technologies

MDPI

Article

Towards a Secure and Scalable IoT Infrastructure: A Pilot Deployment for a Smart Water Monitoring System

Anthony Overmars and Sitalakshmi Venkatraman *

Department of Information Technology, Melbourne Polytechnic, Locked Bag 5, Preston 3072, Australia; AnthonyOvermars@melbournepolytechnic.edu.au

* Correspondence: SitaVenkat@melbournepolytechnic.edu.au

Received: 12 July 2020; Accepted: 23 September 2020; Published: 26 September 2020

Abstract: Recent growth in the Internet of Things (IoT) looks promising for realizing a smart environment of the future. However, concerns about the security of IoT devices are escalating as they are inherently constrained by limited resources, heterogeneity, and lack of standard security controls or protocols. Due to their inability to support state-of-the-art secure network protocols and defense mechanisms, standard security solutions are unsuitable for dynamic IoT environments that require large and smart IoT infrastructure deployments. At present, the IoT based smart environment deployments predominantly use cloud-centric approaches to enable continuous and on-demand data exchange that leads to further security and privacy risks. While standard security protocols, such as Virtual Private Networks (VPNs), have been explored for certain IoT environments recently, the implementation models reported have several variations and are not practically scalable for any dynamically scalable IoT deployment. This paper addresses current drawbacks in providing the required flexibility, interoperability, scalability, and low-cost practical viability of a secure IoT infrastructure. We propose an adaptive end-to-end security model that supports the defense requirements for a scalable IoT infrastructure. With low-cost embedded controllers, such as the Raspberry Pi, allowing for the convergence of more sophisticated networking protocols to be embedded at the IoT monitoring interface, we propose a scalable IoT security model integrating both the IoT devices and the controller as one embedded device. Our approach is unique, with a focus on the integration of a security protocol at the embedded interface. In addition, we demonstrate a prototype implementation of our IoT security model for a smart water monitoring system. We believe that our modest first step would instill future research interests in this direction.

Keywords: security model; Internet of Things (IoT); IoT security; Virtual Private Network (VPN); large-scale IoT; smart water monitoring

1. Introduction

An unprecedented recent expansion in the digital environment is attributed to the advanced use of the internet leveraged by smart devices (e.g., sensors, actuators, smartphones, smart appliances, wearables) that have grown rapidly [1,2]. This paradigm shift, termed as the internet of Things (IoT), connects people (end-users) to everything, including systems, machines, and devices with more and more internet-based IoT applications becoming available in everyday life [3,4]. This seamless connectivity allows remote control of devices, as well as data acquisition from various physical sensor devices, to better understand data patterns for intelligent decision-making in both personal and business domains [5]. With exponential growth in IoT devices expected to reach more than 50 billion in 2020, intelligent applications are being developed and deployed for the future realization of smart homes, smart enterprises, smart cities, smart hospitals, smart transport, and other smart public services [6,7]. While the tremendous advancement in the IoT landscape relies on a futuristic forecast of improving public health, finance, business growth, and various other aspects of the environment as a whole, it poses an inherent security challenge that could have an adverse impact on the success of such large-scale distributed IoT systems [8,9].

With the ubiquitous deployment of business-oriented IoT devices and applications, employees are required to work with various IoT devices and stay connected with the corporate network from anywhere and anytime. While data gets relayed via the internet and many public networks, there is a need for establishing secure end-to-end connectivity. Due to several low-cost and resource-constrained lightweight IoT devices being connected over the internet, security measures have not been compatible with the processing power of these embedded controllers in the past [10,11]. The price of embedded controllers, such as the Raspberry Pi, are now becoming capable of running embedded Linux operating systems. This allows for the convergence of more sophisticated networking protocols to be embedded at the IoT monitoring interface, integrating both the IoT devices and the Linux OS controller in the one embedded device at the environmental interface. If IoT devices are connected using the Transport Layer Security (TLS) protocol of the network, then secure connectivity is ensured via an encrypted channel. However, not all devices and IoT platforms can assume the required TLS support. Low-powered IoT devices do not possess the processing power required to encrypt or decrypt data transmissions over the network. [12,13].

There is an increased level of vulnerability due to developers building backdoors into an IoT device and leaving them open [14]. In addition, default passwords provided with the device are seldom changed and become vulnerable to attacks when connected to the internet without appropriate security configurations [15]. This way, an adversary could gain access to the device leading to privacy and security threats to various IoT data and the entire internet. Recent research and practical reviews have reported that information on a significant number of exploited IoT devices has been hosted by Internet Service Providers [16]. With such an openly available known data, all activities made online can be tracked by internet service providers, device manufacturers, and other agencies. Once an IoT device gets hacked, a malicious attacker can affect various other devices in the network by hijacking the routing and forwarding operations of the device. A hacker could exploit such ineffective authentication mechanisms to perform various malicious activities on IoT devices and network communications by appending spoofed malicious nodes resulting in the loss of data integrity. In such situations, any data sent or received via the internet can be read or even altered by spoofing. IoT devices and nodes that store or communicate sensitive data have become the target for attackers. In the absence of effective security mechanisms, IoT devices are highly susceptible to become bots and carry out malicious activities to other devices in the network. Hence, an IoT device connected to the public internet becomes easily susceptible to viruses, malicious programs, and hackers [17]. Critical infrastructures are then susceptible to a cascade of failures by a simple denial of service (DoS) attack on a sensor or actuator [18]. IoT could then become an easy target for man-in-the-middle (MITM) attacks that would result in a totally compromised network system [19]. In addition, due to the increasing use of smart devices for decision-making, data from any maliciously attacked device could become unreliable, which would have a major impact on the decisions made in life-critical environments, such as smart healthcare systems [20,21].

Looking futuristically into the realization of a smart IoT environment, we become more reliant on large automated systems that are based on high-quality IoT data. While IoT enabled devices for large-scale industrial applications to form a competitive edge for businesses, privacy, data breach, and the resulting interruption in the flow of work, as well as network services, due to such DoS attacks are of significant concern [17]. Hence, addressing IoT security needs is of paramount importance. We need a practically viable end-to-end security model that can enforce the principles of security, privacy, and trust throughout the entire IoT system lifecycle, including the design, implementation, and operational phases [22,23]. In addition to the security burden of a Virtual Private Network

(VPN) based deployment, 50 billion devices cannot be supported with IPv4 directly, as well as IPv6 with its additional support requirements that are essential for internet-facing devices [24]. All these requirements and inherent IoT resource limitations must be given due consideration for a successful and secure deployment of any large-scale IoT infrastructure in the real-world. The aim of this paper is to propose a scalable IoT security model to establish a practically viable large IoT infrastructure by uniquely adopting a convergence between the state-of-the-art knowledge surrounding two separate aspects: (i) A popular security protocol, such as VPN, and (ii) the IoT devices. To our best knowledge, this work is the first of its kind, with a focus on the integration of a secure protocol at the embedded interface. In our pilot deployment of the model for a smart water monitoring system, we choose VPN as an example security protocol to demonstrate how this could be achieved. The advantage is that our adaptive end-to-end security model can support any stronger security protocol that may be available in the future. The purpose is to provide a simple method for large IoT deployments where there is a need for a scalable IoT security model to support an extremely low-cost, and self-managed IoT infrastructure.

