers, since they come with a warning sign that the certificate cannot be trusted due to potential interception between client and server. However, the IoT application will not use a browser for aquaponics readings, and all edge devices will have a trusting relationship with the server since they are part of the same smart-farm. This is very different from interrogating an unknown web server possibly on the other side of the world. Using self-signed certificates, and the DSA, removes many problems of centralised TLS and PKI security.

6.2.2 Certificate-Less Security

On the note of using self-signed certificates, certificate-less key escrow schemes [215], have demonstrated good operation in IoT using the DSA included in the certificate, but with a partial private key to perform the signature. This model is positive towards an alternative solution for TLS, since the devices are capable of creating key pairs and signatures locally, thus enabling the edge devices to accept responsibility for all secure data exchanges. Outsourcing cryptographic processing to edge devices contributes towards scalability, robustness, and decentralisation, with the additional benefit of energy efficiency - the ESP32 hosts dedicated hardware acceleration for four cryptography functions.

Blockchain has demonstrated success in numerous decentralised security applications including cryptocurrencies [216; 217], and IoT [218; 219]. Blockchain is a type of Distributed Ledger Technology (DLT), comparable with a simple bookkeeping ledger where the weighting of profit and expenses is agreed between two parties, and the verification then becomes a block of data added to an ever increasing chain [220]. Applications have been scalable and robust alternatives to the centralised PKI, used in industrial processes [221], farming marketplaces [222] and harvesting [223]. There are two obvious problems with blockchain. Firstly the use of dedicated nodes to hold the smart contracts or consensus mechanism [224], and the full chain [225] - and secondly the continuous mining of the chain for verified transactions, as this quickly exhausts battery life [226; 227], creates complexity, and subsequent security issues [228]. The chain could be stored on the single Pi server, but that creates a similar type of vulnerability to PKI unless there were backup chains beyond the network, or additional nodes.

6.3 Design Attributes

Design attributes are goals which will ideally transform into novel contributions. The aim of this section of research is to reconfigure the use of TLS for a more domain-suitable security application:

1. **Privacy:** Protecting private data whilst allowing selectivity of privacy for insensitive data.
2. **Certificate infrastructure:** Removing X.509 certificates and their structure, without detriment to security.
3. **Session timeout:** Extending persistent secure channels between entities to reduce energy use.
4. **Individual protocol security:** Removing protocol-specific security infrastructure.
5. **Energy consumption:** Reducing time and power used in authentication and data exchange.
6. **Edge processing:** By distributing the processing, the network will be scalable and robust.
7. **IoT-appropriate ciphersuites:** Selective configuration of CIA attributes for energy benefits.
8. **Additional keys:** Additional session keys for separate use of integrity and authenticity.

6.4 Experimental Results and Discussion

This section will prove or disprove the effectiveness of the new configurations by comparing the results of the experiments against regular TLS implementation.

6.4.1 Authentication Results: Key Pair Generation, Sign and Verify

Repetitions of 100 loops per task were performed to eradicate errors of single units, but produce results within a reasonable time - loops of 1000 proved too high an expectation of the ESP32, particularly in RSA - causing the 'guru meditation' [229], or stack overflow error (despite disabling the WDT), so all experiments undertook 100 loops to give a fair test. As with entropy assessments, the curves requiring more entropy (bigger key strengths), and iterations of small loops of 10, took considerable time.

Table 6.1: 1a) RSA key pair generation, sign and verify

RSA Key Bit-Strength	**2048**	**3072**	**4096**	**7680**	**15,360**
Joules	1.390	4.664	11.532	113.350	Error
Millijoules (mJ)	1390	4664	11,532	113,350	Error

Table 6.2: 1b) ECDSA key pair generation, sign and verify

ECDSA curve	**SECP224R1**	**SECP256R1**	**SECP384R1**
Joules	0.160	0.231	0.316
Millijoules (mJ)	160	231	316
ECDSA curve	**SECP521R1**	**SECP224K1**	**SECP256K1**
Joules	0.526	0.246	0.285
Millijoules (mJ)	526	246	285
ECDSA curve	**BP256R1**	**BP384R1**	**BP512R1**
Joules	2.557	4.811	9.769
Millijoules (mJ)	2557	4811	9769
ECDSA curve	**Curve448**	**Curve25519**	-
Joules	Error	0.057	-
Millijoules (mJ)	Error	57	-

RSA demonstrated a phenomenal difference compared with ECDSA - the lightest RSA energy consumption was around 24 times that of elliptic curve mutual authentication, despite the dedicated RSA hardware. It was therefore pointless to continue testing RSA in conjunction with asymmetric key exchange. Curve 448 demonstrated no entropy at all, repeated the same key many times, and so was considered an error in the library.

6.4.2 Authentication Results: Key Exchange

For one key (typical TLS), and three keys (re-configured TLS).

PKC employing RSA consumed around five times more energy than the elliptic curve mutual authentication model (Table 6.4).

Table 6.3: 2a) RSA-PKC key exchange

RSA-PKC	**RSA in 2048-bit strength PKC with ECDSA for AES 128-bit key**	**RSA in 2048-bit strength PKC with ECDSA for three keys**
Joules	1.450	1.622
Millijoules (mJ)	1450	1622

Table 6.4: 2b) ECDHE key exchange

ECDHE	**ECDHE in Curve 25519 with ECDSA for AES 128-bit key**	**ECDHE in Curve 25519 for three keys**
Joules	0.283	0.284
Millijoules (mJ)	283	284

6.4.3 Data Exchange Results: Integrity and Message Authentication Functions

CIA functions were run in iterations of 10,000 loops.

Table 6.5: 3a) HMAC 64 bytes payload, and 3b) HMAC 512 bytes payload

HMAC-SHA224	**64 bytes plaintext**	**512 bytes plaintext**
Joules	0.023	0.063
Millijoules (mJ)	23	63
HMAC-SHA256	**64 bytes plaintext**	**512 bytes plaintext**
Joules	0.014	0.025
Millijoules (mJ)	14	25
HMAC-SHA384	**64 bytes plaintext**	**512 bytes plaintext**
Joules	0.016	0.026
Millijoules (mJ)	16	26
HMAC-SHA512	**64 bytes plaintext**	**512 bytes plaintext**
Joules	0.016	0.026
Millijoules (mJ)	16	26

6.4.4 Data Exchange Results: Full CIA by AEAD and Collective Components

With reference to Table 6.6.

6.4.5 Data Exchange Comparison

From server to client, Table 6.7. From client to server, Table 6.8.

Table 6.6: 4a) AES128 64 bytes payload, and 4b) AES128 512 bytes payload

AES128-GCM	**64 bytes plaintext**	**512 bytes plaintext**
Joules	0.0281	1.352
Millijoules (mJ)	28	135
AES128-CBC-HMAC-SHA256	**64 bytes plaintext**	**512 bytes plaintext**
Joules	0.019	0.043
Millijoules (mJ)	19	43
AES128-CTR-HMAC-SHA256	**64 bytes plaintext**	**512 bytes plaintext**
Joules	0.019	0.055
Millijoules (mJ)	19	55
AES128-CFB8-HMAC-SHA256	**64 bytes plaintext**	**64 bytes plaintext**
Joules	0.047	1.212
Millijoules (mJ)	47	121
AES128-CFB128-HMAC-SHA256	**64 bytes plaintext**	**64 bytes plaintext**
Joules	0.019	0.055
Millijoules (mJ)	19	55

Table 6.7: 5a) From server to client

Reading length in character bytes	**AES128 in GCM mode (mJ)**	**AES128 in full CCM (mJ)**
16	0.022	0.022
64	0.038	0.022
128	0.058	0.023
256	0.099	0.023

Selective components, excluding privacy, typical of the re-configuration.

GCM mode, encouraged in TLS, used energy at an increased rate when compared to CCM (Table 6.8). At smaller data exchanges, GCM was in fact much lighter - but short exchanges represent readings to the aquaponics systems (as opposed to instructional code, and therefore full protection was not required anyway - so the saving was irrelevant.

Table 6.8: 5b) From client to server

Reading length in character bytes	**AES128 in CCM without privacy (mJ)**
16	0.019
64	0.021
128	0.021
256	0.023

6.4.6 Energy Consumption Comparison Over 24-hours in Authentication and Data Exchange

Table 6.9: 6a) Readings (of 16 Characters) and 6b) Instructions (of 512 characters), Comparing typical TLS and the re-configuration (Total Energy Consumptions over a 24-hour period in Millijoules mJ)

Payload characters	**ECDHE in Curve 25519 with ECDSA for AES 128-bit key with readings sent in GCM**	**ECDHE in Curve 25519 with ECDSA for AES 128-bit key with readings sent in CCM**	**ECDHE in Curve 25519 for three keys and readings including privacy, Pi to ESP32**	**ECDHE in Curve 25519 for three keys and readings excluding privacy, ESP32 to Pi**
16	7075.56	7075.488	284.488	284.473
512	7077.463	7075.448	284.448	284.42

GCM and CCM employed by TLS show such a drastic difference because of the full authentication process undertaken on every reading compared with selective AEAD (Table 6.8). If one reading is sent every hour, and each reading requires device authentication before it can exchange data, the process is going to show the energy consumption of authentication 24 times. With the re-configuration, the authentication process is undertaken every 24 hours. Between full protection and removing privacy within the re-configuration, there is a difference of 0.015 millijoules. This does not sound much - it is around 5.475 millijoules a year. However, comparing the GCM of TLS over a year, at 2,582,579 millijoules, against not using privacy on the basis of this re-configuration, at 103,832 millijoules a year, this shows a decrease of annual usage by around 96% (Table 6.7 and Table 6.8).

A similar situation applies to instruction set sizes, at 256 bytes. Over a year, following this example, typical TLS using GCM mode shows an energy consumption of 2,583,274 millijoules, and the re-configuration gives a consumption of 103,813. In this scenario, the re-configuration also illustrates a saving of 96%. If authentication was undertaken every week or month compared to TLS authenticating every hour, this would increase to 100's, and 1,000's of percent in energy consumption saving - potentially changing the feasibility of security as part of a very constrained project.

6.5 Contributions

The security objectives have been transformed into the following contributions:

6.5.1 Privacy

Protecting private data whilst allowing selectivity of privacy for insensitive data. Where TLS employs PKI, key escrow allows the unauthorised browsing of data exchanges by authorities who wish to view it, such as the government. This may not be preferable as the 'snoopers charter' created quite a disturbance [230]. In contrast, benign data such as temperature and pH levels which need not be protected are under confidentiality enforcement as part of the TLS ciphersuite model. To rectify this unsuitability for the aquaponics setup, the re-configuration contained no key escrow, and proposed the separation of security attributes at the authentication stage to permit subsequent selectivity. As a result, insensitive data will not use unnecessary battery power, and unauthorised observers will be disallowed as part of the decentralised model made possible by edge device processing.

6.5.2 Certificates

Removing X.509 certificates and their structure, without detriment to security. Removing certificates from the infrastructure enables authentication to operate without a centralised architecture. This is both more robust than having a single point of failure, and more scalable. As with certificate-less schemes, signing and verification can be used for proving origin without the full certificate. Asymmetric ECC cryptography was used for sign and verify functions, utilising ECDSA as an effective replacement for a full X.509.

6.5.3 Session Time-Out

Extending persistent secure channels between entities to reduce energy use. Since 2015, NIST has recommended an RSA key size of 2048-bit size, or the ECC equivalent of 224-bit [231]. At the time of writing, an RSA key of this size, or the equivalent ECC key, would take an estimated 300 trillion years to break by a classical computer, and in around eight hours using a quantum computer. Quantum computers are not expected for a few years, and even when commonplace, re-authenticating every eight hours rather than every few minutes would be advantageous to the smart-farm re-configuration. In addition, utilising a protocol that did not allow world-wide access such as the internet would further reduce the threat landscape. The results concluded that by using the ECC Curve 25519 rather than its RSA 3072 equivalent, that key generation, sign and verify functions alone can show the ECC as a 98.7% saving. Applied to authentication once every 24 hours rather than once a reading, these energy savings demonstrate a 96% difference when compared against modern TLS recommendations. Particularly with the persistent client and server relationships of the smart farm, resetting device authentication over periods of days, weeks or even months has proven to show 100s or 1,000s of percent.

6.5.4 Individual Protocol Security

Remove protocol-specific security infrastructure. The ESP32 has demonstrated its abilities to process cryptographic primitives and protocols very effectively on-board, and so security can be distributed to the edge devices within the smart-farm application. Apart from the benefits of a decentralised model, it also negates the need to employ TLS ciphersuites - and therefore the rules of ciphersuites can be ignored. BLE, LoRaWAN, MQTT, and any number of other protocols can be secured by payload, and also selectively. The re-configuration encourages selective privacy by separating AEAD functions available to the mbedtls library, enabling further energy consumption reductions on any protocol, without a library for each.

6.5.5 Energy Consumption

Reducing time and power used in authentication and data exchange. By testing all the adequately secure ECC and RSA key sizes, the results have shown that Curve 25519 in application of ECDHE, is the lightest available authentication process available to the ESP32. In addition, providing three session keys during authentication instead of the traditional AES key alone can provide enormous energy savings, which can be critical for smart-farm lifespan when relying on constrained devices.

6.5.6 Edge Processing

By distributing the processing, the network will be scalable and robust. Distributed processing from a central governance, or the hub, to the ESP32 enables a scalability for a larger network without latency or ‘build up’ on system operations. By placing the responsibility of security onto each respective device joining the network, the recipient hub device need only keep a ‘keyring’ of short and symmetric keys required for its own cluster. In addition, each cluster is limited by the range of its protocol, probably BLE - limiting the number of symmetric keys required.

6.5.7 IoT-Appropriate Ciphersuites

Selective configuration of CIA attributes for energy benefits. By removing the enforcement and complexity of the AEAD TLS requirement, particularly in GCM mode, the data exchanges can be relieved of unnecessary protection levels and lower the use of energy enormously. Judging by the figurative projections of the energy assessments, smart systems battery lifespan could be increased by months or years.

6.5.8 Additional Keys

Additional session keys for separate use of integrity and authenticity. In order to harden the CIA attributes, additional keys for IV and HMAC provided a tougher CBC mode in addition to selective privacy enabled by removing encryption, most regularly in readings. The figurative differences by payload length justified the issuing of additional keys at session invocation following the handshake, for a security system with more granular and efficient security than TLS.

Chapter 7

Energy-Conscious Consensus Mechanism

Research question four considers the design of a granular authorisation mechanism for the application on a decentralised network topology. It is to be in addition to the TLS function, which provides a default and binary 'in-or-out' approach to network access, based on the possession of a network key. The aim of such a solution is to avoid centralised governance, and as such, traditional access control methods utilising roles or rules are inappropriate. Therefore, the solution to research question four presents a consensus mechanism based on hierarchy and ephemeral roles for each cluster of devices, with a limited data chain repeated on each node in the cluster. The design is model-checked under formal verification and ascertained as safe.

- Can a lightweight, self-healing consensus mechanism, be proven to be safe?

The aim of this question is to consider how a lightweight authorisation mechanism, based on a distributed or decentralised network may provide a scalable and robust solution. The solution proposed features a novel consensus mechanism for access control in networks populated by constrained devices. The research explores existing mechanisms in the distributed technology blockchain, whilst limiting the scalability issues associated within distributed and duplicated data in Distributed Ledger Technologies (DLT) - particularly challenging for constrained devices such as the ESP32.

Security is a fundamental requirement to assert business continuity [232], and generally the interpretation of security refers to authentication between devices, and the subsequent secure channel for messaging following authentication. However, default authorisation generally allows any device onto a network given the correct password. This approach is relatively black-listing; participants are generally welcome if they have basic entry requirements.

Basic network entry requirements, such as a password, can be provided by any node or living subject with such knowledge, and this opens up the threat landscape enormously, such as to malicious bots [233]. Malicious bots permit the large-scale interference of decision making by targeted algorithms or network seizure. Perhaps a tiered method of device authorisation would be where participants are generally unwelcome onto the network if they have basic entry requirements, and required to undertake further interrogation. This is more comparable to white-listing than it is black-listing.

DLT, [234], such as blockchain, present valuable methods of decentralised data exchanges between multiple nodes [235], based on consensus between groups of nodes rather than individual subjects or devices. When consensus is reached, data will be accepted as part of a block and placed on the chain, accumulating as recorded transactions. Although accumulating data is not ideal for resource-constrained applications, the consensus mechanism is helpful for granular and autonomous authorisation; nodes run by policy can therefore govern and self-heal a network beginning with, but not limited to, network passwords.

We therefore propose a hierarchy-based authorisation model in which a consensus mechanism allows a predefined, limited number of devices, based on rank and network privilege, and unlimited devices without such privileges. Data chains are to be ephemeral and limited, based on clustering of those within range of lightweight protocols such as BLE [236], towards a decentralised model, as opposed to a distributed model. This hybrid security mechanism also aims to be agnostic, protecting heterogeneous messages and devices described as challenging [237], in a range of applications.

We proceed with the challenges surrounding the application domain with regards to authorisation systems, followed by a summary of the research aims. We then consider related works to influence a suitable authorisation solution, and how this solution would address the research aims and domain challenges. Finally we produce formal testing results using an automated tool set.