The VPN technology allows the security of systems to be managed remotely across heterogeneous networks. However, there are several variations in the implementation of VPN, and the most popular state-of-the-art implementation of the security protocol is OpenVPN. As mentioned above, the deployment of VPN protocols at the IoT embedded level was incompatible with the constrained processing capability of the IoT devices in the past [10,11]. OpenVPN is an open-source solution that achieves secure connectivity using SSL (Secure Socket Layer). While OpenVPN has the advantages of strong security and high reliability, it also comes with certain disadvantages, such as high latency depending on the situation, location of access, and distance covered. Hence, the configuring and customizing of OpenVPN needs to be given a unique consideration for different situations. It is also reported in the literature that the VPN fails to comprehend the more advanced and complex threats to the system [24–26]. The scope of this paper does not engage in exploring the limitations of the VPN protocol. Rather, our focus is on the integration of a secure protocol at the embedded interface. In this case, we have chosen VPN as an example, and our proposed IoT security model is highly flexible and scalable that it could be applied with any stronger security protocol that may become available in the future.

Recently, in the IoT context, many ways of implementing VPN technology have been reported, and each research work has been developed with a different context [24–26]. However, the main concerns reported in such studies are lack of adaptability, flexibility, interoperability, scalability, and low-cost viability for deploying a secure large IoT infrastructure. In this paper, we take a modest step to address these existing limitations by proposing an end-to-end security model using a uniquely customized VPN technology that is practically viable for a real-life large IoT environment. We illustrate the application of our model to a unique context of a scalable IoT infrastructure within a smart water monitoring system. Our focus is to address the high overheads, and complex configurations of OpenVPN reported in the literature. Using Raspberry Pi devices and IoT sensors, the aim is to up-shift the security management and control into a single embedded device that has the VPN client software connecting directly to a remote VPN server. Thus, smart mobile apps readily supporting VPN client interfaces will be able to immediately access the remote IoT devices in a secure manner to support remote monitoring, control, and configuration.

The rest of the paper is organized as follows. In Section 2, we highlight the related work, the security and privacy risks reported in recent works, and the key contributions of this research in the given context. Section 3 provides background information on the research design adopted. In Section 4, we identify and describe the key security requirements for the different layers in the IoT architecture. To cater to these essential IoT security requirements, we propose a practical end-to-end security model for a scalable IoT infrastructure in Section 5. The application of our IoT security model to a real-world case scenario of an IoT based water monitoring system is presented, and the implementation details with illustrated outcomes are provided in Section 6. In Section 7, we provide a discussion on the findings and unique contribution of this research work as compared to the current trends in IoT technologies. Finally, we conclude in Section 8, along with future research directions.

2. Related Work and Research Contribution

Literature surveys conducted over the past decade on IoT have identified the need for end-to-end security [9,21,22,25,26]. With an internet connection, a MITM attack has the possibility to gain access and control the IoT networks that could result in hacking several IoT based smart environments [27,28]. IoT technology comes with an inherent trade-off between convenience and control that can affect the critical factors of security and privacy. Attackers tend to scan the internet looking for a specific IoT device vulnerability to steal any personal information that could be misused for eventually resulting in an adverse impact on a large-scale IoT infrastructure [29–31].

Recently, the most emerging communication technology for large-scale IoT infrastructure is Low-Power Wide Area Network (LPWAN), which is a wireless technology that can support large-scale coverage with low bandwidth, long battery life, and long communication distance at a low cost. Among the many competing LPWAN technologies that are predominantly proprietary, LoRa (Long Range), SigFox, and Narrowband-IoT (NB-IoT) are gaining wide acceptance despite having non-standard technical differences [32,33]. However, recent studies in the literature provide details of underlying security mechanisms of each of these LPWAN technologies along with their vulnerabilities and possible attacks.

NB-IoT consists of three layers, perceptron layer, transmission layer, and application layer with complicated network deployment and inherent characteristics of the high capacity battery, and high cost. They work on licensed cellular frequency band, inheriting authentication and encryption of existing cellular infrastructures by mobile operators and have security threats, such as access to high capacity NB-IoT terminals and open network environment [33]. While Sigfox is one of the most secure LPWAN technologies, Sigfox devices predominantly operate offline with a unique symmetrical authentication key given during manufacturing. They may not be well-suited for real-time applications, and the Sigfox application payload is not encrypted [34]. On the contrary, LoRa exhibits open-standards with a unique 128-bit encryption key shared between the end-device and network server, and another unique 128-bit key is shared end-to-end at the application level of a LoRaWAN. Hence, LoRaWAN is the most promising wireless radio access technology that supports long-range communication at low data rates, low power consumption, and end-to-end security using application and network keys. However, LoRa nodes have different levels of vulnerabilities, and compromise of LoRa end-devices by an attacker with physical access, as well as wormhole attacks, are possible using two types of devices that are sniffer and jammer [34,35]. A recently reported security risk analysis of LoRaWAN reveals vulnerabilities against end-device physical capture, rogue gateway, and replay attacks that pose important practical threats [36]. Hence, there is a call for future research directions requiring particular attention by developers and organizations to address relevant security threats while implementing LoRa networks. Overall, various survey-based studies highlighting the vulnerabilities in LPWAN communication technologies have identified an urgent need for secure and uninterrupted communication between an end-device and the gateway for secure and effective IoT networks for large-scale IoT deployments. While there is greater potential in the emergence of software-defined network (SDN) architecture for security in IoT, the protocols in SDN are still under development [37,38].

1. Recent related works have studied the IoT security problem with the main focus of addressing the information leak of different IoT devices in smart environments, such as healthcare medical devices, home/office consumer devices, and educational toy devices [21,39,40]. Other categories of security studies have focused on anomaly detection by monitoring and fingerprinting IoT networks using machine learning techniques, and these solutions are resource-intensive and impractical for large-scale smart environments [41–43]. Further, research studies on secure smart environments are very much focused on specific application domains. One study in the literature proposed a security architecture for smart water management systems that relate to

the real-world case scenario of this research work [44]. However, it ensures secure booting, secure communications, and secure firmware updates of IoT devices in that specific environment. In addition, it adopts cryptographic hash functions that are complex and resource extensive, making such solutions not practically viable for large-scale IoT deployments. Existing security models are complex for resource-constrained IoT and are not generic enough nor dynamically adaptable for a scalable IoT environment. These gaps in existing literature form the main motivation for this research, which is to propose a simple, interoperable, and adaptive security model for large-scale IoT infrastructure.

2. The main goal of this research is to propose a lightweight security model using a simple architecture of VPN suitable for a large-scale IoT deployment. We believe this is an important step in the realm of IoT and Industry 4.0 towards realizing the smart cities of the future. While there are several methods to use VPN in IoT as a common engineering practice, performance and latency are inherent issues with VPN for large-scale deployments in real-world environments [5,24]. Another important aspect to consider in the practical world is its increasing cost and complexity associated with scalability. High administrative time and resources required to manage the network infrastructure could have an impact on the practical viability of a security model. A self-managed IoT infrastructure is warranted for successful adoption in large-scale IoT based smart environments. There is a need for an end-to-end practical solution with an easy-to-use remote device management system that is secure and compatible with the distributed and heterogeneous networks of IoT.
3. IoT devices connected via leading cloud service providers, such as Amazon Web Services (AWS), could be considered as an essential security infrastructure to provide large-scale support for data storage, data processing, and data sharing. However, security challenges posed by each layer of the IoT architecture should be addressed by the cloud service providers to enforce security protocols and privacy standards [45]. The sensor data sent to the edge, fog, and then to the cloud require a network protocol with trusted measures, such as point-to-point encryption, and security certificates. Further, such systems require a paid account with a cloud service provider to have full access to the security certificates, encryption keys, and other resources for achieving cloud-based authorization and authentication mechanisms. A recently proposed model consisting of AWS cloud as master cloud, Raspberry Pi 4 as Edge Node, and Virtual Machines as IoT devices was implemented with an AWS paid account as a proof-of-concept [46]. However, the authors also suggest future studies to be performed on cryptographic security methods that are much more capable of operating on resource-constrained IoT devices (Light Weight Crypto). Further, a replay attack is a major threat towards the cloud infrastructure that raises privacy and security concerns for cloud service adoption for IoT networks [47]. Another recent work proposes a two-factor authentication for IoT security that could restrict unauthorized access to sensitive data communicated by sensors and nodes in an IoT network [48]. Our approach is more suitable for large-scale secure IoT deployments that require an IoT security model to support a simple, extremely low-cost, and self-managed IoT infrastructure.
4. Overall, the main contributions are three-fold: (i) The proposed unique and simple end-to-end IoT security model low-cost leverages off-the-shelf technologies for implementing a large-scale IoT infrastructure, (ii) the practically viable solution has the advantages of an adaptive, interoperable and secure IoT deployment for any smart environment, and (iii) the implementation of our IoT security model within a smart water monitoring system demonstrates its application to any real-world case scenario. The novelty of the proposed solution is in the unique method of integrating the security protocol, such as VPN with the IoT devices, and the controller as one embedded device to establish secure connectivity without having to invest on high-cost proprietary solutions. Our solution also integrates various technologies to provide secure VPN client access to manage, monitor, and control IoT devices in a large-scale smart environment with a user-friendly mobile data analytics capability. This study could instill academic and practical

interest in this dynamically challenging IoT security domain with provisions for future research in studying the solution implementation in various large-scale smart environments.

3. Research Design

In this work, a pragmatic research approach is adopted to explore the security requirements for connecting IoT devices to each other and the internet in addressing the research problem of a scalable IoT security model for any large IoT infrastructure of today. The research design is adopted with the aim to propose a simple, cost-effective end-to-end security model for deploying a scalable and secure smart IoT environment. This section presents the research design, including the epistemological foundation and the rationale in selecting the research methodology for developing a practical security model for a large IoT infrastructure.

In a pragmatic research approach, the focus is more on researching the problem and applying a workable research framework to develop knowledge in finding a solution to the problem [49]. With such a pragmatic lens of "what works", we utilize a qualitative research approach to understand and solve the research problem without touching on any aspect of quantitative research philosophy [50]. This research study aims at developing a security model for cost-effective deployment of a smart IoT environment to seamlessly connect, control, and managing several low-cost IoT devices via the internet. Hence, an interpretive epistemological approach of qualitative research methodology would be applied for achieving this objective as it is suitable for an exploration of the typical security requirements that are warranted within the IoT context of a real-world smart environment [51,52]. We adopt a case study methodology within our workable qualitative research design that aligns well with our research aim. The basic guidelines from the literature [53,54], as summarized below, are adopted to ensure the quality of our research framework:

(a) Research philosophical consideration—we consider an interpretive epistemology as the choice of the research philosophical paradigm [55–57]. We identify the IoT security viewpoints based on literature by identifying the inherent vulnerabilities in each of the four basic layers of the IoT architecture (presented in Section 4). These viewpoints serve as theoretical and practical knowledge forming the basis for proposing an effective solution for the research problem.

(b) inquiry technique consideration—we adopt an inquiry technique that is qualitative in nature employing descriptive data that is interpretive in nature [58]. We propose a practically viable end-to-end lightweight security model through developing network security reference architectures, which is typically design-oriented research that aims at solving the IoT security problem (presented in Section 5). Similar to other IoT related qualitative studies reported in the literature [59,60], we describe the proposed IoT security model with an interpretive approach and establish the credibility, conformability, transferability, and dependability of the solution through practical solution deployment.

(c) research logic consideration—we adopt an abduction logic to infer the application of the proposed secure IoT infrastructure within a single case setting using well-established guidelines [61]. For illustrating a practical use of the proposed secure IoT infrastructure, we include a working prototype in a real-world smart environment. Data analytics and visualization of the data collected via a secure and smart water monitoring system is demonstrated for the research logic consideration in the case scenario (presented in Section 6).

5. Overall, the research contribution is the development of an adaptive end-to-end security model for large-scale IoT infrastructure with essential features of simplicity and scalability. Further, in this study, the pilot deployment of our IoT security model in a real-world case scenario of a smart water monitoring system serves as a starting point for "model testing" within our deductive research journey. In future research, the IoT security model will be applied to other

smart environments as part of an inductive research study. Such an approach of our research design would facilitate to iteratively finetune and evolve with a generalized end-to-end security model that would become applicable for any large-scale IoT deployment.

4. Security Requirements of IoT Architecture

A typical IoT ecosystem consists of sensors, actuators, a processing unit with firmware that operates with constrained resources, and wireless communication infrastructure to receive the sensed data and send them to any location via the IoT gateway and the internet [60,62]. IoT devices are embedded into larger real-world applications that are emerging towards establishing a smart environment with a paramount emphasis on precision and intelligence [63,64]. Innovative IoT applications are being witnessed in healthcare systems, weather forecasting, agriculture monitoring, traffic management, and in many more domains for realizing smart homes and smart cities of the future [21,39,65]. However, in such a heterogeneous operating environment, the IoT network with constrained resources is faced with significant security and privacy challenges. IoT devices with highly primitive security features are susceptible to attacks as they become entry points to infiltrate into critical infrastructures via the connected networks [38,41]. There is an escalation of new IoT threats and security risks, due to the inherent vulnerabilities in each of the four basic layers of the IoT architecture:

(a) Device or Perception Layer;
(b) Network or Transmission Layer;
(c) Middleware or Service Layer;
(d) Application or Business Layer.

In this section, we identify the potential risks and the key security requirements in each IoT layer from reported studies to form the key security requirements for our research problem [29,42,66,67]. We summaries our findings of IoT vulnerabilities and security risks in each layer of IoT architecture below.

(a) Device or Perception Layer

The Device or Perception Layer works with two of the IoT components [68,69]:

(i) sensors that sense data pertaining to human and environment parameters, such as temperature, humidity, motion, location, etc.;
(ii) actuators that control the physical device, such as air conditioner, vehicle transport, irrigation pump, pacemaker, etc.

This layer not only assists in identifying various device sensors and actuators, but also monitors them and takes necessary action for further data processing and data routing to the Network Layer. The low-cost and low-speed wireless personal area network (WPAN) protocol of this layer requires communication via IoT gateway to transmit enormous amounts of sensed data to the cloud storage. Attacks are possible to jam the communication between the device and IoT gateway (jamming attacks) by exploiting the frequency used in WPAN. An adversary having access to the device could tamper the device, including the firmware, by injecting malicious code. Such code injection attacks could physically damage a specific device or even compromise the entire IoT communication network [41,70]. There is a need for core security functionality, such as:

(i) Authentication—verifies the provenance of IoT devices,
(ii) Authorization—allows only valid users to access the device and services,
(iii) Integrity—ensures unauthorized users do not modify the device firmware or data, and
(iv) Confidentiality—enforces privacy in locating the IoT device and the data transmitted via the network.