7.1 Authorisation Mechanism Challenges

The heterogeneous nature of IoT networks presents three concepts difficult to cater for: constrained power and processing in devices; traditional applications; and the challenges of heterogeneity.

7.1.1 IoT Constraint in Scalable Authorisation

When IoT devices are to be networked, the challenge is, as with authentication and data exchange, with power, processing, and storage [238]. Traditionally, when a network is built, there is a central server surrounded by nodes which rely on that server for decisions pertaining to data access - who can access and change the states of what data, when, and why. Such a system is referred to as Access Control (AC), a series of roles and rules, depicting read, write and execute privileges amongst data objects [239]. This central server is generally responsible for storing a lot of functional data that the nodes relay to it, and an IoT network is only different in the sense that it is designed to be autonomous, whereas typical computer networks have users at the end of them. There are further ideologies that can alter the way of things with IoT [240], setting it apart from a traditional network. Firstly, they can be independent of the internet because of low-powered protocols such as BLE, LoRaWAN, and MQTT [241]. This provides for off-grid placements of devices, beyond the range of regular WiFi reception - BLE has a range of around 100 metres, and much more than that when meshed. Meshing and autonomy promotes scalability, but discourages the use of a single server due to range, data growth, and a central point of vulnerability. Some topologies operate in clusters of devices with a central node of their own, working in small numbers of in-range devices to take advantage of the low-energy protocols and data distribution. There is also the more recent development of DLT, such as blockchains [242], hashgraphs [243], and Directed Acyclic Graphs (DAG) [244], which verify and store transaction data based on consensus mechanisms and smart-contracts. DLT are very robust because they are distributed, but the data duplication creates scalability issues and this would rapidly become the downfall of devices with such limited storage facilities as microcontrollers.

7.1.2 Traditional Applications

Most mature AC systems are based on roles or rules, and utilise Access Control Lists (ACL), and are not dissimilar from Relational Database Management System (RDBMS) [146], mapping usernames to privilege levels throughout known sets of data. ACLs present a few problems to the IoT network. Firstly they are central, and have to be kept somewhere that can be accessed by every node or microcontroller, then there's the requirement to know all the data available, and somehow map it to people or devices that will read, write, or execute it - even during the 1990's before the big data we process now, this was becoming an issue. Relational data management is difficult with IoT since it restricts autonomy, and becomes increasingly complex as new developments are integrated. Finally there's the duplication of data - for example if this list covers all possible relationships (for which some M:N relationships will require additional tables for normalisation) [147], this is a big expectation for limited storage space. We consider other, mature AC systems:

- **DACS:** [148], Distributed Access Control System, are rule-based, and operate on the identity of user credentials within the organisation. This type of account setup would apply to every microcontroller in a network, taking time and resources for what should be an automated, preferably self-healing, relatively generic node.

- **DAC:** [245], Discretionary Access Control, is also rule-based, and assigns access rights based on rules specified by users, utilising ACLs and capability tables - rows with subjects, and columns with objects.

- **MAC:** [150], Mandatory Access Control, is considered the most strict of authorisation systems, and uses hierarchy to apply granular levels of access within a varying number of tiers, or 'security labels' in tiers such as classified, secret or top secret. The MAC system also permits access on a 'need to know basis', reminiscent of the GDPR's 'necessary processing', and thus pertinent to the topic of privacy. MAC maps user accounts and credentials to security levels and departments using ACLs, and is notoriously inflexible regardless of conditions or emergency.

- **RBAC:** [246], Role-Based Access Control, operates on roles, rather than individual user accounts, using the 'least privilege' possible for the operative to fulfil the job, similar to MAC but not as strict.
- **PBAC / ABAC:** [247], Policy, or Attribute-Based Access Control, is whereby access rights are granted using policies to combine resource, user, object or environment attributes. This is a Boolean-centric model using 'if' and 'then' statements about who the request is made by, the action they request and for what resource. This model does not use pre-defined roles or lists.

Other applications include capability certificates provided by policy engines and ECC-based authorisation certificates [153], but utilising certificates requires some form of authority by which to negotiate terms such as signatures, origin, roots and leaves - a problem similar to ACLs and centralisation. Similarly, keys have been used to update access and permit certain users - but a centralised system must also be employed to generate and monitor fair usage [154]. [155] presented a hierarchical authorisation system in which users of similar privilege levels are placed in the same group. This type of trust-organised placement perhaps makes more sense in IoT, where greater trust correlates to older, probably original devices.

7.1.3 Challenges of Heterogeneity

The essence of IoT heterogeneity pertains to the differences between communications and capabilities of components belonging to the same network - a big challenge in the IoT domain is resolving relationships between devices, and adjusting our expectations [248]. End users' expectations of handheld and desktop devices, service provision and volume of content delivery have increased as the capacity of such have developed in parallel - but IoT development contradicts that model. IoT presents constrained power, processing and storage, and so opposes 'big data'. In many ways this is positive - moving away from a data deluge, away from servers, hardware storage and energy consumption are very environmentally beneficial, and may work towards changing our expectations of technology improvements towards a 'greener IoT [249].

With communications protocols such as Bluetooth Low Energy (BLE), Long-Range Wireless Area Network (LoRaWAN), Message Queuing Telemetry Transport (MQTT), Zigbee and numerous other examples, IoT protocols are designed to match the capabilities of the boards to which they are native. This is very helpful, since microcontrollers supporting BLE, LoRaWAN etc, are designed with those small packet sizes in mind - sensor readings and actuation code to change the environment around us in any application - domestic, industrial, agricultural and so forth. The caveat comes in when implementing security [250; 251].

TLS, has been the de facto standard for decades, but despite its widespread application, little has been covered in the dimension of authorisation. TLS is strong, and promotes public-key cryptography over applications in banking, e-commerce, ATMs, chip-and-pin in a series of exclusive 1:1 relationships. Perhaps this is why such little emphasis has been placed on AC. At first glance, the principles underpinning TLS practice would be most suitable for an AC system, considering the persistent 1:1 relationships between edge devices supporting aquaponics systems and the data hub - because those same functions, in some way, will support the message and authentication, without additional AC keys, identifiers and other cryptography implementation. However, adjustments must be made for a successful transformation over the state of TLS in a form required for potentially autonomous white-listing of new aquaponics systems. The demands of a larger setup with ideally little human interaction is large, and so although in essence we wish to retain the 1:1 mechanism of TLS in concept, the application must be quite different to the rigidity of traditional AC.

7.2 Related Work

This section considers literature that will influence the design. Considerations include privacy and security in the context of authorisation, appropriate applications based on topology, and the most popular consensus mechanisms.

7.2.1 Privacy and Security in the Context of Authorisation

Access control and authentication are two main challenges in security and privacy requiring an immediate solution in the heterogeneous environment of IoT [143], proposed a capability-based authorisation model in which a token provides trust and a session key is produced to authenticate using Elliptic Curve Cryptography (ECC).

Tokenisation has been introduced as ephemeral data for cryptocurrency transactions alongside various key exchange systems such as Diffie Hellman, RSA and the lighter ECC. Whereas tokens are not strictly cryptographic functions and are typically referred to as advanced pseudonymisation as a privacy function [144], they do limit the spread of data beyond a “need to know” basis, a fundamental concept of the GDPR [145]. Wherever possible in a constrained network, minimising the processing requirements of securitisation is beneficial - for power, time and subsequent carbon consumptions. As a result, privacy functions are preferable if they so address the minimal necessary processing data for a given task - and this leads naturally to hash functions [252], tokens [253], ephemeral chains [254], and minimising the use of multiple identifiers, such as using keys as identifiers. However, a network must still be secure, and this is where the use of keys is required at some point - either asymmetric or symmetric key exchange is used to encrypt data to a standard of security that cannot be guaranteed by privacy functions [255].

7.2.2 Network topology-appropriate authorisation

Centralisation pertains to governance and processing using a central authority, usually for encryption key escrow, device identifiers, intrusion detection, and review and approval of new network devices. Organisations have typically used centralised services to provide X.509 certificates for banking and shopping gateways, and this has been a safe approach for a number of years - appropriate for servers that operate from a single or small number of locations. Centralised infrastructures are however, energy inefficient, usually rely on internet connectivity, must be active and online in order to be used, and are rendered vulnerable by a single point of failure [90].

DLTs, have demonstrated robust and accessible networking for cryptocurrencies [156], data storage [157], asset tokenisation [158], and e-commerce [159]. Although DLTs are generally constructed in graph and blockchain form, they all accumulate data to be operational - and this inevitably requires more storage and energy to operate. Decentralised architectures introduce a hybrid design, in which there is no single point of failure, but a defined group of devices either shares the same data, or relies on its own central device. A decentralised infrastructure can be spread over a long range using any number of technologies with communicating central nodes being elected by voting systems using reputation, such as with P2P ranking such as for file sharing online [160]. A decentralised topology with limited device numbers in groups of nodes, and replicated, ephemeral chains operating on a lightweight consensus mechanism would provide a suitable hybrid topology for a constrained IoT network such as the aquaponics smart-farm.

7.2.3 Consensus Mechanisms

Consensus mechanisms are fault-tolerant processes used to achieve necessary agreement from a group for a single data transaction. The consent of multiple nodes then gives the permission for that transaction to reside as part of a data block on the chain. Below are examples of popular mechanisms:

- **Proof of Work (PoW):** [256], requires members of the network to solve mathematical puzzles to prevent anybody from occupying the system, is known as mining, and widely used in cryptocurrencies to prevent attacks such as 51% - but PoW is energy-intensive.
- **Proof of Stake (PoS):** [257], selects transaction validators based on proportion of ownership on the chain compared to other members, and although does not use extreme amounts of energy like PoW, it is criticised for ‘blockchain oligarchy’, and as such is vulnerable to the 51% attack.
- **Proof of Elapsed Time (PoET):** [119], where users wait a random period of time and the first to finish waiting attains leadership of the new block. PoET uses trusted code with trusted, known users on a permissioned, or non-anonymous, network.
- **Practical Byzantine Fault Tolerance (PBFT):** [258], is based on the Byzantine Fault Tolerance algorithm in the late 1990’s; a scenario in which General’s gather around an enemy city and must communicate without interception. They gather around the city and send one message a time through a messenger, relying on cooperation and coordination, ignoring unauthorised influence.

- **Proof of Importance (PoI):** [259], introduced a rating system to combat the 'oligarchy' issue of PoS in which the rich become richer. The rating system requires a minimum vested stake, a minimum transferred rate, a recent transaction bound by time limit, and valid transaction partners identified as separate users. The algorithm does not require complex computation, but rather a series of experience proofs to read through quickly.
- **Proof of Capacity (PoC):** [260], uses the mining node's hard-drive space to determine the mining rights on the network. Considered an improvement on PoW, it takes 40% of the time to produce a block by dedicating storage and processing before mining begins, and removes the mining conflict of the same puzzles by allowing different 'plot' routes for working through puzzles - so there is both hard drive plotting, and block mining.
- **Proof of Authority (PoA):** [261], uses identity as a stake for reputation-based consensus, which means validators are not staking currency, but their reputation instead. PoA blockchains are therefore secured by random validating nodes as trustworthy entities. In supply chains and hierarchies, PoA is considered an effective and reasonable solution.
- **Proof of Location (PoL):** [262], enables a device's physical location coordinates to be broadcast to the blockchain without relying on that particular device. Radio, GPS and BLE-enabled devices can assess the physical location of nearby devices, but must be encrypted for reliability.

Influences of 'necessary processing', privacy, decentralisation and minimal energy consumptions through lightweight consensus mechanisms influenced a design that could be later model-checked for secrecy.

7.3 Design and Model-Checking

With considerations of the problem and related work, the design proposes decentralisation, low consumption security functions, privacy principles, autonomous policy-oriented authorisation, and group consensus. This section presents the design model:

7.3.1 Decentralised

Decentralisation in preference to centralisation or full distribution. Ideally, the topology should avoid duplication of the same data between all nodes, but allow duplication between associated nodes. Decentralisation introduces robustness unavailable in centralisation, as there is no central point of vulnerability, without the scalability issues of full distribution.

A decentralised topology of clustered devices is proposed, where clusters are arranged by the BLE range of up to 50m (100m is the maximum range), and each cluster contains a series of devices divided by rank and privilege:

Table 7.1: Device Hierarchy

Rank	**Privileges**	**Devices in Cluster**	**Security**
1 Field Marshal	Read Write	1	Does not advertise MAC or public key
2 General	Read Write Execute	2	Does not advertise MAC, advertises public key
3 Private	Read-only	Unlimited	Actively advertises MAC and public key

This is a simple but granular design, beneficial for cloud and data governance [263], and requires three devices to operate properly; an active General, a passive FM for backup, and a Private to join the network who will then become an authorised member and next in line for promotion if a ranked device fails in the future. In a range of systems, the higher privileged devices are few enough to be realistic, but robust enough to form a group consensus beyond the powers of unranked, new, unproven devices. Generals may execute changes in privilege levels to unranked devices which will then become Privates, and eventually a General or FM if either experience downtime. The idea of a hierarchy is not to allow widespread privilege, but minimise the privileges wherever possible so that the number of higher-privileged devices remain intact, and the purpose of authorised devices in business continuity. Should the system suffer a majority loss, assuming that at least a General or FM survives within BLE network range, the whole authorisation system can self-heal.

Now, privilege levels are purposefully kept simple:

1. **Field Marshals:** the highest and most protected part of the rank. They can only be accessed by a General or other FM, and do not have the power to grant authority. This is to protect them from manipulation, since they are the highest in the chain. There is only one FM in every cluster, but they can transcend clusters as backup for each other.

2. **Generals:** neither the top of the hierarchy, nor vulnerable to every potentially harmful device. The ability to execute means they can grant the 'Private' rank identity to new cluster members, and so they require all permissions. There must be two Generals or more in each cluster as they cannot transcend clusters and may have several Privates to rank simultaneously. The term General is commonly used in cryptography literature such as in the aforementioned 'PBFT', and is an active role.

3. **Privates:** the newest members of the cluster, in unlimited quantity and with read-only privilege. They can be exposed to a threat landscape without risking the integrity of the network structure. They can hold position as any aquaponics system and have potential for promotion if either FM or a General suffers downtime.

7.3.2 Low Consumptions

Since there are many ways of implementing security between nodes, such as keys and certificates, this aim requires a consideration into the lightest application of security functions available on the ESP32. The objective here is for the least infrastructure possible and fewest messages between nodes for safe network access.

As the system continues in its daily life cycle, it will utilise the four functions hosted on the ESP32 for authentication and message security. Although the authorisation system will also use a selection of these functions, it will be a separate process than those required on a daily basis. Authorisation is expected to be infrequent compared with the other processes, and ideally without conflict of other security keys and sensor-actuator tasks.

The following functions are both native to the dedicated hardware of the ESP32, and mostly used in other areas of the security design, which means they require little additional energy:

- **MAC:** Media Access Card is the 12-digit BLE address given to identify each ESP32, or each aquaponics system, in the aquaponics farm. The MAC address can be changed to a meaningful label describing geographical location or crop type.
- **Public key:** used in asymmetric cryptography is order to encrypt a message intended only for that recipient, so that no other devices can decrypt it.
- **Private key:** used as part of asymmetric cryptography for proving origin of the message by providing a digital signature. This is important because for ascertaining the source of a message and ensuring that it has not come from an intruder.
- **Nonce:** a value used oNly-Once, and important for 'freshness' between the node being interrogated, and the node verifying the authenticity of the newcomer.
- **Blacklist:** the bank of declined public keys identifying attempted intrusions. This list should be cluster-specific, ephemeral, and hosted by a General.
- **Shadows:** shares in a symmetric key between two or more ranked members. This is to protect against granting authority without group consent.
- **XOR:** representing 'exclusive or', or the logical operation that is true only if its arguments differ, so that one is true and the other is false. Used to create shadows of shared secrets. Timestamps: allow the authorisation model to determine the most appropriate replacement for lost entities by the length of time the others have served the system for.

7.3.3 Private

Where the GDPR expresses the notion of 'necessary processing', the network should refuse any request for data access beyond the minimum required for processing. Anonymisation, pseudonymisation and tokenisation are explored as lightweight alternatives to the heavier and more complex security encryption ciphers for enforcing consent management and access control [264].

Anonymisation is the process of removing personal identifiers that may lead to the identification of an individual. Now, there are no named individuals in the authorisation system, or the aquaponics smart-farm. However, it could be seen as wise, wherever possible, to disguise the identity of higher-ranking devices able to grant authority as they will undoubtedly attract attention. For this reason, in a cluster of several devices, the MAC addresses do not broadcast BLE information to join the network - they do not actively operate GATT server-client advertising. The ranked devices also pose as any other regular system, and they cannot be communicated with by an unauthorised device. Pseudonymisation is to replace any information which could be used to identify an individual, with a pseudonym. Again, although there are no individuals in the application, pseudonymisation is a useful way of quickly locating devices by their existing security and communications attributes. By using the MAC address as a location and crop reference that will only make sense to the farm operators, it provides an easy way of managing systems without creating additional data. Tokenisation is an advanced form of pseudonymisation, whereby a meaningful piece of data such as an account number is changed into a seemingly random number which has no value if breached. Tokens are useful to safely send over public channels, disguising their true value. In the authorisation design, tokens are used in the form of nonces for freshness to verify authenticity on a single-use basis.