(b) Network or Transmission Layer

The Network or Transmission Layer manages the device communication in the IoT infrastructure using the nodes, gateways, and the firmware [71]. Device data could be transferred using wired or wireless transmission technologies, such as 6LowPan, Bluetooth, or Zigbee [72]. Due to the limited processing and power energy resources of IoT devices and Wireless Sensor Nodes (WSN), an adversary gaining access to the nodes/gateways could launch MITM attacks, spoofing, and distributed denial of service (DDoS) attacks [73–75]. User and device credentials could be stolen resulting in physically compromised nodes/gateways while the device is in sleep mode. This could further lead to code injection, where attackers could take control of the IoT network infrastructure and even the entire network domain. Practical security solutions are required to cater to the heterogeneity of IoT network infrastructure and to support lightweight features using edge-intelligence and decentralized management.

(c) Middleware or Service Layer

The layer that bridges between the Network Layer and the Application Layer is the Middleware *or Service Layer*. This layer is responsible for processing the data for each vendor-specific service of various IoT devices. It deals with the pre-processing of IoT data for different third-party applications. It makes use of machine learning and intelligent data mining, for facilitating automatic actions with real-time response requirements in critical environments, such as traffic or health care systems [62,63]. Hence, further data processing required in the Application Layer depends on the security and trust of the Middleware Layer for enforcing the integrity of IoT data [23]. The level of security very much depends on third-party application platforms. With the IoT data predominantly stored in the cloud servers, the IoT infrastructure is posed with various malicious attacks and threats. Unauthorized access to open ports of services and other backdoors could be used by malicious attackers to affect the security of the IoT infrastructure. Hence, the IoT security requirements should include good identity management to support the integration of various services across different devices, users, and different platforms, including cloud servers [76]. In addition, the security architecture should support the scalability of the IoT infrastructure to interoperate with new middleware applications and services [77].

(d) Application or Business Layer

The topmost layer of the IoT architecture is the Application or Business Layer, which has the role in processing the transmitted data further using machine learning and other intelligent models to result in smart IoT device actions. Applications in this layer include third-party Apps, websites, portals, and other smart software solutions for various enterprises with different suitable business models. The User Datagram Protocol (UDP) is one of the core IoT protocols. Though web infrastructure is available for IoT devices, internet-specific protocols, such as TCP, come with overheads and are not suitable for most IoT applications [78]. Other lightweight protocols, such as CoAP and MQTT-SN, for sensor networks, are designed to use UDP [79]. IoT supports many more protocols than the web, which are yet to demonstrate reliability and standards. Hence, scripting attacks are possible through application-based control of IoT devices via mobile Apps. Much similar to web application layer vulnerabilities, phishing, and buffer overflow attacks are possible in the IoT infrastructure. In addition, side-channel attacks capitalize on constrained resources of IoT, such as shorter encryption keys and power consumption analysis of IoT devices.

Overall, a set of key security requirements for IoT communication through the various layers of IoT architecture are (i) interoperability for traversing through different domains that support varied security technologies; (ii) simple, lightweight end-to-end security; (iii) highly-flexible security model to cater to various changes in the IoT infrastructure, due to the dynamically joining and leaving of IoT devices, users, services and applications; and (iv) low-cost and practically viable solution for any large-scale IoT deployment.

5. Proposed Security Model for a Scalable IoT Infrastructure

The security requirement based on the four layers of IoT architecture discussed above highlights that an IoT device needs to support the TCP/IP protocol stack, as well as some environmental support function (a switch, sensor, or actuator). The processing capabilities of resources constrained IoT devices to support such security protocols, including the well-accepted VPN or IPv6, have not been practically viable for a large IoT infrastructure deployment where low-cost is the dominating attribute. Technological developments with devices, such as Raspberry Pi, to support sensor and actuator management at the local level have shown promise in accommodating the required security protocols [80,81]. However, they were originally designed to be more expensive and were not readily viable with end-to-end security requirements for large-scale deployments. More recently, advanced versions of such devices (Raspberry Pi4) along with cloud services for supporting the essential security required in large-scale IoT infrastructure were explored [46,47]. However, privacy threats, security attacks, and risks of multi-tenant cloud platforms form gaps in the literature. In addition, the technological viability of currently available low-cost embedded controllers, such as the Raspberry Pi, form the key motivation for our novelty to propose the integration of a secure protocol at the embedded interface for a scalable IoT security model. The aim is to develop the convergence of more sophisticated networking protocols to be embedded at the IoT monitoring interface by integrating both the IoT devices and the controller as one embedded device that would minimize privacy and security risks.

We propose a security model to off-load the security functions, such as VPN and IPv6, the protocol to an internet-facing device, and cluster the IoT sensory environment behind a firewall using Network Address Translation (NAT) to access the IoT using IPv4. However, there are many VPN-based security solutions reported in the literature as each development model varies with the application environment and is not a simple and straightforward solution [10,11,24]. Recently, even in the IoT context, many different ways of implementing the VPN technology are reported [25,26]. Each research work has been developed with a different real-world context, and our aim is to propose a simple, low-cost, end-to-end IoT security model that can be easily applied to any context of a self-managed scalable IoT infrastructure, such as a smart water monitoring system.

In our proposed IoT security model, we consider OpenVPN as the VPN technology for the integration of the secure protocol at the embedded interface. Here, we describe in detail the development of our security model based on our ongoing research with VPN technology developments and how integrating both the IoT devices and the controller as one embedded device can be achieved. An OpenVPN client running on a Raspberry Pi 4 can be deployed to do the forward internet-facing using IPv6, and the IoT devices sitting behind the firewall can be port forwarded to appear on the internet using NAT and port forwarding. In this way, the IoT devices appear on the internet, but can only be accessible to other devices on the VPN server. An OpenVPN server can also be deployed on a Raspberry Pi, and these issues IPv4 addresses to the VPN clients, behind which numerous IoT devices may reside. The VPN server may reside behind a firewall on another network. However, its IP address is known to the VPN clients. A possible configuration that we have deployed is to have the VPN server behind a firewall, and it is port forwarded via network address translation such that it appears on the network. The internet forward-facing router IP address must be known to the VPN clients. This can be achieved using a static IP address on the network if the ISP allows this or if there is a dynamic IP address to which a label is applied, and this label is registered with a DNS service. The authors used a service from a dynamic DNS (DynDNS) to achieve this. The VPN clients then establish a VPN connection with the Label of the VPN Server. The VPN Server establishes the VPN connection and issues valid IP addresses to the VPN clients. Any devices on the same VPN network can now communicate with each other. Low cost, off the shelf routers which support OpenVPN in both Client and Server modes are readily available, and our deployment was done using ASUS RT-AC66U. This has a VPN configuration interface which can be configured as a Server or a Client. Devices, such as Raspberry Pi, can be attached via the RT-AC66 USB ports or through NAT on the wired LAN ports (4) or the Wi-Fi network.

To achieve an end-to-end security model, we consider a Session Initiation Protocol (SIP) based VoIP adapter at a remote location, and include the VoIP adapter into the "VPN Client-Side Device" on one end with “VPN Server-Side Device” on the other end. Adapting from the OpenVPN standard protocols [82], we establish two connections to the router: (i) To receive the “tunneled” data, and (ii) to send the unencrypted data back onto the local network from the VoIP adapter. Several different IoT devices could be connected on the Client-Side in this manner. In earlier work, the authors provisioned secure VoIP using UDP packet protocols (Patented) [83]. The UDP protocol affords some packet loss and does not provide acknowledgment of packets received. Whilst this is satisfactory for VoIP implementations, the UDP protocol is not well suited to IoT implementations where packet loss may lead to a loss of monitoring and/or control messaging. In this paper, we advance further by using the TCP/IP protocol for deploying a secure IoT infrastructure.