7.3.4 Autonomous

Self-healing and automated management in the authorisation tiers are the objective for an IoT consensus mechanism capable of operating remotely. Policy or attribute-oriented authorisation systems have demonstrated good functionality using 'if' and 'or' logic. Policies can form the fundamentals of security models; write, execute permissions, replacing old data with new, and providing workflow. There are mature and well-known security models used in both military and civilian applications which focus on data confidentiality and integrity. These models are mature and operate on effective simple rule sets making them a suitable influence on IoT-oriented and constrained environments. Notably, these models are Take-Grant [265], depicting transference of rights, Biba [266], for preservation of data integrity, and Multi-Level Security (MLS) [267], providing classification. Take-Grant represents a directed graph, in which vertices are either subjects or objects, and the edges between them depict the rights towards the destination. There are two possible rights which occur in every instance, take and grant, forming four rules; subject takes rights of an object, subject grants own rights to another object, subject creates a new object, or subject removes rights it has over an object. This model is utilised when promoting devices in the event of a lost, existing one. The Biba integrity model is characterised by the phrase "read up, write down", and defines the rule that a subject of a given integrity level must not read data at a lower integrity level, nor write to a higher one. This model influences the protection of higher-ranked devices against lower-ranked. MLS demonstrates processing data between classifications of devices. Applied to the design 'ranks', and there is a hierarchy to prevent users from obtaining access to data for which they lack authorisation. Now, self-protection can be difficult to automate, and there are no strict guidelines on how to do this - so the design has formed a three-tier proposal by which sets of 'clustered' devices have limited authority. When one of those authoritarian devices falls, it is replaced by the most mature unranked device. This concept aims to maintain a limited set of backup whilst employing, but not trusting, any other - when group consensus has been obtained, of course.

7.3.5 Consensus

Group consent has been exercised in the field of cryptocurrency with enormous success. The same processes for validation influence the design proposal, but preferably without the cumulative data chains that accompany them. Attaining group consensus is far preferable to multi-party authentication, since secret sharing using eXclusive-or (XOR) [268], functions between a few devices uses considerably less energy consumption than escrow, particularly the X.509 infrastructure, Zero Knowledge Succinct Non-Interactive Argument of Knowledge (ZK-SNARKS) [269], or centralised trusted third parties such as Key Generation Centres (KGC) [270].

Where data validation is concerned, mining should be discouraged. In many cryptocurrency applications, mining exemplifies equality and robustness using the Proof of Work (PoW) mechanism. However, a domain in which clusters are contained within finite bounds of authority and geographic range, ownership is not such a concern, and Proof of Stake (PoS), can provide a more suitable template. The great disadvantage associated with PoS is 'blockchain oligarchy'. Many other consensus mechanisms exist of course; Proof of Elapsed Time (PoET), Practical Byzantine Fault Tolerance (PBFT), Proof of Importance (PoI), Proof of Capacity (PoC), Proof of Authority (PoA), Proof of Experience (PoE), as common examples. The objective of this exercise is to simplify and automate authorisation towards the lowest consumptions in time, power and energy available to this specific IoT application. This is possible by reducing processes, maths, and avoiding complex procedures such as learning algorithms or reputations systems.

7.3.6 Authorisation life cycle (Figure 7.1)

Below is an overview of the authorisation algorithm. It describes any device joining the network, from zero members to unlimited. From here, the integrity of the cluster will be attested by the number of ranked devices being checked. If there are not enough ranked devices at each of the three hierarchy tiers, the algorithm will automatically add them by promotion to a rank as devices accumulate to the network. When the ranks are full, they will remain in this state until clusters begin to realise beyond range, or when a device fails.

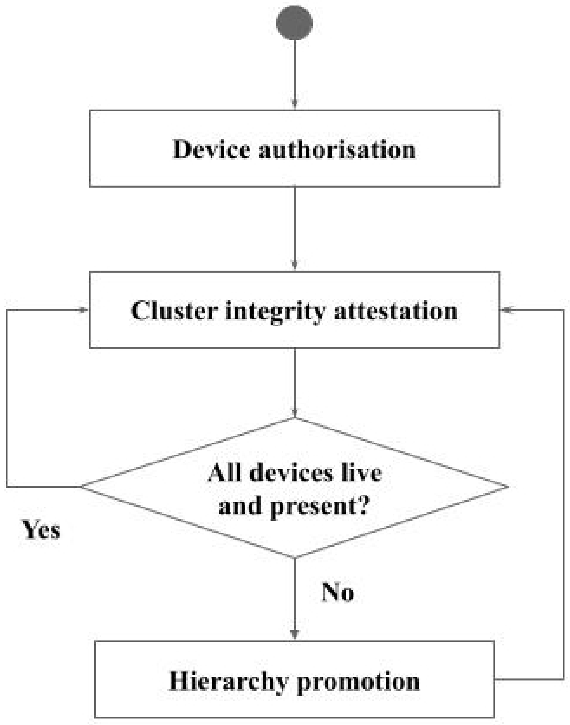

Figure 7.1: Authorisation Life Cycle.

Table 7.2: Function Symbols

Ref	Symbol	Meaning
1	N	Identifier of device N, annotates direction of message
2	K_n	Public key of Device n, providing decryption for only device n
3	K_n^{-1}	Private key of device n, providing signature proving origin from device n
4	Nn	Nonce from device n, showing freshness of value
5	SSID	Service Set ID, the cluster network name
6	K_{as}	Shared network key
6	Sn	Secret shadow of device n, their part of the whole rank hash
7	$\oplus$	XOR function, eXclusive OR to create secret shadows
8	RH	Rank Hash, the full combined rank secret from the collective shadows

7.3.7 Function Symbols

1. **Identifier** of the device which a message is destined for, towards, or containing identifier material for challenge-response of authenticity. This identifier could be a public key, a signature or MAC address. It is not really important in practice what this data is, but serves useful in the activity diagrams as to ascertain workflow.

2. **Public key** or the public key of the public and private key pair belonging to device n, is the key designed to be in the public domain. All devices have a public key, but some devices can be used to communicate with freely, and others have restrictions. This is to prevent ranked devices from reading, writing, or executing commands provided by untrusted guests on the network. This key is also used as an identifier, legible to machines more than humans.

3. **Private key** of the public and private key pair belonging to device n, is the key used for proving authenticity of messages sent from one device to another. Within the activity diagrams, this key denotes a Digital Signature Algorithm (DSA), an algorithm representing part of a device's private key that proves a message has come from the device it claims to have come from. This is an integrity-oriented piece of data to prevent imposters from communicating with authorised members.

4. **Nonce** or a oNly-once, value sent from one device to another as part of a challenge-response scenario. Now, when dealing with authentication or unique messages, a nonce represents freshness; a random piece of data transferred within a limited period of time that cannot be stored and replicated to falsely impersonate during time to come. The nonce can be used once within a challenge-response scenario to ascertain data freshness in addition to something less fresh, like a network shared key.

5. **SSID** is the network identifier that operates alongside the shared network key. It identifiers the cluster of aquaponics systems which new network members will need t join within the BLE range.

6. **Shared network key** is the secret symmetric key used for new members to gain initial access to the network before they can become Privates or promotion to General or FM.

7. **Sn** is the shadow of device n, or the part secret it holds as a member of a rank. The secret shadows together form a Rank Hash (RH), a bitstring representing the complete symmetric key to all the devices in a cluster of the same hierarchy position and cluster when each shadow is XOR'd against the Random String (RS).

8. $\oplus$ eXclusive OR, or XOR function, used to create ciphertext from the Boolean logic of 1 or 0 input to create a different output in the same manner as a truth table.

9. **RH** or Rank Hash, is the shared symmetric key between all devices of a rank, accessible only by a Field Marshal or upon cluster attestation when a single device temporarily

acts as a central authority by which to request and compare the shadows of all others in that cluster rank. In doing so the rank will either have a matching RH and be complete, or it will not match, and a new device will reset the rank's hash and shadows.

The three algorithms pertaining to the authorisation system are designed to provide granularity on top of the TLS secure channel - to prevent against inside attacks, DDoS, and to alert us of system downtime by an absent device or faulty cluster. The network topology is decentralised, and operates as a series of clustered devices allocated by BLE range. Some devices overlapping in range will be able to cross-communicate between clusters for backup across the network.

With reference to the three-tier hierarchy:

1. **Field Marshall (FM):** Communicates with FM from any cluster and Generals from his own.
2. **General:** Communicates with all parties from his cluster only.
3. **Private:** Communicates with any General belonging to his own cluster only.

7.3.8 Device authorisation

Device authorisation describes a process in which the newest device to join a cluster correlates to lowest rank, or the least trustworthy.

1. **Goals:**
   (a) **Entity authentication:** A believes B recently replied to a specific challenge.
   (b) **Secure key establishment:** A has a key which A believes is suitable for communication with B.
   (c) **Key confirmation:** A has received evidence confirming that B knows K.
   (d) **Message authentication:** A has received evidence that B has acted with timely freshness.
   (e) **Secrecy:** Freshness has been ascertained as secret.
2. **Principles:**
   (a) Device A acts as General.
   (b) Algorithms exemplify various ranks, where A stands for 'AgentA'.
   (c) The purpose of this exercise is not to replace TLS but to add granularity to it.
   (d) The shared key is a network key and not the exclusive property of A and B.
   (e) Since the shared key has multiple users, a N-once is employed for unique authentication.

3. **Assumptions:**

   (a) The network key is unique to each cluster of aquaponics systems.

   (b) Each device in a cluster represents a single aquaponics system.

   (c) Network key is shared with devices intending to join a cluster, as with any normal network.

   (d) The network SSID and key are used in conjunction with a nonce to authenticate.

   (e) The cluster key acts as a network key for the group of devices belonging to, or expected to belong to, a cluster of aquaponics systems. These systems are bounded by the range of a protocol suitable for the IoT application domain, probably BLE, with a range of up to 100m.

4. **Workflow and syntactics:**

   (a) Acting as General, A advertises for new cluster members periodically as per usual BLE GATT stack, whilst broadcasting their public key.

   ```
   A → B : Ka
   ```

   (b) B responds to the GATT with the SSID and network key, B's public key, signed by B, and encrypted by A's public key.

   ```
   B → A : {{SSID, Kas, Kb}Kb-1}Ka
   ```

   (c) The General, A, checks that the SSID and network key match the correct cluster key in their possession and responds to B with a N-once as an OTP, encrypted using B's public key, and signed by A. If this OTP is not entered within a strict time limit, B will be blacklisted by their public key for a period of time against DDoS.

   ```
   A → B : {{Na}Kb}Ka-1
   ```

   (d) B receives the OTP and enters it into the tokenisation function to join the network as a Private. This Private is now in a position to receive instructions from a General.

   ```
   B → A : {{Na}Ka}Kb-1
   ```

5. **AVISPA + SPAN results for the goals of secrecy of SSID, and shared key** $\mathbf{K}_a s$

   (a) **OFMC:** summarised as safe under a bounded number of sessions.
   (b) **CL-ATSE:** summarised as safe under bounded number of sessions and typed model.
   (c) **SATMC:** summary inconclusive.
   (d) **TA4SP:** summarised as safe under typed model, over approximation and unbounded number of sessions, no attack trace found.

6. **AVISPA + SPAN results for the goals of secrecy and freshness of N-once Na**

   (a) **OFMC:** summarised as safe under a bounded number of sessions.
   (b) **CL-ATSE:** summarised as safe under bounded number of sessions and typed model.
   (c) **SATMC:** summary inconclusive.
   (d) **TA4SP:** summarised as safe under typed model, over approximation and unbounded number of sessions, no attack trace found.

7. **ProVerif verification summary results for two scenarios:**

   (a) **Query not attacker SSID is true, Query not attacker Kas is true:** In the first scenario, this confirms that the attacker cannot gain knowledge of either SSID nor the shared symmetric key.
   (b) **Query not attacker Na is true:** In the second scenario, this confirms that the attacker cannot gain knowledge of the N-once Na bitstring.

7.3.9 Cluster integrity attestation

Cluster integrity attestation describes a process in which a General periodically checks a cluster for all present and functioning devices. This is to prevent downtime of each aquaponics system, represented by the completeness of all clustered devices. This is performed every time a device joins the cluster, as the cluster will require a reset of known devices to include the newest Private.

1. **Goals:**

   (a) **Entity authentication:** A believes B .. N recently replied to a specific challenge.
   (b) **Key confirmation:** A has received evidence confirming that B .. N knows Shadows Sb .. Sn.
   (c) **Message authentication:** A has received evidence that B .. N has acted with timely freshness.

   (d) **Secrecy:** Freshness has been ascertained as secret, for each new Shadow and full Rank Hash.

2. **Principles:**
   (a) Device A is acting as General, and is an Authorised device.
   (b) Devices B .. N stands for any device already authorised as FM, Generals, and Privates.
   (c) Devices B .. N will all have Shadows apart from the device which has just become Authorised.
   (d) Devices B .. N will all have Shadows if this attestation is a routine check only.
   (e) The purpose of this exercise is to detect downtime or incompleteness.
   (f) The Shadows add together to act as a Rank Hash demonstrating the known devices of a cluster.

3. **Assumptions:**
   (a) The Rank Hash is unique to each cluster.
   (b) Each device in a cluster represents a single aquaponics system.
   (c) The General has an existing RH of all Shadows to verify devices against previous attestation.

4. **Workflow and syntactics:**
   (a) Acting as General, A requests the Shadows of each device by broadcasting his public key, SSID and network key, encrypted and signed, to all known devices belonging to that cluster. Since the messages are broadcasted to each device in the cluster, each message is encrypted by the public key of its respective recipient. The number of messages broadcasted will therefore be the same as the number of authorised devices, as the General will hold a 'keyring' of all cluster members.

```
A → B .. N : {{SSID, Ka, Kas}Kb .. n}Ka-1
```

   (b) Devices B .. N respond to the challenge with their Shadows and public keys, encrypted by the General's public key and signed by their individual signature.

```
B .. N → A : {{Shb .. n, Kb .. n}Ka}Kb . . n-1
```

   (c) The General checks all Shadows and public keys against the ones on his keyring, and verifies that the sum of the Shadows matches the Rank Hash he holds by XOr'ing the Shadows.

```
A : SHa XOR SHb .. n = RH
```

(d) If this attestation is a routine check and all devices are correct and functioning, the General stores the new Shadows on his keyring and the full Rank Hash. However, if this attestation follows on from a new device joining the cluster or there's a missing device, we proceed to hierarchy promotion.

5. **Results for the goals of secrecy of shadows Sb...Sn**

   (a) **OFMC:** summarised as safe under a bounded number of sessions.

   (b) **CL-ATSE:** summarised as safe under bounded number of sessions and typed model.

   (c) **SATMC:** summarised as safe under strongly typed model, bounded number of sessions, and bounded message depth. No attack traces were found.

   (d) **TA4SP:** summarised as safe under typed model, over approximation and unbounded number of sessions, no attack trace found.

6. **ProVerif verification summary results for three scenarios:**

   (a) **Query not attacker SSID is true, Query not attacker Kas is true:** This confirms that the attacker cannot gain knowledge of either SSID nor the shared symmetric key.

   (b) **Query not attacker Shb is true:** In the second scenario, this confirms that the attacker cannot gain knowledge of the shadow secret Shb.

   (c) **Query not attacker Sha and Shb is true:** In the third scenario, this confirms that the attacker cannot gain knowledge of example multi-party shadow secrets Sha nor Shb.

7.3.10 Hierarchy promotion

Hierarchy promotion describes a process in which the cluster recovers from a missing or dysfunctional device by restoring the cluster to its intended hierarchical state, or to a new state to include a recently authorised device. Where the hierarchy must include at least one FM and two Generals, if either of these devices is found to be missing during cluster attestation, the oldest Private in the group will be promoted to the next position, and the Shadows and Rank Hash will be recalculated. If the cluster has a new member, the hierarchy remains but Shadows and Rank Hash are recalculated to reflect the additional device and added to the General's keyring.

1. **Goals:**

   (a) **Entity authentication:** A believes B .. N recently replied to a specific challenge.

   (b) **Key confirmation:** A has received evidence confirming that B .. N knows Shadows Sb .. Sn.
   (c) **Message authentication:** A has received evidence that B .. N has acted with timely freshness.
   (d) **Secrecy:** Freshness has been ascertained as secret, for each new Shadow and full Rank Hash.

2. **Principles:**
   (a) Device A is acting as General, and is an Authorised device.
   (b) Devices B .. N stands for any device already authorised as FM, Generals, and Privates.
   (c) Devices B .. N will all have Shadows apart from the device which has just become Authorised.
   (d) Devices B .. N will all have Shadows if this attestation is a routine check only.
   (e) The purpose of this exercise is to detect downtime or incompleteness.
   (f) The Shadows add together to act as a Rank Hash demonstrating the known devices of a cluster.

3. **Assumptions:**
   (a) The Rank Hash is unique to each cluster.
   (b) Each device in a cluster represents a single aquaponics system.
   (c) The General has an existing RH of all Shadows to verify devices against previous attestation.
   (d) New devices that have recently been authorised as part of the cluster will require a Shadow.
   (e) Adding a new device or resetting the hierarchy will trigger new Shadows and Rank Hash.

4. **Workflow and syntactics:**
   (a) The first, second and third part of this process is the same as cluster attestation and do not need to be included. To reiterate:

      Acting as General, A requests the Shadows of each device by broadcasting his public key, SSID and network key, encrypted and signed, to all known devices belonging to that cluster. Since the messages are broadcast to each device in the cluster, each message is encrypted by the public key of its respective recipient. The number of messages broadcast will therefore be the same as the number of authorised devices, as the General will hold a ‘keyring’ of all cluster members.

```
A → B .. N : {{SSID, Ka, Kas}}Kb .. n}}Ka-1
```

(b) Authorised devices B .. N respond to the challenge with their Shadows and public keys, encrypted by the General's public key and signed by their individual signature.

```
B .. N → A : {{Shb .. n, Kb .. n}}Ka}Kb .. n-1
```

(c) The General checks all Shadows and public keys against the ones on his keyring, and verifies that the sum of the Shadows matches the Rank Hash he holds by XOr'ing the Shadows.

```
A : SHa XOR SHb .. n = RH
```

(d) Assuming there is a difference between the Rank Hash and the sum of the Shadows he has received, the General calculates new Shadows based on the number of devices in the cluster according to the number of responses he has just had, plus his own.

This is done by generating random bit strings for each device plus his own, XOR'd against an additional bit string, noted below as a n-once.

```
A : Na XOR Nb .. n XOR Nx = RH
```

(e) The General now sends a copy of the Rank Hash to the Field Marshall, who holds it for backup and this is the reason why he does not communicate with any other participants other than FMs from other clusters. The FM will also hold Rank Hashes from other clusters for additional backup. This message includes the General's public key to demonstrate authenticity, is encrypted with the FMs public key, and signed by the General.

```
A → FM : {{RH, Ka}Kfm}Ka-1
```

(f) Lastly, The General broadcasts Shadows to each respective device. For this, the Shadows will be sent as individual messages, including the General's public key, and encrypted by the public key of each individual device, signed by the General.

```
A → B .. N : {Ka, SHb .. n, Kb .. n}Ka-1
```

5. **Results for the goals of secrecy of RH**

   (a) **OFMC:** summarised as safe under a bounded number of sessions.

   (b) **CL-ATSE:** summarised as safe under bounded number of sessions and typed model.

   (c) **SATMC:** summarised as safe under strongly typed model, bounded number of sessions, bounded message depth.

   (d) **TA4SP:** summarised as safe under typed model, over approximation and unbounded number of sessions, no attack trace found.

6. **Results for the goals of secrecy of Sb...Sn**

   (a) **OFMC:** summarised as safe under a bounded number of sessions.

   (b) **CL-ATSE:** summarised as safe under bounded number of sessions and typed model.

   (c) **SATMC:** summarised as inconclusive.

   (d) **TA4SP:** summarised as safe under typed model, over approximation and unbounded number of sessions, no attack trace found.

7. **ProVerif verification summary results for the last three scenarios, where items a-c are reiterations of cluster integrity attestation:**

   (a) **Query not attacker N is true, Query not attacker Nb is true, Query not attacker Nx is true:** In the first scenario, this confirms that the attacker cannot gain knowledge of any example XOR function as would be broadcast as shadow secrets.

   (b) **Query not attacker RH is true:** In the second scenario, this confirms that the attacker cannot gain knowledge of the full Rank Hash or sum of shadows secret parts, RH.

   (c) **Query not attacker Shb is true, Query not attacker Shc is true:** In the third scenario, this confirms that the attacker cannot gain knowledge of example multi-party shadow secrets Shb nor Shc, representing numerous broadcasting to Shn.

7.4 Contributions

Through formal verification, the authorisation model has been proven as safe, through a majority of three out of four model checking methods, and in some instances, four out of four. The sat-prover was sometimes inconclusive, probably due to the efficiencies of the messaging between nodes - but these efficiencies were positive towards a lightweight effort.

This section concludes the benefits of the authorisation model:

7.4.1 Low Consumptions

By reducing the number of messages between nodes to as few as possible, the freshness of the nonces and secrecy of the priority data is assured. This has been proven using the four backend model checkers in the AVISPA -SPAN framework. In addition, the less processing that the ESP32 has to do for ascertaining authenticity for each new or refreshed network member, the less energy the network will use, and the longer the battery life of devices will last.

7.4.2 Self-Healing

The design includes a periodic cluster integrity attestation function to assure the network that the hierarchy devices are still live and responsive. The primary function of the hierarchy is to separate levels of data access away from access to the public, or authorised devices allowed to participate on the network that may not be as trustworthy as they should. In comparison to TLS which is either permitted access or not, this tiered system separates rights into the four levels between public, unranked, high-ranked, and secret backup. In addition to preventing malicious attempts, this system is designed to form automated business continuity in the event of network loss.

7.4.3 Limited Chain

The restricted nature of the privileged devices within each cluster means that only the correlating set of identifiers, secret shadows and keys are maintained in a type of chain. The repetition of this data is limited to only a few devices, and although there will be an overlap of the higher rank, Field Marshal, between clusters due to the longer range, it is predictable, finite, and ephemeral by nature - when a previous device is lost, the new data set will replace their part in the chain. An ephemeral application enables the robustness of data replication and dispersion in the same enabling way as the famous blockchain, without the unmanageable scalability issues of storage.

7.4.4 Agnostic Application

The security properties expected of any desktop, mobile or hardware application require the same as TLS - a public and private key pair, hash functions, Boolean logic (XOR), exponentiation operators, hash, and symmetric cryptography. These functions are native to the ESP32 hardware acceleration, and since security is becoming an increasingly important facet to any application, a multi-purpose approach to functions between applications lessens the burden of processing and storage. Cryptographic functions have therefore been used for identification and other attributes to reduce data volume.

7.4.5 Consensus

Consent is becoming more important for IoT applications because of privacy, and security can enable the principles of the GDPR by supporting privacy through consent. Although the aquaponics smart-farm does not utilise the private and sensitive information of living subjects, it exemplifies the utility of a tiered authorisation system. The separation of privilege levels, pseudonymisation of device identifiers, and anonymity of 'rankings' demonstrates the overlap between security and privacy using majority consensus and protection of authorisation.

7.4.6 Further Work

This part of the research demonstrated an automated application of self-healing and ephemeral data sets, replicated in a limited way between closely associated devices. Applications of such a managed authorisation system could extend into domains such as healthcare, domestic and industrial. Such an energy and storage-conscious design could greatly benefit challenges faced in energy production, environmental conservation and healthcare, in which monitoring over long periods of time could take advantage of smaller data chains and device restoration.

Chapter 8

Conclusion and Further Work

We now conclude with a summary of contributions to confirm the hypothesis:

An energy-efficient IoT security solution can be created by combining a reduced variant of TLS with low-overhead consensus-based access control.

The answer to this hypothesis is yes - it is possible to create an energy-efficient variant of TLS by careful deconstruction and re-combination of its various elements, and consensus-based authorisation is a viable alternative to heavier options.

8.1 Research Contributions

Our research comprised three empirical investigations that each affirmed elements of the hypothesis, the divisions between which were dictated by the particular testing methodology appropriate to each case. The three empirical chapters above correlate to three contributions of novelty for the benefit of the IoT domain in general and the smart-farm domain in particular. We began by modelling a risk framework, followed by implementation and power analysis of the results of re-configuring TLS within that framework, measured against existing TLS configurations. Finally, we proposed a granular authorisation system to fulfil the access control niche that TLS provided for us.

8.1.1 A Risk-Driven Energy Assessment

Research Question 1: What are the security and privacy requirements of the aquaponics smart-farm?

We performed a DPIA (Data Protection Impact Assessment), which invoked the following properties, together comprising a novel metrics framework for IoT-oriented energy-conscious systems:

- **Security application:** an analysis of the range of IoT protocols and targeting of security provision using this data allowed for a reduction of energy requirements in both key and data exchange.
- **Energy consumption:** scoring risk severity by session length and data exchange requirements provided a platform to reduce unnecessary overheads.
- **Privacy substitution:** many of the confidentiality functions previously in use were viably substituted for privacy functions, allowing protection without the overheads of encryption.

8.1.2 Energy-Conscious Authentication and Data Exchange

Research Questions 2 and 3: How far can energy consumption be reduced during device authentication and subsequent data exchange?

The handshake and secure session elements of TLS were reconfigured in a manner more closely modelled on the specific characteristics of the IoT, in order to achieve a method of secure authentication and payload-protection that reduces overheads by around 96%:

- **Privacy:** Following the proposal of the DPIA removed the need for unnecessary confidentiality.
- **Certificates:** Removing reliance on X.509 and its infrastructure, the model became more decentralised and robust.
- **Session timeout:** Promoting long-term relationships between devices enabled fewer handshakes.
- **Payload protection:** Protection at source removed burden and vulnerability from the central hub.
- **Energy reduction:** By ascertaining the lightest functions, suitable keys and ciphersuites applied.
- **Edge processing:** Using hardware acceleration on edge devices boosted productivity.
- **IoT-appropriate ciphersuites:** Selective configuration of CIA to target low cost primitives.
- **Additional keys:** Separating out confidentiality by producing an extra key allowed flexibility.

8.1.3 Energy-Conscious Consensus Mechanism

Research Question 4: Can a lightweight, self-healing consensus mechanism be shown to be safe?

The binary approach to TLS can be improved by granularity, autonomy and group consensus in clusters of self-healing systems on a decentralised network. The authorisation system, which we subjected to formal verification, contributes a model in which smart farms could connect without human monitoring:

- **Low consumption:** Existing cryptographic functions were recycled to reduce processing.
- **Self-healing:** Periodic cluster attestation enabled fault detection and automatic fix.
- **Limited chain:** Cluster hierarchies allowed limited ranked devices to ensure consensus was not vulnerable to typical attacks by majority, unlimited vote of new additions.
- **Agnostic application:** Applications using similar security functions can adopt the same processes.
- **Consensus:** Autonomous but limited group consensus removes the vulnerability of human interference.

8.2 Research Limitations and Further Work

Despite the positive outcome of the research, there were clear limitations. Predominantly, the empirical evidence and projected benefits of the investigations were based on a prototype, in environments which represent production as much as practically possible - but still this is not actually a production implementation. In the 'wild', other caveats may apply which will not obvious in an academic context.

At the time of writing, the study presents a series of key lengths and other CIA computations considered safe under the NIST and FIPS guidelines. These may eventually become redundant, as quantum computers and developments in TLS 1.3+ become the new de facto standard. This is a reflection on key strength and algorithms underpinning the key exchange processes and data exchange. Of course, the study is inherently limited to the technology of its era, particularly where libraries and hardware platforms such as mbedtls provide limited options. Scalability of an IoT domain is also difficult to fathom when operating on theoretic projections - but tools to simulate the realities do exist.

Lastly, our work has been specific to a particular microcontroller architecture, and should be repeated for each different target architecture. In addition, the energy-conscious focus of this study could be expanded to include alternative energy generation by location and natural power sources - apparently, the sun, wind and rain as 'green' technologies have a lot to offer.

Bibliography

[1] C. higher education sustainability conference, “California higher education sustainability conference,” https://chesc.org/, 2018, accessed: 2022-6-18.

[2] S. S. Jersey, “Sustainable south jersey,” https://sustainablesouthjersey.org/, Oct. 2018, accessed: 2022-6-18.

[3] Raspberry Pi Ltd, “Raspberry pi,” https://www.raspberrypi.com/, 2022, accessed: 2022-6-18.

[4] R. Mischianti, “Renzo mischianti,” https://www.mischianti.org/, 2021, accessed: 2022-6-18.

[5] Espressif, “ESP32,” https://commons.wikimedia.org/wiki/File:Espressif_ESP32_Chip_Function_Block_Diagram.svg, 2021.

[6] WiPro, “TLS 1.2 and 1.3 Handshakes,” https://www.wipro.com/blogs/suresha-ejari/five-ways-tls-1-3-will-take-your-privacy-and-performance-readiness-to-the-next-level/, 2022.

[7] R. L. Nelson and J. S. Pade, *Aquaponic food production.* Nelson and Pade, Inc., 2008.

[8] S. Bernstein, *Aquaponic Gardening: A Step-By-Step Guide to Raising Vegetables and Fish Together.* New Society Publishers, Oct. 2011.

[9] J. L. Edmondson, H. Cunningham, D. O. Densley Tingley, M. C. Dobson, D. R. Grafius, J. R. Leake, N. McHugh, J. Nickles, G. K. Phoenix, A. J. Ryan, V. Stovin, N. Taylor Buck, P. H. Warren, and D. D. Cameron, “The hidden potential of urban horticulture,” *Nature Food*, vol. 1, no. 3, pp. 155–159, Mar. 2020.

[10] H. Cunningham, “Localising food production with aquaponics,” Department of Computer Science, University of Sheffield, Tech. Rep., 2015.

[11] H. Cunningham and B. Kotzen, “Meet the sustainable vegetables that thrive on a diet of fish poo,” *The Conversation*, Nov. 2015.

[12] S. Rudd and H. Cunningham, “Low-Energy Authentication with Selective Privacy for Heterogeneous IoT Devices in Smart-Farms,” in *2021 30th Conference of Open Innovations Association FRUCT*, Oct. 2021, pp. 230–238.

[13] ——, “Selective Privacy in IoT Smart-Farms for Battery-Powered Device Longevity,” *Proceedings from the Conference of Applied Computing 2021*, Aug. 2021.

[14] ——, “Towards lightweight authorisation of iot-oriented smart-farms using a self-healing consensus mechanism,” *Proceedings of the 31st Conference of Open Innovations Association FRUCT*, pp. 265–276, 2022.

[15] ——, “Threat modelling with the gdpr towards a security and privacy metrics framework for iot smart-farm application,” *7th International Conference on Internet of Things, Big Data and Security*, pp. 91–102, 2022.

[16] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A literature review journal of computer and communications,” *International Research Journal of Engineering and Technology (IRJET)*, 2015.

[17] Z. Yan, P. Zhang, and A. V. Vasilakos, “A survey on trust management for internet of things,” *Journal of Network and Computer Applications*, vol. 42, pp. 120–134, Jun. 2014.

[18] R. Nicolescu, M. Huth, P. Radanliev, and D. De Roure, “Mapping the values of IoT,” *J. Inf. Technol. Impact*, vol. 33, no. 4, pp. 345–360, Dec. 2018.

[19] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, “Information centric networking in the IoT,” 2014.

[20] J. H. Nord, A. Koohang, and J. Paliszkiewicz, “The internet of things: Review and theoretical framework,” *Expert Syst. Appl.*, vol. 133, pp. 97–108, Nov. 2019.

[21] I. Lee and K. Lee, “The internet of things (IoT): Applications, investments, and challenges for enterprises,” *Bus. Horiz.*, vol. 58, no. 4, pp. 431–440, Jul. 2015.

[22] C. Maple, “Security and privacy in the internet of things,” *Journal of Cyber Policy*, vol. 2, no. 2, pp. 155–184, May 2017.

[23] P. Kirci, E. Ozturk, and Y. Celik, “Smart greenhouse and smart agriculture,” in *Conference of Open Innovations Association, FRUCT*, 2021, pp. 455–459.

[24] T. Searchinger, C. Hanson, J. Ranganathan, B. Lipinski, R. Waite, R. Winterbottom, A. Dinshaw, R. Heimlich, M. Boval, P. Chemineau, and Others, “Creating a sustainable food future. a menu of solutions to sustainably feed more than 9 billion people by 2050. world resources report 2013-14: interim findings,” Ph.D. dissertation, World Resources Institute (WRI); World Bank Groupe-Banque Mondiale; United . . . , 2014.

[25] O. Mora, C. Le Mouël, M. de Lattre-Gasquet, C. Donnars, P. Dumas, O. Réchauchère, T. Brunelle, S. Manceron, E. Marajo-Petitzon, C. Moreau, M. Barzman, A. Forslund, and P. Marty, “Exploring the future of land use and food security: A new set of global scenarios,” *PLoS One*, vol. 15, no. 7, p. e0235597, Jul. 2020.

[26] X. Lin, X. Sun, G. Manogaran, and B. S. Rawal, “Advanced energy consumption system for smart farm based on reactive energy utilization technologies,” *Environ. Impact Assess. Rev.*, vol. 86, p. 106496, Jan. 2021.

[27] T. Khaoula, R. A. Abdelouahid, I. Ezzahoui, and A. Marzak, “Architecture design of monitoring and controlling of IoT-based aquaponics system powered by solar energy,” *Procedia Comput. Sci.*, vol. 191, pp. 493–498, Jan. 2021.

[28] M. Lu, G. Fu, N. B. Osman, and U. Konbr, “Green energy harvesting strategies on edge-based urban computing in sustainable internet of things,” *Sustainable Cities and Society*, vol. 75, p. 103349, Dec. 2021.