A common practice for enterprises of today is to make use of one of the two deployment models to reduce MITM attacks as given below:

(i) On-premise networks isolating their systems to enforce utmost security;

(ii) External VPN providers to create secure encrypted tunnels rather than public networks. However, both these deployment approaches exhibit disadvantages of performance, latency, and high complexity and cost with large scale IoT implementation and configuration in real-life. In order to incorporate the security features of a VPN with seamless configuration and deployment, we do not use an external provider for a VPN. Our implementation involves a VPN server behind a firewall, which is port forwarded to a NAT address behind the firewall. By running this as a dynamically assigned IPV6 internet address using DYNDNS servers, it allows the provision of a dynamically assigned IPv6 internet-facing address. This provides a robust security model for remotely configuring, controlling, and self-managing IoT devices over an encrypted end-to-end connection.

In 2008, an irrigation system was developed, which could be operated over the internet with end-to-end security [83]. This was prior to the IoT becoming mainstream. The system involved several ZigBee devices controlled from a central ZigBee master that was connected to a 3G router. Access to the remote system was established using a simple Windows XP remote desktop session via port 3389. However, many security exploits were developed for port 3389, and the system though robust and reliable, was vulnerable to unauthorized access. This system was further extended by Overmars in 2009 [84]. We develop the concept further by keeping in mind the security requirements essential in the recent IoT landscape towards the realization of smart homes and cities.

The vulnerabilities of IoT devices are because they are generally small microcontrollers that are not able to run the full TCP/IP protocol stack. Moreover, since the current and future IoT infrastructure is likely to have billions of devices, the media access control (MAC) layer will be required to implement TCP/IP v6 addresses. This additional processor burden is generally well beyond the capacity of most 8-bit or 16-bit processors of IoT devices. The architecture outlined in [85] proposed that the security and the TCP/IP protocol stack should be off-loaded to a mobile router with a 3G data interface and that all the peripheral devices, now known as IoT devices, are required to be network address translation (NAT) via a TCP/IP v4 address range or via a Bluetooth connection. These network stacks could be optimized to provide the very minimum of interface processing. This would allow more central processing unit (CPU) capability for the device’s environment monitoring and system control, and thereby reduce the overall system cost.

The deployment of our proposed model allows the OpenVPN clients to be either fixed or mobile, while the OpenVPN server is fixed in one location and can be provisioned with an uninterruptible power supply. OpenVPN clients can support 252 IoT hubs limited by the NAT protocol. Each of these hubs supports a single IoT device mapped on each of the I/O ports. According to RFC 793, the port range is 0–65,535. A registered port is one assigned by the internet corporation for assigned names and numbers (ICANN) to a certain use. Each registered port is in the range 1024–49,151. Therefore, about 48,000 IoT devices can be assigned to one IPv4 address. The available IPv4 subnet addresses are assigned to each of the OpenVPN clients (of which there can be up to 252). Further, each of these clients can support up to 48,000 IoT devices. The practical limitations are shown in Table 1.

Table 1. VPN server-client configuration for a secure, scalable IoT deployment of the case study.

Device	VPN	IP Address
Gateway #1	**Server #1**	**192.168.0.1**
Mobile Phone Monitor #1	Client	192.168.0.2
Water Tank #0.1	Client	192.168.0.3
Water Tank #0.N	Client	192.168.0.x
Water Tank #0.250	Client	192.168.0.251
Gateway #2	**Server #2**	**192.168.1.1**
Mobile Phone Monitor #2	Client	192.168.1.2
Water Tank #1.1	Client	192.168.1.3
Water Tank #1.N	Client	192.168.1.x
Water Tank #1.250	Client	192.168.1.251
Gateway #10	**Server #10**	**192.168.9.1**
Mobile Phone Monitor #10	Client	192.168.9.2
Water Tank #9.1	Client	192.168.9.3
Water Tank #9.N	Client	192.168.9.x
Water Tank #9.250	Client	192.168.9.251

Our proposed model recognizes that the mobile phone platform is well suited in providing the necessary interface hardware (3G, 4G, 5G) adaptively for upstream internet data connection, whilst acting as a local gateway providing NAT to the locally distributed IoT devices. These local devices could then be connected via Wi-Fi or Bluetooth or ZigBee. Further, mobile phones with the OctaCores are now operating with CPU speeds of more than 2GHz and can provide relatively inexpensive "gateways".

With the massive advancements in wireless technologies, mobile phones are now capable of offering secure client sessions to remote servers via VPN or IPSec tunnels, using the always open port 500 on all network routers and switches. Further, these secure upstream client sessions could be simply integrated into the phones' operating system or the customizable solutions that are available via their respective App stores. More recently, configuring the phone to be a downstream hotspot providing Wi-Fi and Bluetooth is also becoming rudimentary and part of both Android and Apple operating systems. The upstream server infrastructure, which provides the IPSec and VPN services, are also off-the-shelf, and many Linux offerings are both secure and open-source. A patented work [86] also showed that open-source routers, such as OpenWRT, could easily be reconfigured as both VPN/IPSec servers or clients and offer a diverse number of IoT device configurations, via Wi-Fi TCP/IPv4 or USB with ZigBee or Bluetooth adaptors. These open-source routers are well suited to a multitude of tasks. Performing patching and updates in device drivers for many I/O devices is nowadays automatic. Further, we adapted the secure IoT model to remote farm locations with applications in solar power/water distribution, as well as to mobile vehicular environments.

In this research, we visualize our proposed a scalable IoT security model deployable on-farm infrastructure for a water tank monitoring system, as shown in Figure 1. The on-farm IoT Raspberry Pi device manages both the environment (the water tank) and creates the VPN tunnel to the VPN server with the integrated VPN client. The Raspberry Pi also has a Wi-Fi link that connects directly to the Access Point (AP). The AP then connects to the internet. The VPN tunnel created by the Raspberry Pi, bypasses the AP through port 500 (always open) and connects directly to the VPN server, which exists in a remote location.

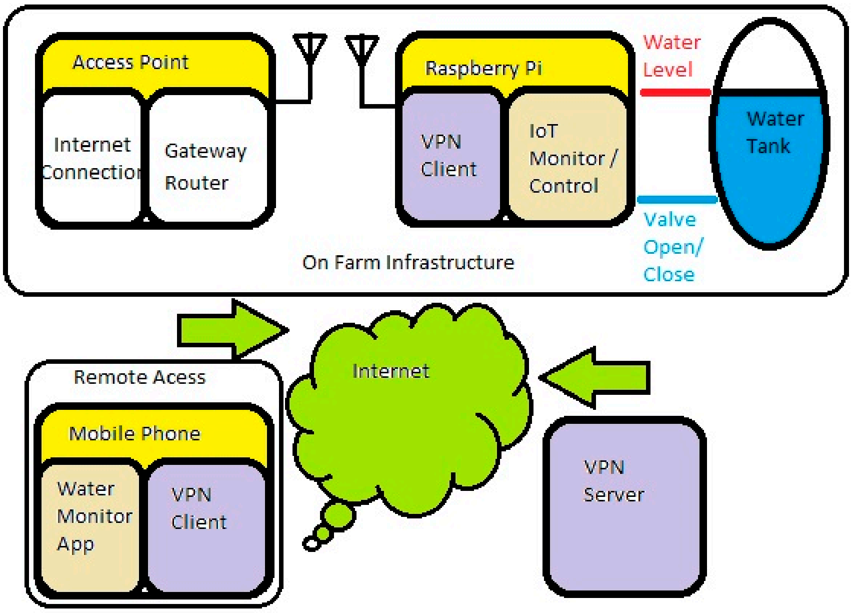


Figure 1. Proposed a scalable security model for a large Internet of Things (IoT) infrastructure. VPN, Virtual Private Network.