[29] R. A. Yalçın and H. Ertürk, “Improving crop production in solar illuminated vertical farms using fluorescence coatings,” *Biosystems Eng.*, vol. 193, pp. 25–36, May 2020.

[30] Y. Shao, J. Li, Z. Zhou, Z. Hu, F. Zhang, Y. Cui, and H. Chen, “The effects of vertical farming on indoor carbon dioxide concentration and fresh air energy consumption in office buildings,” *Build. Environ.*, vol. 195, p. 107766, May 2021.

[31] M. Liebman-Pelaez, J. Kongoletos, L. K. Norford, and C. Reinhart, “Validation of a building energy model of a hydroponic container farm and its application in urban design,” *Energy Build.*, vol. 250, p. 111192, Nov. 2021.

[32] Y.-J. Son, J.-E. Park, J. Kim, G. Yoo, T.-S. Lee, and C. W. Nho, “Production of low potassium kale with increased glucosinolate content from vertical farming as a novel dietary option for renal dysfunction patients,” *Food Chem.*, vol. 339, p. 128092, Mar. 2021.

[33] J. Aschemann-Witzel, R. F. Gantriis, P. Fraga, and F. J. A. Perez-Cueto, “Plant-based food and protein trend from a business perspective: markets, consumers, and the challenges and opportunities in the future,” *Crit. Rev. Food Sci. Nutr.*, vol. 61, no. 18, pp. 3119–3128, 2021.

[34] A. P. Kale and S. P. Sonavane, “IoT based smart farming : Feature subset selection for optimized high-dimensional data using improved GA based approach for ELM,” pp. 225–232, 2019.

[35] M. Taneja, N. Jalodia, J. Byabazaire, A. Davy, and C. Olariu, “SmartHerd management: A microservices-based fog computing-assisted IoT platform towards data-driven smart dairy farming,” *Softw. Pract. Exp.*, vol. 49, no. 7, pp. 1055–1078, Jul. 2019.

[36] M. A. Zamora-Izquierdo, J. Santa, J. A. Martínez, V. Martínez, and A. F. Skarmeta, “Smart farming IoT platform based on edge and cloud computing,” *Biosystems Eng.*, vol. 177, pp. 4–17, Jan. 2019.

[37] A. Shrivastava, C. K. Nayak, R. Dilip, S. R. Samal, S. Rout, and S. M. Ashfaque, "Automatic robotic system design and development for vertical hydroponic farming using IoT and big data analysis," *Materials Today: Proceedings*, Jul. 2021.

[38] S.-C. Wang, W.-L. Lin, C.-H. Hsieh, and Others, "To improve the production of agricultural using IoT-based aquaponics system," *International Journal of Applied Science and Engineering*, vol. 17, no. 2, pp. 207–222, 2020.

[39] K. Taji, R. Ait Abdelouahid, I. Ezzahoui, and A. Marzak, "Review on architectures of aquaponic systems based on the internet of things and artificial intelligence: Comparative study," in *Proceedings of the 4th International Conference on Networking, Information Systems & Security*, ser. NISS2021, no. Article 50. New York, NY, USA: Association for Computing Machinery, Apr. 2021, pp. 1–9.

[40] S. C. Lauguico, R. S. Concepcion, J. D. Alejandrino, R. R. Tobias, D. D. Macasaet, and E. P. Dadios, "A comparative analysis of machine learning algorithms modeled from machine vision-based lettuce growth stage classification in smart aquaponics," *Int. J. Environ. Sci. Dev.*, vol. 11, no. 9, pp. 442–449, 2020.

[41] A. Reyes-Yanes, S. Gelio, P. Martinez, and R. Ahmad, "Wireless sensing module for IoT aquaponics database construction," *International Journal of Electronics and Electrical Engineering*, vol. 9, no. 2, pp. 43–47, 2021.

[42] A. R. Yanes, P. Martinez, and R. Ahmad, "Towards automated aquaponics: A review on monitoring, IoT, and smart systems," *J. Clean. Prod.*, vol. 263, p. 121571, Aug. 2020.

[43] I. Ezzahoui, R. A. Abdelouahid, K. Taji, and A. Marzak, "Hydroponic and aquaponic farming: Comparative study based on internet of things IoT technologies," *Procedia Comput. Sci.*, vol. 191, pp. 499–504, Jan. 2021.

[44] L. Ada, "Adafruit HUZZAH32 - ESP32 feather," https://learn.adafruit.com/adafruit-huzzah32-esp32-feather/using-with-arduino-ide, May 2017, accessed: 2021-12-16.

[45] L. S. Ezema, E. C. Ifediora, A. A. Olukunle, and N. C. Onuekwusi, "Design and implementation of an ESP32-Based smart embedded industrial poultry farm," *European Journal of Engineering and Technology Research*, vol. 6, no. 3, pp. 103–108, Apr. 2021.

[46] M. Reddy, M. K. Saiteja, J, Gurupriyanka, N, Sridhar, and N. K. G N, "IOT based crop monitoring system for smart farming," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, Jul. 2021, pp. 562–568.

[47] R. K. Kodali, M. S. Kuthada, and Y. K. Yogi Borra, "LoRa based smart irrigation system," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, Dec. 2018, pp. 1–5.

[48] Y. Choudhary, B. Umamaheswari, and V. Kumawat, "A study of threats, vulnerabilities and countermeasures: An IoT perspective," *Humanit. Rep.*, vol. 8, no. 4, pp. 39–45, 2021.

[49] F. Q. Kareem, S. Y. Ameen, A. A. Salih, D. M. Ahmed, S. F. Kak, H. M. Yasin, I. M. Ibrahim, A. M. Ahmed, Z. N. Rashid, and N. Omar, "SQL injection attacks prevention system technology," *Asian Journal of Research in Computer Science*, vol. 13, p. 32, 2021.

[50] C. S. Cheah and V. Selvarajah, "A review of common web application breaching techniques (SQLi, XSS, CSRF)," in *3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021)*, 2021, pp. 540–547.

[51] Espressif, "SPIFFS filesystem," 2021.

[52] Mongoose, "Mongoose OS," 2021.

[53] OWASP, "OWASP top ten," 2021.

[54] S. Rizvi, R. Pipetti, N. McIntyre, J. Todd, and I. Williams, "Threat model for securing internet of things (IoT) network at device-level," p. 100240, 2020.

[55] A. Altaf, H. Abbas, F. Iqbal, M. M. Z. M. Khan, A. Rauf, and T. Kanwal, "Mitigating service-oriented attacks using context-based trust for smart cities in IoT networks," *Int. J. High Perform. Syst. Archit.*, vol. 115, p. 102028, May 2021.

[56] H. S. Shreenidhi, S. Prabakar, and P. A. Kumar, "Intrution detection system using IoT device for safety and security," in *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, Mar. 2021, pp. 340–344.

[57] X. Jiang, M. Lora, and S. Chattopadhyay, "An experimental analysis of security vulnerabilities in industrial IoT devices," pp. 1–24, 2020.

[58] N. N. Thilakarathne, "Security and privacy issues in IoT environment," 2020, accessed:2021-12-01.

[59] P. A. Boampong and L. A. Wahsheh, "Different facets of security in the cloud," in *Proceedings of the 15th Communications and Networking Simulation Symposium*, ser. CNS '12, no. Article 5. San Diego, CA, USA: Society for Computer Simulation International, Mar. 2012, pp. 1–7.

[60] L. Zhang, X. Deng, and Y. Wang, "Shellshock bash vulnerability modeling analysis based on petri net," in *2021 International Conference on Networking and Network Applications (NaNA)*, Oct. 2021, pp. 242–247.

[61] NHS Digital, "BASHLITE botnet," 2021.

[62] GDPR, “General Data Protection Regulation,” https://gdpr-info.eu/, 2023.

[63] PCI-DSS, “Official PCI Security Standards Council Site,” https://www.pcisecuritystandards.org/, 2023.

[64] H. A. Abdulghani, N. A. Nijdam, A. Collen, and D. Konstantas, “A study on security and privacy guidelines, countermeasures, threats: IoT data at rest perspective,” *Symmetry*, vol. 11, no. 6, p. 774, Jun. 2019.

[65] L. M. Engineers, “ESP32 over the air (OTA) web updater in arduino IDE - 3 simple steps,” https://lastminuteengineers.com/esp32-ota-web-updater-arduino-ide/, Dec. 2018, accessed: 2021-12-27.

[66] D. Ramesh Reddy, K. Bikash, K. P. Vivek, U. Z. Sadek, and P. Kodali, “Design and development of an IoT-Based smart medication device,” in *Soft Computing for Problem Solving.* Springer Singapore, 2021, pp. 77–88.

[67] Espressif, “ESP32 fault injection vulnerability - impact analysis,” 2020.

[68] CVE, “Cve-2019-17391,” 2019.

[69] D. Oswald, “Side-Channel attacks on SHA-1-Based product authentication ICs,” in *Smart Card Research and Advanced Applications.* Springer International Publishing, 2016, pp. 3–14.

[70] L. Mauri and E. Damiani, “STRIDE-AI: An approach to identifying vulnerabilities of machine learning assets,” in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, Jul. 2021, pp. 147–154.

[71] Y.-J. Shin, E. Cho, and D.-H. Bae, “PASTA: An efficient proactive adaptation approach based on statistical model checking for Self-Adaptive systems,” *Fundamental Approaches to Software Engineering*, vol. 12649, p. 292, 2021.

[72] A. Robles-González, J. Parra-Arnau, and J. Forné, “A LINDDUN-Based framework for privacy threat analysis on identification and authentication processes,” *Comput. Secur.*, vol. 94, p. 101755, Jul. 2020.

[73] M. Walkowski, J. Oko, and S. Sujecki, “Vulnerability management models using a common vulnerability scoring system,” *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, vol. 11, no. 18, p. 8735, Sep. 2021.

[74] B. Mantha, B. García de Soto, and R. Karri, “Cyber security threat modeling in the AEC industry: An example for the commissioning of the built environment,” *Sustainable Cities and Society*, vol. 66, p. 102682, Mar. 2021.

[75] P. Saitta, B. Larcom, and M. Eddington, “Trike v. 1 methodology document [draft],” *URL: http://dymaxion. org/trike/Trike v1 Methodology Documentdraft. pdf*, 2005.

[76] P. S. Checkland and B. Scholes, “J. 1990. soft systems methodology in action,” *UK: John Wiley and Sons*, 1990.

[77] P. J. Ryan and R. B. Watson, “Research challenges for the internet of things: What role can OR play?” *Systems*, vol. 5, no. 1, p. 24, Mar. 2017.

[78] D. Petkov, O. Petkova, T. Andrew, and T. Nepal, “Mixing multiple criteria decision making with soft systems thinking techniques for decision support in complex situations,” *Decis. Support Syst.*, vol. 43, no. 4, pp. 1615–1629, Aug. 2007.

[79] D. T. Kitaw and C. Beaumont, “Analysing information security in a bank using soft systems methodology,” *Information & Computer Security*, vol. 25, no. 3, pp. 240–258, Jan. 2017.

[80] D. Torres, “Cyber security and cyber defense for venezuela: an approach from the soft systems methodology,” *Complex & Intelligent Systems*, vol. 4, no. 3, pp. 213–226, Oct. 2018.

[81] R. Sharma, C. Zhang, S. C. Wingreen, N. Kshetri, and A. Zahid, “Design of blockchain-based precision Health-Care using soft systems methodology,” *Industrial Management & Data Systems*, vol. 120, no. 3, pp. 608–632, Jan. 2020.

[82] B. AlSabbagh and S. Kowalski, “Security from a systems thinking perspective - applying soft systems methodology to the analysis of an information security incident,” *ISSS-2013*, 2014.

[83] D. Parra, O. . Ruiz-Vanoye, J. A. . Barrera-Cámara, R. A. . Fuentes-Penna, A. . Sandoval, O. Díaz-Parra, J. A. Ruiz-Vanoye, R. A. Barrera-Cámara, A. Fuentes-Penna, and N. Sandoval, “Soft systems methodology for the strategic planning of the enterprise computer security,” https://www.redalyc.org/pdf/2652/265232976002.pdf, 2014, accessed: 2021-12-29.

[84] M. Mshangi, E. N. Nfuka, and C. Sanga, “Using soft systems methodology and activity theory to exploit security of web applications against heartbleed vulnerability,” *International Journal of Computing and ICT Research*, 2015.

[85] A. Varanda, L. Santos, R. L. d. C. Costa, A. Oliveira, and C. Rabadão, “Log pseudonymization: Privacy maintenance in practice,” *Journal of Information Security and Applications*, vol. 63, p. 103021, Dec. 2021.

[86] H. El Alloussi, L. Fetjah, and A. Chaichaa, “Securing the payment card data on cloud environment: Issues & perspectives,” *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 14, no. 11, p. 14, 2014.

[87] K. Razikin and A. Widodo, “General cybersecurity maturity assessment model: Best practice to achieve payment card Industry-Data security standard (PCI-DSS)

compliance," *CommIT (Communication and Information Technology) Journal*, vol. 15, no. 2, pp. 91–104, Aug. 2021.

[88] F. Yang and S. Manoharan, "A security analysis of the oauth protocol," in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 2013, pp. 271–276.

[89] OWASP, "OWASP risk rating methodology," 2014.

[90] H. Blaine, "The threat landscape of PKI: System and cryptographic security of x. 509, algorithms, and their implementations," *NATO Sci. Ser. Ser. A. Life Sci.*, p. 286, 2013.

[91] D. Li, W. Peng, W. Deng, and F. Gai, "A Blockchain-Based authentication and security mechanism for IoT," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, Jul. 2018, pp. 1–6.

[92] J. Larisch, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "CRLite: A scalable system for pushing all TLS revocations to all browsers," in *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017, pp. 539–556.

[93] T. Meskanen, V. Niemi, and J. Kuusijäarvi, "Privacy-Preserving peer discovery for group management in p2p networks," in *2020 27th Conference of Open Innovations Association (FRUCT)*, Sep. 2020, pp. 150–156.

[94] H. Abelson, R. Anderson, S. M. Bellovin, J. Benaloh, M. Blaze, W. Diffie, J. Gilmore, M. Green, S. Landau, P. G. Neumann, R. L. Rivest, J. I. Schiller, B. Schneier, M. A. Specter, and D. J. Weitzner, "Keys under doormats: mandating insecurity by requiring government access to all data and communications," *J Cyber Secur*, vol. 1, no. 1, pp. 69–79, Nov. 2015.

[95] D. Shaw, "Hundreds arrested as crime chat network cracked," *BBC*, Jul. 2020.

[96] Dierks & Rescorla, "RFC 5246," https://datatracker.ietf.org/doc/html/rfc5246, 2008, accessed: 2022-1-1.

[97] S. K. Tayyaba, M. A. Shah, N. S. A. Khan, Y. Asim, W. Naeem, and M. Kamran, "Software-defined networks (SDNs) and internet of things (IoTs): A qualitative prediction for 2020," *Network*, vol. 7, no. 11, 2016.

[98] D. He, Y. Chen, and J. Chen, "An efficient certificateless proxy signature scheme without pairing," *Math. Comput. Model.*, vol. 57, no. 9, pp. 2510–2518, May 2013.

[99] H. Qu, Z. Yan, X.-J. Lin, Q. Zhang, and L. Sun, "Certificateless public key encryption with equality test," pp. 76–92, 2018.

[100] A. S. Rawat and M. Deshmukh, "Efficient extended Diffie-Hellman key exchange protocol," in *2019 International Conference on Computing, Power and Communication Technologies (GUCON)*, Sep. 2019, pp. 447–451.

[101] Schneier, “Applied cryptography protocols,” *Algorithms and Source Code in C*, 1995.

[102] I. Psaras, “Decentralised Edge-Computing and IoT through distributed trust,” in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’18. New York, NY, USA: Association for Computing Machinery, Jun. 2018, pp. 505–507.

[103] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder, and F. Siebenlist, “X. 509 proxy certificates for dynamic delegation,” in *3rd annual PKI R&D workshop*, vol. 14, 2004.

[104] N. Vratonjic, J. Freudiger, V. Bindschaedler, and J.-P. Hubaux, “The inconvenient truth about web certificates,” in *Economics of Information Security and Privacy III*. Springer New York, 2013, pp. 79–117.

[105] C. Esposito, O. Tamburis, X. Su, and C. Choi, “Robust decentralised trust management for the internet of things by using game theory,” *Inf. Process. Manag.*, vol. 57, no. 6, p. 102308, Nov. 2020.

[106] Z. Lv, L. Qiao, A. Kumar Singh, and Q. Wang, “AI-empowered IoT security for smart cities,” *ACM Trans. Internet Technol.*, vol. 21, no. 4, pp. 1–21, Jul. 2021.

[107] T. Saba, K. Haseeb, A. A. Shah, A. Rehman, U. Tariq, and Z. Mehmood, “A Machine-Learning-Based Approach for Autonomous IoT Security,” *IT Prof.*, vol. 23, no. 3, pp. 69–75, May 2021.

[108] Y. Kortesniemi, D. Lagutin, T. Elo, and N. Fotiou, “Improving the privacy of IoT with decentralised identifiers (DIDs),” *Journal of Computer Networks and Communications*, vol. 2019, Mar. 2019.

[109] R. McPherson and J. Irvine, “Secure decentralised deployment of LoRaWAN sensors,” *IEEE Sens. J.*, vol. 21, no. 1, pp. 725–732, Jan. 2021.

[110] B. Farahani, F. Firouzi, and M. Luecking, “The convergence of IoT and Distributed Ledger Technologies (DLT): Opportunities, challenges, and solutions,” *Journal of Network and Computer Applications*, vol. 177, p. 102936, Mar. 2021.

[111] L. Lai, T. Zhou, Z. Cai, Z. Liang, and H. Bai, “A survey on security threats and solutions of bitcoin,” *Journal of Cybersecurity; Henderson*, vol. 3, no. 1, pp. 29–44, 2021.

[112] G. A. F. Rebello, G. F. Camilo, L. C. B. Guimarães, L. A. C. de Souza, G. A. Thomaz, and O. C. M. B. Duarte, “A security and performance analysis of proof-based consensus protocols,” *Ann. Telecommun.*, Nov. 2021.

[113] X.-L. Zhang, “Authenticated Key Exchange Protocol in One-Round,” in *Algorithms and Architectures for Parallel Processing.* Springer Berlin Heidelberg, 2009, pp. 226–233.

[114] J. I. Choi and K. R. B. Butler, “Secure multiparty computation and trusted hardware: Examining adoption challenges and opportunities,” *Security and Communication Networks*, vol. 2019, Apr. 2019.

[115] D. Evans, V. Kolesnikov, and M. Rosulek, “A pragmatic introduction to secure multi-party computation,” *Foundations and Trends® in*, 2017.

[116] N. Satoshi, “Bitcoin: A peer-to-peer electronic cash system,” *Consulted*, vol. 1, no. 2012, p. 28, 2008.

[117] S. Sayeed and H. Marco-Gisbert, “Assessing blockchain consensus and security mechanisms against the 51% attack,” *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, vol. 9, no. 9, p. 1788, Apr. 2019.

[118] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, “Securing Proof-of-Stake blockchain protocols,” in *Data Privacy Management, Cryptocurrencies and Blockchain Technology.* Springer International Publishing, 2017, pp. 297–315.

[119] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, “On security analysis of Proof-of-Elapsed-Time (PoET),” in *Stabilization, Safety, and Security of Distributed Systems.* Springer International Publishing, 2017, pp. 282–297.

[120] Z. Team and Others, “The zilliqa technical whitepaper,” *Retrieved Sept*, vol. 16, p. 2019, 2017.

[121] T. Nem, “Nem technical reference,” *URL https://nem. io/wpcontent/themes/nem/files/NEM_techRef. pdf*, 2018.

[122] S. Gauld, F. Von Ancoina, and R. Stadler, “‘the burst dymaxion an arbitrary scalable, energy efficient and anonymous transaction network based on colored tangles,” *Proc. CryptoGuru PoC SIG*, 2017.

[123] P. K. Singh, R. Singh, S. K. Nandi, and S. Nandi, “Managing smart home appliances with proof of authority and blockchain,” in *Innovations for Community Services.* Springer International Publishing, 2019, pp. 221–232.

[124] Xtrabytes, “Xtrabytes whitepaper,” 2021.

[125] F. Space, “FOAM whitepaper URL: https://www. foam. space/publicassets,” *FOAM_Whitepaper. pdf [accessed 2019-02-01][WebCite Cache ID 75s4C7tCr]*, 2018.

[126] P. Oscarson, “Information security fundamentals,” in *Security Education and Critical Infrastructures.* Springer US, 2003, pp. 95–107.

[127] D. A. McGrew, "Counter mode security: Analysis and recommendations," *Cisco Systems, November*, vol. 2, no. 4, 2002.

[128] D. Whiting, R. Housley, and N. Ferguson, "Counter with cbc-mac (ccm)," *Internet Engineering Task Force (IETF)*, 2003.

[129] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering: Design Principles and Practical Applications.* John Wiley & Sons, Feb. 2011.

[130] M. Vaidehi and B. J. Rabi, "Design and analysis of AES-CBC mode for high security applications," in *Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014.* ieeexplore.ieee.org, Jul. 2014, pp. 499–502.

[131] D. Sravana Kumar, C. H. Suneetha, and A. Chandrasekhar, "A block cipher using rotation and logical XOR operations," *ArXiv Pre-Print*, Feb. 2012.

[132] D. McGrew and J. Viega, "The Galois Counter Mode of operation (GCM)," *submission to NIST Modes of Operation Process*, vol. 20, pp. 0278–0070, 2004.

[133] D. Blazhevski, A. Bozhinovski, B. Stojchevska, and V. Pachovski, "Modes of operation of the aes algorithm," http://ciit.finki.ukim.mk/data/papers/10CiiT/10CiiT-46.pdf, 2013, accessed: 2022-3-27.

[134] Browserling, "Hash generation," 2021.

[135] R. Rivest and S. Dusse, "The MD5 message-digest algorithm," 1992.

[136] M. Ajtai, "The complexity of the pigeonhole principle," *Combinatorica*, vol. 14, no. 4, pp. 417–433, Dec. 1994.

[137] P. Gupta and S. Kumar, "A comparative analysis of SHA and MD5 algorithm," *Architecture*, vol. 1, p. 5, 2014.

[138] Q. H. Dang, "Secure hash standard," National Institute of Standards and Technology, Tech. Rep., Jul. 2015.

[139] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "Keccak," in *Advances in Cryptology – EUROCRYPT 2013.* Springer Berlin Heidelberg, 2013, pp. 313–314.

[140] J. Wetzels and W. Bokslag, "Sponges and engines: An introduction to Keccak and Keyak," *ArXiv Pre-Print*, Oct. 2015.

[141] A. Mittelbach and M. Fischlin, "Iterated hash functions in practice," in *The Theory of Hash Functions and Random Oracles: An Approach to Modern Cryptography*, A. Mittelbach and M. Fischlin, Eds. Cham: Springer International Publishing, 2021, pp. 585–618.

[142] I. Velásquez, A. Caro, and A. Rodríguez, "Authentication schemes and methods: A systematic literature review," *Information and Software Technology*, vol. 94, pp. 30–37, Feb. 2018.

[143] J. Ahamed and F. Khan, "An enhanced context-aware capability-based access control model for the internet of things in healthcare," in *2019 Sixth HCT Information Technology Trends (ITT)*, Nov. 2019, pp. 126–131.

[144] W. Newhouse, M. Ekstrom, J. Finke, and M. Harriston, "Securing property management systems," 2021.

[145] A. Calabrò, S. Daoudagh, and E. Marchetti, "Integrating access control and business process for GDPR compliance: A preliminary study," in *ITASEC*, 2019.

[146] S. Rautmare and D. M. Bhalerao, "MySQL and NoSQL database comparison for IoT application," in *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*. ieeexplore.ieee.org, Oct. 2016, pp. 235–238.

[147] J. Bhogal and I. Choksi, "Handling big data using NoSQL," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*. ieeexplore.ieee.org, Mar. 2015, pp. 393–398.

[148] A. Almutairi, M. Sarfraz, S. Basalamah, W. Aref, and A. Ghafoor, "A distributed access control architecture for cloud computing," *IEEE Softw.*, vol. 29, no. 2, pp. 36–44, Mar. 2012.

[149] R. Sandhu and Q. Munawer, "How to do discretionary access control using roles," in *Proceedings of the third ACM workshop on Role-based access control*, ser. RBAC '98. New York, NY, USA: Association for Computing Machinery, Oct. 1998, pp. 47–54.

[150] H. Lindqvist, "Mandatory access control," *Master's thesis in computing science, Umea University, Department of Computing Science, SE-901*, vol. 87, 2006.

[151] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, Feb. 1996.

[152] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, "Attribute-Based access control," *Computer*, vol. 48, no. 2, pp. 85–88, Feb. 2015.

[153] J. L. Hernández-Ramos, A. J. Jara, L. Marin, and A. F. Skarmeta, "Distributed capability-based access control for the internet of things," *Journal of Internet Services and Information Security (JISIS)*, vol. 3, no. 3/4, pp. 1–16, 2013.

[154] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," in *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, Nov. 2011, pp. 91–98.

[155] D. He, J. Bu, S. Zhu, S. Chan, and C. Chen, “Distributed access control with privacy support in wireless sensor networks,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3472–3481, Oct. 2011.

[156] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, “Zcash protocol specification,” *GitHub: San Francisco, CA, USA*, 2016.

[157] R. K. Raman and L. R. Varshney, “Dynamic distributed storage for blockchains,” in *2018 IEEE International Symposium on Information Theory (ISIT)*. ieeexplore.ieee.org, Jun. 2018, pp. 2619–2623.

[158] J. Roth, F. Schär, and A. Schöpfer, “The tokenization of assets: Using blockchains for equity crowdfunding,” in *Theories of Change: Change Leadership Tools, Models and Applications for Investing in Sustainable Development*, K. Wendt, Ed. Cham: Springer International Publishing, 2021, pp. 329–350.

[159] H. Treiblmaier and C. Sillaber, “The impact of blockchain on e-commerce: a framework for salient research topics,” *Electron. Commer. Res. Appl.*, vol. 48, p. 101054, 2021.

[160] K. Kypriotaki, E. Zamani, and G. Giaglis, “From bitcoin to decentralized autonomous corporations-extending the application scope of decentralized peer-to-peer networks and blockchains,” in *International conference on enterprise information systems*, vol. 2. scitepress.org, 2015, pp. 284–290.

[161] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) protocol version 1.2,” Internet Engineering Task Force, Tech. Rep., Aug. 2008.

[162] S. Contini and Y. L. Yin, “Forgery and partial Key-Recovery attacks on HMAC and NMAC using hash collisions,” in *Advances in Cryptology – ASIACRYPT 2006*. Springer Berlin Heidelberg, 2006, pp. 37–53.

[163] Espressif, “IoT Development Framework, Espressif Systems,” 2022.

[164] ——, “Watchdogs - ESP32 - — ESP-IDF programming guide latest documentation,” 2022.

[165] M. Zwerg, A. Baumann, R. Kuhn, M. Arnold, R. Nerlich, M. Herzog, R. Ledwa, C. Sichert, V. Rzehak, P. Thanigai, and B. O. Eversmann, “An 82μA/MHz microcontroller with embedded FeRAM for energy-harvesting applications,” in *2011 IEEE International Solid-State Circuits Conference*, Feb. 2011, pp. 334–336.

[166] S. Kamath and J. Lindh, “Measuring Bluetooth low energy power consumption,” *Texas instruments application note AN092, Dallas*, 2010.

[167] W. Waluyo, A. Widura, F. Hadiatna, and D. Anugerah, “Comparative power and energy consumptions between scheduled and fuzzy controlling on an IoT-based vertical

farming," in *2021 3rd International Conference on High Voltage Engineering and Power Systems (ICHVEPS)*, Oct. 2021, pp. 278–282.

[168] S. Mödersheim, L. Viganò, and D. Basin, "Constraint differentiation: Search-space reduction for the constraint-based analysis of security protocols," *J. Comput. Secur.*, vol. 18, no. 4, pp. 575–618, Jun. 2010.

[169] M. Turuani, "The CL-Atse protocol analyser," in *Term Rewriting and Applications.* Springer Berlin Heidelberg, 2006, pp. 277–286.

[170] A. Armando, R. Carbone, and L. Compagna, "SATMC: A SAT-Based model checker for Security-Critical systems," in *Tools and Algorithms for the Construction and Analysis of Systems.* Springer Berlin Heidelberg, 2014, pp. 31–45.

[171] N. Liu, W.-Y. Zhu, and Y.-F. Zhu, "Security protocol analysis based on rewriting approximation," in *2009 Second International Symposium on Electronic Commerce and Security*, vol. 1. ieeexplore.ieee.org, May 2009, pp. 318–322.

[172] E. A. Emerson and C. S. Jutla, "Tree automata, mu-calculus and determinacy," in *FoCS*, vol. 91. Citeseer, 1991, pp. 368–377.

[173] D. N. Le, L. Le Tuan, and M. N. Dang Tuan, "Smart-building management system: An Internet-of-Things (IoT) application business model in vietnam," *Technol. Forecast. Soc. Change*, vol. 141, pp. 22–35, Apr. 2019.

[174] A. T. Domingues, C. A. Marianne, and F. D. Castro, "Building blocks for the development of an IoT business model," *Journal of Strategy and Management*, vol. 13, no. 1, pp. 15–32, Jan. 2019.

[175] I. Lee, "The internet of things for enterprises: An ecosystem, architecture, and IoT service business model," *Internet of Things*, vol. 7, p. 100078, Sep. 2019.

[176] M. Fayad, A. Mostefaoui, S. Chouali, and S. Benbernou, "Model-Oriented methodology for developing a social based healthcare system," in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, ser. Q2SWinet '20. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 101–107.

[177] K. Sewchurran and D. Petkov, "A systemic framework for business process modeling combining soft systems methodology and UML," *IRMJ*, vol. 20, no. 3, pp. 46–62, Jul. 2007.

[178] C. Vasilakis, D. Lecznarowicz, and C. Lee, "Developing model requirements for patient flow simulation studies using the unified modelling language (UML)," *Journal of Simulation*, vol. 3, no. 3, pp. 141–149, Sep. 2009.

[179] M. Iwashita, S. Tanimoto, and Y. Fujinoki, "Approaches to analyze requirements of billing management for cloud computing services," in *2013 IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS)*. ieeexplore.ieee.org, Jun. 2013, pp. 17–22.

[180] C. G. Sørensen, S. Fountas, E. Nash, L. Pesonen, D. Bochtis, S. M. Pedersen, B. Basso, and S. B. Blackmore, "Conceptual model of a future farm management information system," *Comput. Electron. Agric.*, vol. 72, no. 1, pp. 37–47, Jun. 2010.

[181] J.-F. Guay and J.-P. Waaub, "Application of a territorial soft system approach for conceptual modeling of an agroecosystem," *Environment Systems and Decisions*, vol. 35, no. 3, pp. 363–374, Sep. 2015.

[182] G. Lazaroiu, M. Andronie, C. Uţă, and I. Hurloiu, "Trust management in organic agriculture: Sustainable consumption behavior, environmentally conscious purchase intention, and healthy food choices," *Front Public Health*, vol. 7, p. 340, Nov. 2019.

[183] G. Kiradoo, "A review of the challenges of green marketing: The current scenario," Dec. 2019, accessed: 2021-12-29.

[184] W. Lin, X. Huang, H. Fang, V. Wang, Y. Hua, J. Wang, H. Yin, D. Yi, and L. Yau, "Blockchain technology in current agricultural systems: From techniques to applications," *IEEE Access*, vol. 8, pp. 143 920–143 937, 2020.

[185] J. Li and X. Wang, "Research on the application of blockchain in the traceability system of agricultural products," in *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*. ieeexplore.ieee.org, May 2018, pp. 2637–2640.

[186] A. O. Bada, A. Damianou, C. M. Angelopoulos, and V. Katos, "Towards a green blockchain: A review of consensus mechanisms and their energy consumption," in *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. ieeexplore.ieee.org, Jul. 2021, pp. 503–511.

[187] J. Beccuti, C. Jaag, and Others, "The Bitcoin mining game: On the optimality of honesty in proof-of-work consensus mechanism," *Swiss Economics Working Paper 0060*, 2017.

[188] N. Lasla, L. Alsahan, M. Abdallah, and M. Younis, "Green-PoW: An Energy-Efficient blockchain Proof-of-Work consensus algorithm," *ArXiv Pre-Print*, Jul. 2020.

[189] R. Xu, Y. Chen, E. Blasch, and G. Chen, "Microchain: A hybrid consensus mechanism for lightweight distributed ledger for IoT," *ArXiv Pre-Print*, Sep. 2019.

[190] K. M. S. Rana, M. Jahan, Z. Ferdous, and others, "Production performance of lettuce (lactuca sativa): aquaponics versus traditional soil," *Asian Journal of Medical*, 2018.

[191] S. Alderman, "The practicality and sustainability of aquaponic agriculture versus traditional agriculture with emphasis on application in the middle east," *Communications in Computer and Information Science*, 2015.

[192] M. Shafahi and D. Woolston, "Aquaponics: A sustainable food production system," in *ASME 2014 International Mechanical Engineering Congress and Exposition.* American Society of Mechanical Engineers Digital Collection, Mar. 2015.

[193] R. Holz, L. Braun, N. Kammenhuber, and G. Carle, "The SSL landscape: a thorough analysis of the X.509 PKI using active and passive measurements," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, ser. IMC '11. New York, NY, USA: Association for Computing Machinery, Nov. 2011, pp. 427–444.

[194] M. Aly, F. Khomh, M. Haoues, A. Quintero, and S. Yacout, "Enforcing security in internet of things frameworks: A systematic literature review," p. 100050, 2019.

[195] N. Javaid, "Integration of context awareness in internet of agricultural things," *ICT Express*, Sep. 2021.

[196] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with ZigBee/802.15.4," in *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Apr. 2012, pp. 232–237.

[197] X. Larrucea, M. Moffie, D. Mor, and Others, "Enhancing GDPR compliance through data sensitivity and data hiding tools," *JUCS-Journal of Universal Computer Science*, 2021.