We walk-through how our proposed solution establishes end-to-end secure connectivity among all the entities, such as mobile App, VPN client, VPN server, AP, gateway, and IoT devices, using Raspberry Pi, as follows. Remote access to the Raspberry Pi is achieved securely through an App running on a mobile phone. This is achieved by the mobile device first establishing a secure connection to the VPN server using a VPN client on the mobile device. Once a VPN secure connection is established between the VPN server, and the VPN client on the mobile device, the VPN server issues an IP address to the mobile phone as if it was on the same subnet as the Raspberry Pi. The application on the mobile phone can now access the IoT with the end-to-end security established for monitoring and controlling the application running on the Raspberry Pi. Both the Raspberry Pi and the mobile device are now connected on the same subnet via the VPN server, and the VPN server is the gateway controller issuing NAT addresses to both the Raspberry Pi and the mobile device. The VPN server appears as 192.168.1.1, and the Raspberry Pi and the mobile device appear as 192.168.1.2 and 192.168.1.3, respectively.

In summary, our proposed model is a low-cost end-to-end secure deployment for any large IoT infrastructure. The main contribution is our unique deployment approach to consider scalability and easy-to-use implementation as key factors for its practical viability. In our solution, ASUS routers are configured as VPN servers and clients to establish a VPN network. On the other hand, other state-of-the-art solutions are implemented through higher cost VPN servers, such as CISCO, devices. Further, the advantage of our cost-effective deployment model is that our implementation in the large-scale real-world applications can use low-cost devices, such as Raspberry Pi 4 devices, which allow for VPN servers and clients using OpenVPN running in a Linux Kernel. The integration of OpenVPN with the environmental controller allows for IoT devices to offer VPN client access, as well as to deploy their environmental control functionalities. The OpenVPN Server provides the downstream security by provisioning OpenVPN clients with an internal IP address, which then provides distributed IoT devices with a secure method of interconnection between each other over a virtually secure private network in accordance with the VPN standard. Each client can then communicate with each of the other clients on the network. Each client router supports several IoT devices as peripheral devices on each of the client's subnets. In this way, IoT devices pertaining to a client's subnet can be interrogated and/or manipulated in their respective control environments.

Next, we describe the implementation of our proposed IoT security model in real-world applications with a case scenario as an illustration.

6. IoT Security Model Deployment—A Case Scenario of Smart Water Monitoring System

In this section, we describe a case scenario as an illustration for deploying a secure IoT infrastructure using our proposed model to monitor and control remote water tanks in a smart "on-farm" environment. We consider a farm that harnesses rainwater in addition to regular town water supply for the case study. Since the level of water in each tank can change dynamically based on the amount of rainfall, monitoring the level of water for efficient use and distribution in the farm. Depleted tank levels occur when the utilization rate exceeds the resupply. Alternatives to town water or scheduled trucked in water could be better managed with rainfall measurement and prediction. Remote monitoring of these resources using the level measurement of each tank via a user-friendly mobile device, facilitates decisions for an optimal and economic rainwater /town water resource balance.

We developed a prototype water tank with a water level monitoring device using a simple conduction sensor, as shown in Figure 2. We applied our proposed IoT security model to the case scenario to implement smart water monitoring and management with end-to-end security. For our proof-of-concept pilot implementation, two tanks were monitored, a household greywater system (2000 L), and an on-farm tank in another location (45,000 L). These were fitted with ultrasonic sensors, valve solenoids, and pressurizing pumps. Figure 3 shows an illustration of the deployment of our end-to-end security model using OpenVPN with one of the water tanks. The levels of the individual tanks can then be checked remotely, and refilling can be automated based on parameters, such as time of day, low-level minimum values, or by remote manual intervention. An ultrasonic tank level sensor with a Sentryrobotic Wi-Fi transmitter is adopted for this case scenario based on the SMART water tank monitor system [87] and the pi-tank-watcher [88].

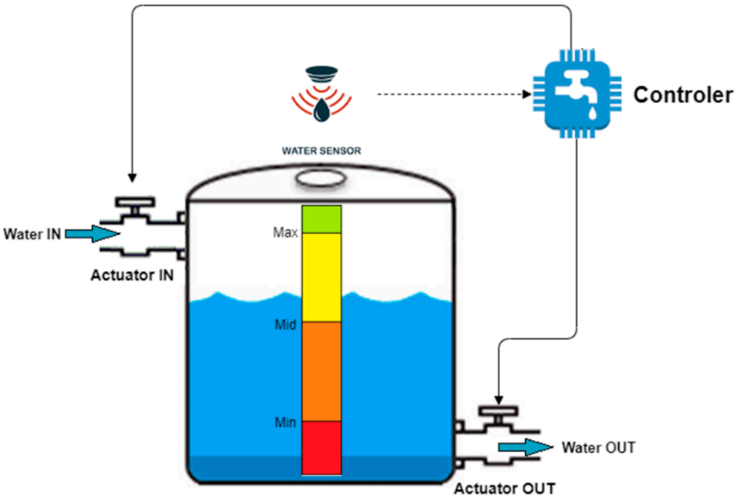


Figure 2. A prototype water tank with sensors to monitor and control the water level.

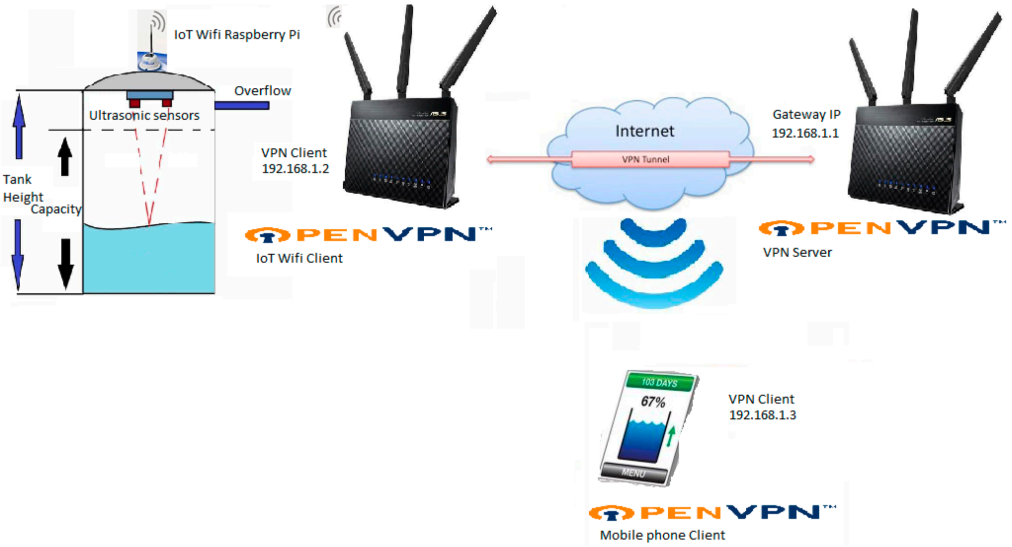
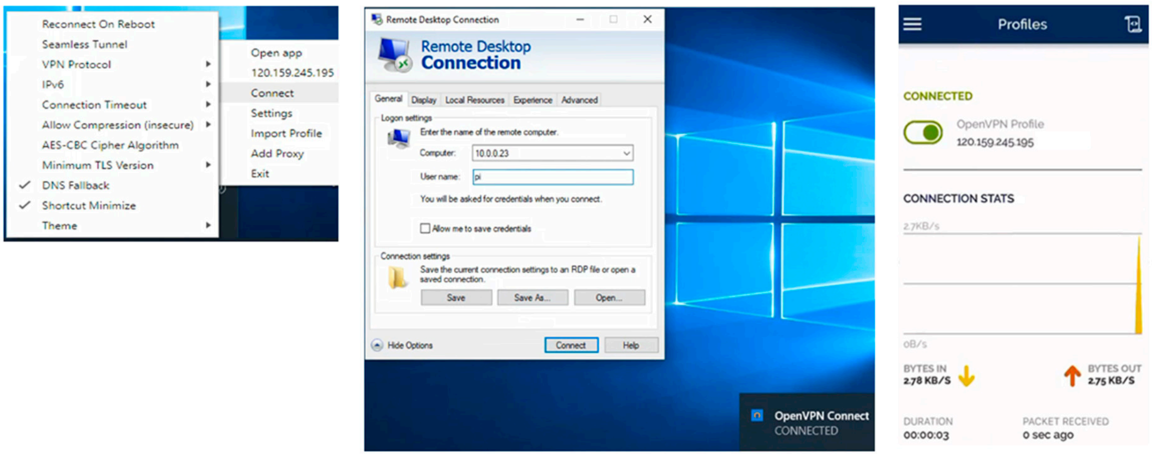