[198] M. Kangwa, C. S. Lubobya, and J. Phiri, "Prevention of personally identifiable information leakage in e-commerce via offline data minimisation and pseudonymisation," *Int. J. Innov. Sci. Res. Technol*, vol. 6, no. 1, pp. 209–212, 2021.

[199] A. Goldsteen, G. Ezov, R. Shmelkin, M. Moffie, and A. Farkash, "Data minimization for GDPR compliance in machine learning models," 2021.

[200] K. Broen, R. Trangucci, and J. Zelner, "Measuring the impact of spatial perturbations on the relationship between data privacy and validity of descriptive statistics," *Int. J. Health Geogr.*, vol. 20, no. 1, p. 3, Jan. 2021.

[201] M. Slokom and M. Larson, "Doing data right: How lessons learned working with conventional data should inform the future of synthetic data for recommender systems," *ArXiv Pre-Print*, Oct. 2021.

[202] M. Barati, G. S. Aujla, J. T. Llanos, K. A. Duodu, O. F. Rana, M. Carr, and R. Rajan, "Privacy-Aware cloud auditing for GDPR compliance verification in online healthcare," *IEEE Trans. Ind. Inf.*, pp. 1–1, 2021.

[203] E. Barker and A. Roginsky, “Transitioning the use of cryptographic algorithms and key lengths,” National Institute of Standards and Technology, Tech. Rep., 2018.

[204] L. E. Kane, J. J. Chen, R. Thomas, V. Liu, and M. Mckague, “Security and performance in IoT: A balancing act,” *IEEE Access*, vol. 8, pp. 121 969–121 986, 2020.

[205] D. Gan and R. Heartfield, “Social engineering in the internet of everything,” *Cutter IT Journal*, vol. 29, no. 7, pp. 20–29, Jul. 2016.

[206] C. Del-Valle-Soto, L. J. Valdivia, R. Velázquez, L. Rizo-Dominguez, and J.-C. López-Pimentel, “Smart campus: An experimental performance comparison of collaborative and cooperative schemes for wireless sensor network,” *Energies*, vol. 12, no. 16, p. 3135, Aug. 2019.

[207] D. Díaz-Sánchez, A. Marín-Lopez, F. Almenarez, P. Arias, and R. S. Sherratt, “TLS/PKI challenges and certificate pinning techniques for IoT and M2M secure communications,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.

[208] M. Y. Afanasev, A. A. Krylova, S. A. Shorokhov, Y. V. Fedosov, and A. S. Sidorenko, “A design of cyber-physical production system prototype based on an ethereum private network,” in *2018 22nd Conference of Open Innovations Association (FRUCT)*, May 2018, pp. 3–11.

[209] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, “DTLS based security and two-way authentication for the internet of things,” *Ad Hoc Networks*, vol. 11, no. 8, pp. 2710–2723, Nov. 2013.

[210] N. Nikolov and O. Nakov, “Research of Secure Communication of ESP32 IoT Embedded System to.NET core cloud structure using MQTTS SSL/TLS,” in *2019 IEEE XXVIII International Scientific Conference Electronics (ET)*, Sep. 2019, pp. 1–4.

[211] H. Chiang, J. Hong, K. Kiningham, L. Riliskis, P. Levis, and M. Horowitz, “Tethys: Collecting sensor data without infrastructure or trust,” in *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*. ieeexplore.ieee.org, Apr. 2018, pp. 249–254.

[212] Arduino, “TLS (transport Layer Security) running on an arduino UNO,” Dec. 2016.

[213] O. Galinina, K. Mikhaylov, S. Andreev, A. Turlikov, and Y. Koucheryavy, “Smart home gateway system over bluetooth low energy with wireless energy transfer capability,” *Eurasip J. Wirel. Commun. Network.*, vol. 2015, no. 1, pp. 1–18, Jun. 2015.

[214] V. Petrov, S. Edelev, M. Komar, and Y. Koucheryavy, “Towards the era of wireless keys: How the IoT can change authentication paradigm,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, Mar. 2014, pp. 51–56.

[215] M. K. Aditia, S. Paida, F. Altaf, and S. Maity, “Certificate-less public key encryption for secure e-healthcare systems,” in *2019 IEEE Conference on Information and Communication Technology.* ieeexplore.ieee.org, Dec. 2019, pp. 1–5.

[216] P. Tarasov and H. Tewari, “Internet voting using zcash,” *IACR Cryptol. ePrint Arch.*, vol. 2017, p. 585, 2017.

[217] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, “Everything you wanted to know about the blockchain: Its promise, components, processes, and problems,” *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6–14, Jul. 2018.

[218] D. Fakhri and K. Mutijarsa, “Secure IoT communication using blockchain technology,” in *2018 International Symposium on Electronics and Smart Devices (ISESD)*, Oct. 2018, pp. 1–6.

[219] W. F. Silvano and R. Marcelino, “Iota tangle: A cryptocurrency to communicate Internet-of-Things data,” *Future Gener. Comput. Syst.*, vol. 112, pp. 307–319, Nov. 2020.

[220] A. Pieroni, N. Scarpato, and L. Felli, “Blockchain and IoT Convergence—A systematic survey on technologies, protocols and security,” *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, vol. 10, no. 19, p. 6749, Sep. 2020.

[221] N. Teslya and I. Ryabchikov, “Blockchain platforms overview for industrial IoT purposes,” in *2018 22nd Conference of Open Innovations Association (FRUCT)*, May 2018, pp. 250–256.

[222] G. Leduc, S. Kubler, and J.-P. Georges, “Innovative blockchain-based farming marketplace and smart contract performance evaluation,” *J. Clean. Prod.*, vol. 306, p. 127055, Jul. 2021.

[223] T. H. Pranto, A. A. Noman, A. Mahmud, and A. B. Haque, “Blockchain and smart contract for IoT enabled smart agriculture,” p. e407, 2021.

[224] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng, “When internet of things meets blockchain: Challenges in distributed consensus,” *IEEE Netw.*, vol. 33, no. 6, pp. 133–139, Nov. 2019.

[225] M. Gorodnichev, A. Kukharenko, E. Kukharenko, and T. Salutina, “Methods of developing systems based on blockchain,” https://www.fruct.org/publications/acm24/files/Gor.pdf, 2019, accessed: 2021-5-11.

[226] M. Salimitari, M. Chatterjee, and Y. Fallah, “A survey on consensus methods in blockchain for resource-constrained IoT networks,” 2020.

[227] D. Puthal, S. P. Mohanty, V. P. Yanambaka, and E. Kougianos, “PoAh: A novel consensus algorithm for fast scalable private blockchain for large-scale IoT frameworks,” *Distributed, Parallel, and Cluster Computing*, Jan. 2020.

[228] T. Krupa, M. Ries, I. Kotuliak, K. Košťál, and R. Bencel, “Security issues of smart contracts in ethereum platforms,” in *2021 28th Conference of Open Innovations Association (FRUCT)*, Jan. 2021, pp. 208–214.

[229] Espressif, “Fatal errors ESP2,” 2022.

[230] B. Gill, “The wheel of reincarnation turns again: Time for another go at the clipper chip,” *Researchgate,[SI], dec*, 2019.

[231] E. Barker and Q. Dang, “NIST Special Publication 800-57 part 1, revision 4,” *NIST, Tech. Rep*, vol. 16, 2016.

[232] J. A. Ali, Q. Nasir, and F. T. Dweiri, “Business continuity framework for internet of things (IoT) services,” *International Journal of System Assurance Engineering and Management*, vol. 11, no. 6, pp. 1380–1394, Dec. 2020.

[233] M. Kolomeets and A. Chechulin, “Analysis of the malicious bots market,” in *2021 29th Conference of Open Innovations Association (FRUCT)*, May 2021, pp. 199–205.

[234] M. Gorbunova, P. Masek, M. Komarov, and A. Ometov, “Distributed ledger technology: State-of-the-art and current challenges,” *Comput. Sci. Inf. Syst.*, vol. 19, no. 01, pp. 37–37, 2021.

[235] I. Struchkov, A. Lukashin, B. Kuznetsov, I. Mikhalev, and Z. Mandrusova, “Agent-Based modeling of blockchain decentralized financial protocols,” in *2021 29th Conference of Open Innovations Association (FRUCT)*, May 2021, pp. 337–343.

[236] M. Bulygin and D. Namiot, “A new approach to clustering districts and connections between them based on cellular operator data,” in *2021 29th Conference of Open Innovations Association (FRUCT)*, May 2021, pp. 71–80.

[237] S. Leech, D. Malone, and J. Dunne, “Heads or tails: A framework to model supply chain heterogeneous messages,” in *2021 30th Conference of Open Innovations Association FRUCT*, Oct. 2021, pp. 129–140.

[238] A. J. Goldsmith and S. B. Wicker, “Design challenges for energy-constrained ad hoc wireless networks,” *IEEE Wirel. Commun.*, vol. 9, no. 4, pp. 8–27, Aug. 2002.

[239] S. Alnefaie, S. Alshehri, and A. Cherif, “A survey on access control in IoT: models, architectures and research opportunities,” *Int. J. Secur. Netw.*, vol. 16, no. 1, pp. 60–76, Jan. 2021.

[240] S. Pal, A. Dorri, and R. Jurdak, “Blockchain for IoT access control: Recent trends and future research directions,” *Journal of Network and Computer Applications*, Jun. 2021.

[241] P. Kanakaraja and E. Al, “IoT enabled BLE and LoRa based indoor localization without GPS,” *TURCOMAT*, vol. 12, no. 4, pp. 1637–1651, Apr. 2021.

[242] L. Vishwakarma and D. Das, “SCAB - IoTA: Secure communication and authentication for IoT applications using blockchain,” *J. Parallel Distrib. Comput.*, vol. 154, pp. 94–105, Aug. 2021.

[243] O. Green, “HashGraph—Scalable hash tables using a sparse graph data structure,” *ACM Trans. Parallel Comput.*, vol. 8, no. 2, pp. 1–17, Jul. 2021.

[244] I. A. Prostov, S. S. Amfiteatrova, and N. G. Butakova, “Construction and security analysis of private directed acyclic graph based systems for internet of things,” in *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus).* ieeexplore.ieee.org, Jan. 2021, pp. 2394–2398.

[245] J. D. Moffett, M. Sloman, and K. Twidle, “Specifying discretionary access control policy for distributed systems,” *Comput. Commun.*, vol. 13, no. 9, pp. 571–580, 1990.

[246] R. S. Sandhu, “Role-based access control,” in *Advances in computers.* Elsevier, 1998, vol. 46, pp. 237–286.

[247] E. C. Lupu, D. A. Marriott, M. S. Sloman, and N. Yialelis, “A policy based role framework for access control,” in *Proceedings of the first ACM Workshop on Role-based access control*, ser. RBAC ’95. New York, NY, USA: Association for Computing Machinery, Dec. 1996, pp. 11–es.

[248] F. Kausar, M. A. K. Sadiq, and H. M. Asif, “Convergence of blockchain in IoT applications for heterogeneous networks,” in *Real-Time Intelligence for Heterogeneous Networks: Applications, Challenges, and Scenarios in IoT HetNets*, F. Al-Turjman, Ed. Cham: Springer International Publishing, 2021, pp. 71–86.

[249] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, “Green IoT: An investigation on energy saving practices for 2020 and beyond,” *IEEE Access*, vol. 5, pp. 15 667–15 681, 2017.

[250] H. Tschofenig and T. Fossati, “Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things,” in *RFC 7925.* Internet Engineering Task Force, 2016.

[251] E. P. Frigieri, D. Mazzer, and L. Parreira, “M2M protocols for constrained environments in the context of IoT: A comparison of approaches,” in *International Telecommunications Symposium*, 2015, p. 5.

[252] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger, "Poseidon: A new hash function for zero-knowledge proof systems," in *30th {USENIX} Security Symposium ({USENIX} Security 21).* usenix.org, 2021.

[253] B. B. Rao and A. A. Waoo, "DESIGN a NOVEL APPROACH FOR TOKEN BASED AUTHENTICATION IN IOT NETWORKS," *Ilkogretim Online*, vol. 20, no. 4, 2021.

[254] R. Bharanidharan, "A novel blockchain approach for improve the performance of network security using polynomial ephemeral Blockchain-Based secure routing in wireless sensor network," pp. 5598–5604, 2020.

[255] A. D. Miyazaki and A. Fernandez, "Consumer perceptions of privacy and security risks for online shopping," *J. Consum. Aff.*, vol. 35, no. 1, pp. 27–44, Jun. 2001.

[256] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 3–16.

[257] D. Larimer, "Transactions as proof-of-stake," *Nov-2013*, vol. 909, 2013.

[258] M. Castro, B. Liskov, and Others, "Practical byzantine fault tolerance," in *OsDI*, vol. 99. usenix.org, 1999, pp. 173–186.

[259] B. Xiao, C. Jin, Z. Li, B. Zhu, X. Li, and D. Wang, "Proof of importance: A consensus algorithm for importance based on dynamic authorization," in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, ser. WI-IAT '21. New York, NY, USA: Association for Computing Machinery, Dec. 2021, pp. 510–513.

[260] S. Aggarwal and N. Kumar, "Cryptographic consensus mechanisms," in *Advances in Computers.* Elsevier, 2021, vol. 121, pp. 211–226.

[261] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," https://iris.uniroma1.it/bitstream/11573/1337256/1/DeAngelis_PBFT_2018.pdf, 2018, accessed: 2022-4-15.

[262] A. Imteaj, M. Hadi Amini, and P. M. Pardalos, "Toward smart contract and consensus mechanisms of blockchain," in *Foundations of Blockchain: Theory and Applications*, A. Imteaj, M. H. Amini, and P. M. Pardalos, Eds. Cham: Springer International Publishing, 2021, pp. 15–28.

[263] A. Bates, B. Mood, M. Valafar, and K. Butler, "Towards secure provenance-based access control in cloud environments," in *Proceedings of the third ACM conference on Data and application security and privacy*, ser. CODASPY '13. New York, NY, USA: Association for Computing Machinery, Feb. 2013, pp. 277–284.

[264] S. Daoudagh, E. Marchetti, V. Savarino, R. Di Bernardo, and M. Alessi, "How to improve the GDPR compliance through consent management and access control," in *ICISSP*, 2021, pp. 534–541.

[265] J. Frank and M. Bishop, "Extending the take-grant protection system," https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.907&rep=rep1&type=pdf, 1996, accessed: 2021-9-15.

[266] K. J. Biba, "Integrity considerations for secure computer systems," MITRE CORP BEDFORD MA, Tech. Rep., 1977.

[267] B. Danner, C. Muckenhirn, T. Valle, C. McElveen, J. Bragdon-Handfield, and A. Colegrove, "Multilevel security feasibility in the M&S training environment," in *Proceedings of the Interservice/Industry Training Simulation and Education Conference (I/ITSEC).* Citeseer, 2002.

[268] N. Ferguson and B. Schneier, *Practical cryptography.* Wiley, 2003.

[269] A. M. Pinto, "An introduction to the use of zk-SNARKs in blockchains," in *Mathematical Research for Blockchain Economy.* Cham: Springer International Publishing, 2020, pp. 233–249.

[270] M. Shen, H. Liu, L. Zhu, K. Xu, H. Yu, X. Du, and M. Guizani, "Blockchain-Assisted secure device authentication for Cross-Domain industrial IoT," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 942–954, May 2020.

Appendices

Appendix A

Timer Sketch Pseudocode

Where J energy in Joules = T (time in seconds) x P (Power in Watts), outlined below is the sketch code for each mbedtls function timed in microseconds.