Figure 3. Pilot deployment of the proposed IoT security model for a smart water tank prototype.

Table 1 provides a typical set of IP address configuration using our proposed end-to-end IoT security model accommodating up to 10 VPN servers and 250 water tanks per server, facilitating a scalable and larger deployment of a total of 2500 water tanks. This implementation is deployable on a per farm basis. It is not intended to be for a city-based water board, though it may be sufficient for a small municipality.

Implementation of Our Proposed IoT Security Model

Our proposed IoT security model ensures that the security measures are first enforced with OpenVPN optimization, and tuning before deploying the IoT enabled devices for the smart water monitoring system. We provide details on how the OpenVPN connecting the IoT enabled nodes establishing the end-to-end security protocols are implemented. Figure 4 demonstrates the authenticated OpenVPN connection established using a simple user-interface. The Open VPN server creates the OVPN script, a script file with extension .ovpn, which is shared securely with the OpenVPN clients. This is used by the clients to establish a secure connection to the server. Figure 5 provides an illustration of running the OVPN script for generating Rivest–Shamir–Adleman (RSA) keys, and Figure 6 shows the creation of a VPN certificate using SSL security protocol successfully. Once a secure connection is established, the server issues a Dynamic IP address using Network Address Translation (NAT) protocols. Once a NAT address has been issued to the client, the client is free to communicate with all other clients in the VPN network. Currently, there are two types of clients in the VPN network. One has the IoT devices associated with it in an integrated Raspberry Pi acting as a discrete element. The other device is the remote monitor, which is implemented on a mobile phone. Further, we adopt the authentication method for the nodes with the admin having read and write access, while other users are limited to read access only.



Enter the name of the remote computer.

Computer: 10.0.0.23

User name: pi

You will be asked for credentials when you connect.

□ Allow me to save credentials

Save the current connection settings to an RDP file or open a saved connection.

OpenVPN Profile
120.159.245.195

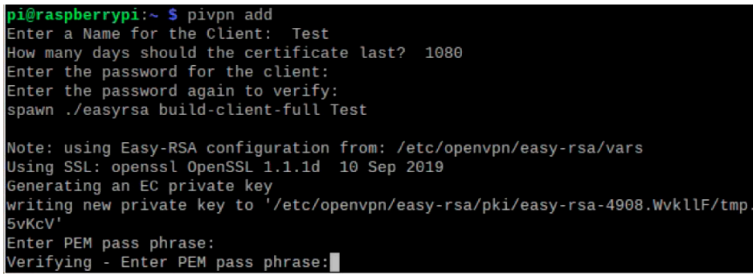
BYTES IN 2.78 KB/S

BYTES OUT 2.75 KB/S

DURATION 00:00:03

PACKET RECEIVED 0 sec ago

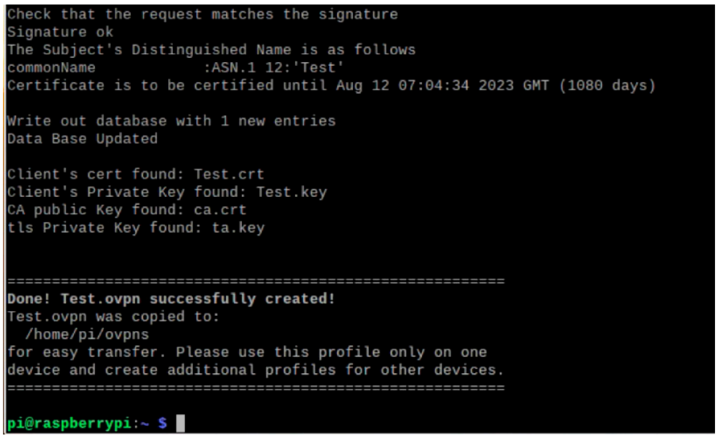
Figure 4. User-interface for the authenticated open VPN connection.



```
pi@raspberrypi:~ $ pivpn add
Enter a Name for the Client:  Test
How many days should the certificate last?  1080
Enter the password for the client:
Enter the password again to verify:
spawn ./easyrsa build-client-full Test

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa/vars
Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019
Generating an EC private key
writing new private key to '/etc/openvpn/easy-rsa/pki/easy-rsa-4908.WvkllF/tmp.
5vKcV'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Figure 5. Illustration of OVPN script running the RSA configuration.



```
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'Test'
Certificate is to be certified until Aug 12 07:04:34 2023 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Client's cert found: Test.crt
Client's Private Key found: Test.key
CA public Key found: ca.crt
tls Private Key found: ta.key


========================================================
Done! Test.ovpn successfully created!
Test.ovpn was copied to:
  /home/pi/ovpns
for easy transfer. Please use this profile only on one
device and create additional profiles for other devices.
========================================================