A.1 Key pair generation, sign and verify

RSA key strengths represent the current options for asymmetric key generation and authentication of, whilst Elliptic Curve Digital Signature Algorithm (ECDSA), offer shorter versions of the same RSA bit-strength range:

A.1.1 RSA: 5 key strengths

1. 2048
2. 3072
3. 4096
4. 7680
5. 15,360

A.1.2 ECDSA: 11 curves

1. SECP224R1
2. SECP256R1
3. SECP384R1
4. SECP521R1
5. SECP224K1
6. SECP256K1

7. BP256R1
8. BP384R1
9. BP512R1
10. Curve448
11. Curve25519

```
1. Define key strength or curve
2. Create and initialise necessary context for either RSA or ECDSA
3. Start timer
4. Begin entropy and seed from random number generator
5. Generate the keypair
6. Complete and check private key
7. Define data to be signed
8. Sign data
9. End timer
10. Print time
11. Print signature
12. Free contexts
```

A.2 Key exchange

RSA Public Key Cryptography (PKC) is asymmetric encryption, whilst Elliptic Curve Diffie-Hellman Ephemeral (ECDHE), is mutual key exchange:

A.2.1 RSA-PKC: Single key (typical TLS), and multiple keys (re-configured TLS)

1. 2048-bit strength PKC with ECDSA for AES 128-bit key
2. 2048-bit strength PKC with ECDSA for three CIA keys separately

A.2.2 ECDHE: Single key (typical TLS), and multiple keys (re-configured TLS)

1. ECDHE in Curve25519 with ECDSA for AES 128-bit key
2. ECDHE in Curve25519 for three CIA keys separately

```
1. Define key strength or curve
2. Create and initialise necessary context for either RSA or ECDHE
3. Start timer
```

```
4. Begin entropy and seed from random number generator
5. Generate the keypair
6. Complete and check private key
7. Compute message hash
8. Create signature and verify
9. Sign message, verify
10. Encrypt message in RSA or ECDHE
11. Decrypt message in RSA or ECDHE
12. End timer
13. Print time
14. Print signature
15. Print message
16. Free contexts
```

A.3 Integrity and message authentication functions

SHA and HMAC functions working together, and producing an authentication key:

A.3.1 HMAC: 4 variables in 64 bytes payload

1. HMAC-SHA224
2. HMAC-SHA256
3. HMAC-SHA384
4. HMAC-SHA512

A.3.2 HMAC: 4 variables in 512 bytes payload

1. HMAC-SHA224
2. HMAC-SHA256
3. HMAC-SHA384
4. HMAC-SHA512

```
1. Define HMAC SHA bit-strength
2. Initialise message digest context
3. Start timer
4. Begin HMAC operation by specifying key
5. Perform message digest operation by applying key to plaintext message
6. Collect resulting HMAC
7. End timer
```

```
8. Print time
9. Print HMAC
10. Free contexts
```

A.4 Full CIA by AEAD and collective components:

Full AEAD operations by confidentiality, integrity and availability functions together:

A.4.1 AES128: 5 modes in 64 bytes payload

1. AES128-GCM
2. AES128-CBC-HMAC-SHA256
3. AES128-CTR-HMAC-SHA256
4. AES128-CFB8-HMAC-SHA256
5. AES128-CFB128-HMAC-SHA256

A.4.2 AES128: 5 modes in 512 bytes payload

1. AES128-GCM
2. AES128-CBC-HMAC-SHA256
3. AES128-CTR-HMAC-SHA256
4. AES128-CFB8-HMAC-SHA256
5. AES128-CFB128-HMAC-SHA256

```
1. Define AES encryption strength, HMAC key, mode variables
    (In this example of CBC, an IV (Initialisation Vector)
2. Initialise encryption and message digest context
3. Start timer
4. Begin AEAD operation by specifying HMAC key and AES key
5. Perform full operation by applying keys to plaintext message
6. Collect resulting HMAC and ciphertext
7. End timer
8. Print time
9. Print HMAC and ciphertext
10. Free contexts
```

A.5 Data exchange

A.5.1 From server to client: Fully protected GCM and collective components, typical TLS

The following operations are repeats of previous pseudocode:

1. **AES128 in GCM mode:** AEAD pseudocode with mode of GCM.

2. **AES128 in full CCM:** AEAD pseudocode same as example CBC.

A.5.2 From client to server: Selective components and excluding privacy, the re-configuration

The following operation is a repeat of previous pseudocode:

1. **AES128 in CCM without confidentiality:** The same as HMAC-SHA256 without encryption. In this example, the AES encryption key is created, in addition to the HMAC key and the IV, but the data is not actually encrypted. The key is kept for future sessions where encryption is required, such as with instructional code of shared key refresh, typically from the server to the client, as opposed to from the client to the server.

A.6 Energy Consumption Comparison Over 24-hours in Authentication and Data Exchange

A.6.1 Readings: 16 character readings comparing typical TLS and the re-configuration

A combination of authentication and data exchange operations in small data strings:

1. ECDHE in Curve25519 with ECDSA for AES128-bit key with readings sent in GCM

2. ECDHE in Curve25519 with ECDSA for AES128-bit key with readings sent in CCM

3. ECDHE in Curve25519 for three keys and readings including confidentiality, Pi to ESP32

4. ECDHE in Curve25519 for three keys and readings excluding confidentiality, ESP32 to Pi

A.6.2 Instructions: 256 character instructions comparing typical TLS and the re-configuration

A combination of authentication and data exchange operations in longer data strings:

1. ECDHE in Curve25519 with ECDSA for AES128-bit key with readings sent in GCM
2. ECDHE in Curve25519 with ECDSA for AES128-bit key with readings sent in CCM
3. ECDHE in Curve25519 for three keys and readings including confidentiality, Pi to ESP32
4. ECDHE in Curve25519 for three keys and readings excluding confidentiality, ESP32 to Pi

Appendix B

Formal Verification Code

B.1 Device authorisation

B.1.1 Alice-Bob notation

1. `A → B : Ka`

2. `B → A : {{SSID, Kas, Kb}Kb-1}Ka`

3. `A → B : {{Na}Kb}Ka-1`

4. `B → A : {{Na}Ka}Kb-1`

B.1.2 CAS+

Four device authorisation data exchanges:

```
protocol DA;
identifiers
A, B : user;
KPa KPb : public_key;
SSID, Kas, Na : number;

messages
1. A -> B : KPa
2. B -> A : {{A, B, SSID, Kas, KPb}KPb'}KPa
3. A -> B : {{A, B, Na}KPb}KPa'
4. B -> A : {{Na}KPa}KPb'
```

```
knowledge
A : A,B,KPa,KPb;
B : A,B,KPa,KPb;

session_instances
    [A:alice,B:bob,KPa:kpa,KPb:kpb];

goal
secrecy_of Kas [A,B];
secrecy_of Na [A,B];
```

B.1.3 ProVerif input

1.
```
(* Declarations *)
free c:channel.
free skA:bitstring.
fun pk(bitstring):bitstring.

(* Protocol *)
let agentA(skA:bitstring) =
    let pkA = pk(skA) in
    out(c, pkA).

let agentB() =
    in(c, pkA:bitstring);
    0.

(* Process *)
process
    ( !agentA(skA) | !agentB() )
```

2.
```
(* Declarations *)
free c:channel.
free Ka:bitstring.
free Kb:bitstring.
free SSID:bitstring.
free Ks:bitstring [private].
query attacker(Ks).

fun pk(bitstring):bitstring.
fun sk(bitstring):bitstring.
fun sign(bitstring, bitstring):bitstring.
fun checksign(bitstring, bitstring):bitstring.
fun aenc(bitstring, bitstring):bitstring.
fun adec(bitstring, bitstring):bitstring.

(* Protocol *)
let agentB(Ka:bitstring, Ks:bitstring, Kb:bitstring) =
    let msg = (SSID, Ks, Kb) in
    let signed_msg = sign(msg, sk(Kb)) in
    out(c, aenc(signed_msg, Ka)).


let agentA(Ka:bitstring, Kb:bitstring) =
    in(c, enc_msg:bitstring);
    let signed_msg = adec(enc_msg, Ka) in
    let msg = checksign(signed_msg, Kb) in
    let (SSID_rec:bitstring, Ks_rec:bitstring, Kb_rec:bitstring) = msg in
    0.

(* Process *)
process
    ( !agentB(Ka, Ks, Kb) | !agentA(Ka, Kb) )
```

3.

```
(* Declarations *)
free c:channel.
free Ka:bitstring.
free Kb:bitstring.
free Na:bitstring [private].
query attacker(Na).

fun pk(bitstring):bitstring.
fun sk(bitstring):bitstring.

fun aenc(bitstring, bitstring):bitstring.
fun adec(bitstring, bitstring):bitstring.

(* Protocol *)
let agentA(Ka:bitstring, Kb:bitstring, Na:bitstring) =
    let msg = aenc(Na, Kb) in
    out(c, aenc(msg, sk(Ka))).
```

```
let agentB(Ka:bitstring, Kb:bitstring) =
    in(c, enc_msg:bitstring);
    let msg = adec(enc_msg, pk(Ka)) in
    let Na_rec = adec(msg, sk(Kb)) in
    0.

(* Process *)
process
    ( !agentA(Ka, Kb, Na) | !agentB(Ka, Kb) )
```

4.
```
(* Declarations *)
free c:channel.
free Ka:bitstring.
free Kb:bitstring.
free Na:bitstring [private].
query attacker(Na).

fun pk(bitstring):bitstring.
fun sk(bitstring):bitstring.

fun aenc(bitstring, bitstring):bitstring.
fun adec(bitstring, bitstring):bitstring.

(* Protocol *)
let agentB(Ka:bitstring, Kb:bitstring, Na:bitstring) =
    let msg = aenc(Na, Ka) in
    out(c, aenc(msg, sk(Kb))).

let agentA(Ka:bitstring, Kb:bitstring) =
    in(c, enc_msg:bitstring);
    let msg = adec(enc_msg, pk(Kb)) in
    let Na_rec = adec(msg, sk(Ka)) in
    0.

(* Process *)
process
    ( !agentB(Ka, Kb, Na) | !agentA(Ka, Kb) )
```

B.2 Cluster integrity attestation

B.2.1 Alice-Bob notation

1. `A → B .. N : {{SSID, Ka, Kas}Kb .. n}Ka-1`

2. `B .. N → A : {{Shb .. n, Kb .. n}Ka}Kb . . n-1`

3. `A : SHa XOR SHb .. n = RH`

B.2.2 CAS+

Three attestation data exchanges:

```
protocol CIA;
identifiers
A, B, N : user;
KPa, KPb, KPn : public_key;
SSID, Kas, Ka, Shb, Shn, SHa, RH : number;

messages
1. A -> B, N : {{A, SSID, Ka, Kas}KPn}KPa’
2. B, N -> A : {{B, N, Shb, Shn, KPb, KPn}KPa}KPb’, KPn’
3. A : calculate_xor(SHa, Shb, Shn)

knowledge
A : A, B, N, KPa, KPb, KPn;
B : A, B, N, KPa, KPb, KPn;
N : A, B, N, KPa, KPb, KPn;

session_instances
    [A:alice,B:bob,N:nonce,KPa:kpa,KPb:kpb,KPn:kpn];

goal
secrecy_of Kas [A,B,N];
secrecy_of SHa, Shb, Shn [A,B,N];
```

B.2.3 ProVerif input

1.
```
(* Declarations *)
free c:channel.
free Ka:bitstring.
free Kb:bitstring.
free Kc:bitstring.
free SSID:bitstring.
free Ks:bitstring [private].
query attacker(Ks).

fun pk(bitstring):bitstring.
fun sk(bitstring):bitstring.

fun aenc(bitstring, bitstring):bitstring.
fun adec(bitstring, bitstring):bitstring.

(* Protocol *)
let agentA(Ka:bitstring, Kb:bitstring, Kc:bitstring, SSID:bitstring,
    Ks:bitstring) =
    let msg = aenc((SSID, Ka, Ks), Kb) in
    let msg2 = aenc((SSID, Ka, Ks), Kc) in
    out(c, aenc((msg, msg2), sk(Ka))).

let agentB(Ka:bitstring, Kb:bitstring) =
    in(c, enc_msg:bitstring);
    let (msg1:bitstring, msg2:bitstring) =
        adec(enc_msg, pk(Ka)) in
    let (SSID_rec:bitstring, Ka_rec:bitstring, Ks_rec:bitstring) =
        adec(msg1, sk(Kb)) in
    0.

let agentC(Ka:bitstring, Kc:bitstring) =
    in(c, enc_msg:bitstring);
    let (msg1:bitstring, msg2:bitstring) =
        adec(enc_msg, pk(Ka)) in
    let (SSID_rec:bitstring, Ka_rec:bitstring, Ks_rec:bitstring) =
        adec(msg2, sk(Kc)) in
    0.

(* Process *)
process
    ( !agentA(Ka, Kb, Kc, SSID, Ks) | !agentB(Ka, Kb) | !agentC(Ka, Kc) )
```

2.

```
(* Declarations *)
```

```
free c:channel.
free Ka:bitstring.
free Shb:bitstring [private].
free Kb:bitstring.
query attacker(Shb).

fun pk(bitstring):bitstring.
fun sk(bitstring):bitstring.

fun aenc(bitstring, bitstring):bitstring.
fun adec(bitstring, bitstring):bitstring.

(* Protocol *)
let agentB_to_N(Ka:bitstring, Kb:bitstring, Shb:bitstring) =
    let msg1 = aenc(Shb, Kb) in
    let msg2 = aenc(msg1, sk(Kb)) in
    out(c, msg2).

let agentA(Ka:bitstring, Kb:bitstring) =
    in(c, enc_msg:bitstring);
    let msg1 = adec(enc_msg, pk(Ka)) in
    let Shb_rec = adec(msg1, Kb) in
    0.

(* Process *)
process
    ( !agentB_to_N(Ka, Kb, Shb) | !agentA(Ka, Kb) )
```

3.

```
(* Declarations *)
free c:channel.
free SHa:bitstring [private].
free SHb:bitstring [private].
query attacker(SHa).
query attacker(SHb).

fun xor(bitstring, bitstring):bitstring.

(* Protocol *)
let xor_operation(SHa:bitstring, SHb:bitstring) =
    let RH = xor(SHa, SHb) in
    out(c, RH).
```

```
(* Process *)
process
    ( !xor_operation(SHa, SHb) )
```

B.3 Hierarchy promotion

B.3.1 Alice-Bob notation

1. `A : Na XOR Nb .. n XOR Nx = RH`

2. `A → FM : {{RH, Ka}Kfm}Ka-1`

3. `A → B .. N : {Ka, SHb .. n, Kb .. n}Ka-1`

B.3.2 CAS+

Three hierarchy promotion data exchanges, following the three steps in step 2, cluster integrity attestation:

```
protocol HP;
identifiers
A, B, N, FM : user;
Ka, Kb, Kn, Kfm : public_key;
Na, Nb, Nx, RH : number;
SHb, SHn : number;

messages
1. A : calculate_xor(Na, Nb, Nx)
2. A -> FM : {{A, RH, Ka}Kfm}Ka-1
3. A -> B, N : {A, Ka, SHb, SHn, Kb, Kn}Ka-1

knowledge
A : A, B, N, FM, Ka, Kb, Kn, Kfm, Na, Nb, Nx, RH, SHb, SHn;
FM : A, Ka, Kfm;
B : A, Ka, Kb, SHb;
N : A, Ka, Kn, SHn;

session_instances
```

```
[A:alice,B:bob,N:nonce,FM:fieldmarshall,Ka:ka,Kb:kb,Kn:kn,Kfm:kfm,
    Na:na,Nb:nb,Nx:nx,RH:rh,SHb:shb,SHn:shn];

goal
secrecy_of RH [A,B,N,FM];
secrecy_of Ka [A,B,N,FM];
```

B.3.3 ProVerif input

1.

```
(* Declarations *)
free c:channel.
free N:bitstring [private].
free Nb:bitstring [private].
free Nx:bitstring [private].
free RH:bitstring [private].
query attacker(N).
query attacker(Nb).
query attacker(Nx).

fun xor(bitstring, bitstring):bitstring.

(* Protocol *)
let xor_operation(N:bitstring, Nb:bitstring, Nx:bitstring) =
    let RH = xor(xor(N, Nb), Nx) in
    out(c, RH).

(* Process *)
process
    ( !xor_operation(N, Nb, Nx) )
```

2.

```
(* Declarations *)
free c:channel.
free Ka:bitstring.
free Kfm:bitstring.
free RH:bitstring [private].
query attacker(RH).

fun pk(bitstring):bitstring.
fun sk(bitstring):bitstring.

fun aenc(bitstring, bitstring):bitstring.
fun adec(bitstring, bitstring):bitstring.

(* Protocol *)
let agentA(Ka:bitstring, Kfm:bitstring, RH:bitstring) =
    let inner_msg = aenc((RH, Ka), Kfm) in
    out(c, aenc(inner_msg, sk(Ka))).

(* Process *)
process
    ( !agentA(Ka, Kfm, RH) )
```

3.
```
(* Declarations *)
free c:channel.
free Ka:bitstring.
free Shb:bitstring [private].
free Shc:bitstring [private].
free Kb:bitstring.
free Kc:bitstring.
query attacker(Shb).
query attacker(Shc).

fun pk(bitstring):bitstring.
fun sk(bitstring):bitstring.

fun aenc(bitstring, bitstring):bitstring.
fun adec(bitstring, bitstring):bitstring.

(* Protocol *)
let agentA(Ka:bitstring, Kb:bitstring, Kc:bitstring, Shb:bitstring,
    Shc:bitstring) =
    let msg1 = aenc(Ka, sk(Ka)) in
    let msg2 = aenc(Shb, sk(Ka)) in
    let msg3 = aenc(Shc, sk(Ka)) in
    out(c, (msg1, msg2, msg3)).

let agentB(Ka:bitstring, Kb:bitstring) =
    in(c, (enc_msg1:bitstring, enc_msg2:bitstring, enc_msg3:bitstring));
    let Ka_rec = adec(enc_msg1, pk(Ka)) in
    let Shb_rec = adec(enc_msg2, pk(Ka)) in
    0.
```

```
let agentC(Ka:bitstring, Kc:bitstring) =
    in(c, (enc_msg1:bitstring, enc_msg2:bitstring, enc_msg3:bitstring));
    let Ka_rec = adec(enc_msg1, pk(Ka)) in
    let Shc_rec = adec(enc_msg3, pk(Ka)) in
    0.

(* Process *)
process
    ( !agentA(Ka, Kb, Kc, Shb, Shc) | !agentB(Ka, Kb) | !agentC(Ka, Kc) )
```