pi@raspberrypi:~ $
```

Figure 6. Creation of a VPN certificate using the Secure Socket Layer (SSL) security protocol.

For our smart water monitoring system case scenario, the water level in each tank is measured periodically based on the water depth reading of the sensor, and its rate of outflow determines the valve opening rate. The pressure of the mains is likely to vary, and the rate of filling versus the rate of outflow determines the valve is opening duration. The controller makes the decision of how long the valve should be kept open based upon the rate of refilling. The tank sensing and filling are on one sub-system, and the decision control is separate. These sub-systems are on different networks. The decision control and monitoring are performed using a mobile app. A secure connection using our proposed security model is established among the IoT devices, such as the tank sensor, tank valve, and the controller. Figure 7 shows a prototype of Raspberry Pi and the water sensor connected with a breadboard for our pilot implementation and testing.

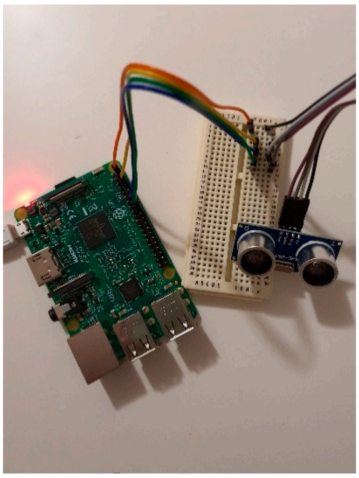


Figure 7. Prototype of Raspberry Pi and the water sensor connected with a breadboard.

A process flow diagram for the operation of water sensors and valves is given in Figure 8. The "Calculate Percentage" node calculates the percentage of water level based on the data from the water sensor, and the result is transferred to the "Water Level" dashboard node to display the output on the mobile app dashboard for the monitor and control of water level remotely. An illustration of the output is shown in the dashboard is given in Figure 9. To perform an auto refill of water with "Valve l", a rule is set, such as "if the water level is less than 15% of tank capacity, turn ON the valve; if the water level is greater than 80%, turn OFF the valve". Valve 1 is then connected to a valve switch control "Water In" node that triggers the action accordingly with the status, "Statute" which is set to communicate "Water is refilling" if Valve 1 is ON, or "Water is ready to use" if Valve 1 is OFF.

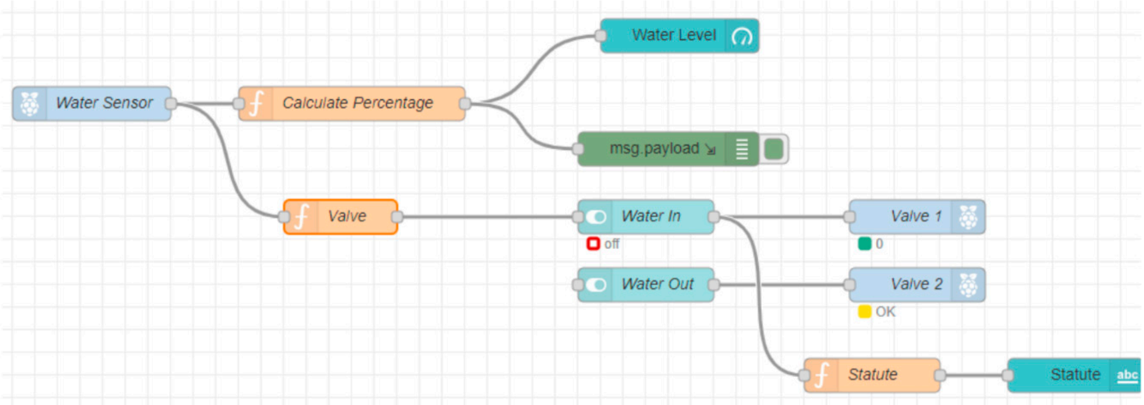


Figure 8. Water monitoring system process flow diagram.

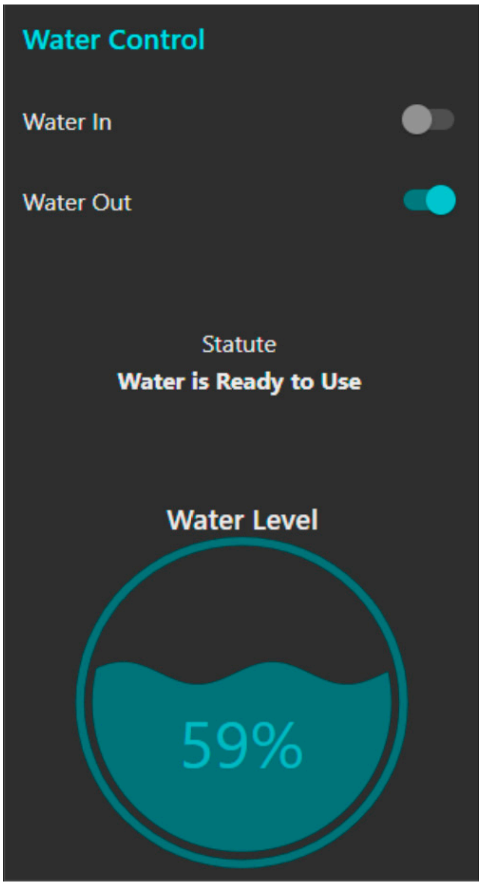


Figure 9. Smart mobile app dashboard to monitor and control water level remotely.

Further, as shown in Figure 10, dashboards for Raspberry Pi mobile devices are designed to monitor resource overheads and utilization, such as memory and CPU load, including the CPU temperature. We considered the design of such a dashboard for future power consumption minimization and optimization as it is intended that these devices would eventually be running from solar power resources. To illustrate the monitoring of water level trends over a longer time interval, we provide from publicly available resources [87,88], the outputs of data analytics using software tools in Figure 11. Such graphical trends would provide data insights for making an informed decision for remotely operating the water tank sensors with a user-friendly mobile App. In addition, using a cross-reference against weather data, many predictive models could be employed to make decisions on the usage of water. For instance, when the water level drops, adjustments to water consumption could be programmed to water the farm appropriately. Similarly, data from weather forecasts and rain patterns could be correlated with the water tank data. For instance, the correlation between the water tank level and the weather condition could be determined. With such data analytics, more informed and intelligent decisions could be made for both water storage and water usage. Various trends on water inflow and outflow of rainwater and town water tanks could provide useful data insights to identify correlations among the control parameters.

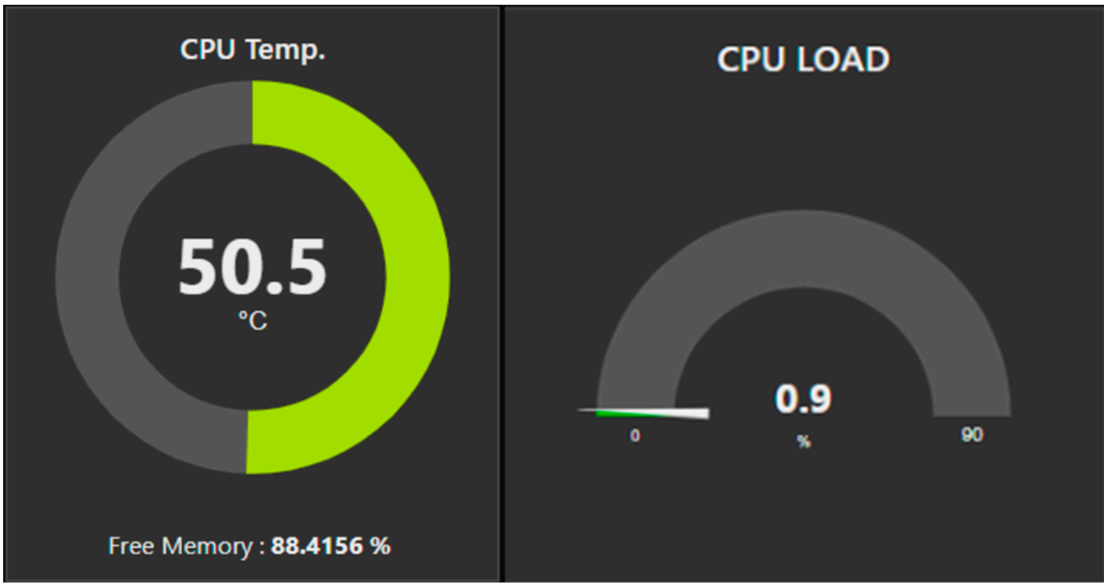


Figure 10. Sample dashboards to monitor resource utilization and overheads.

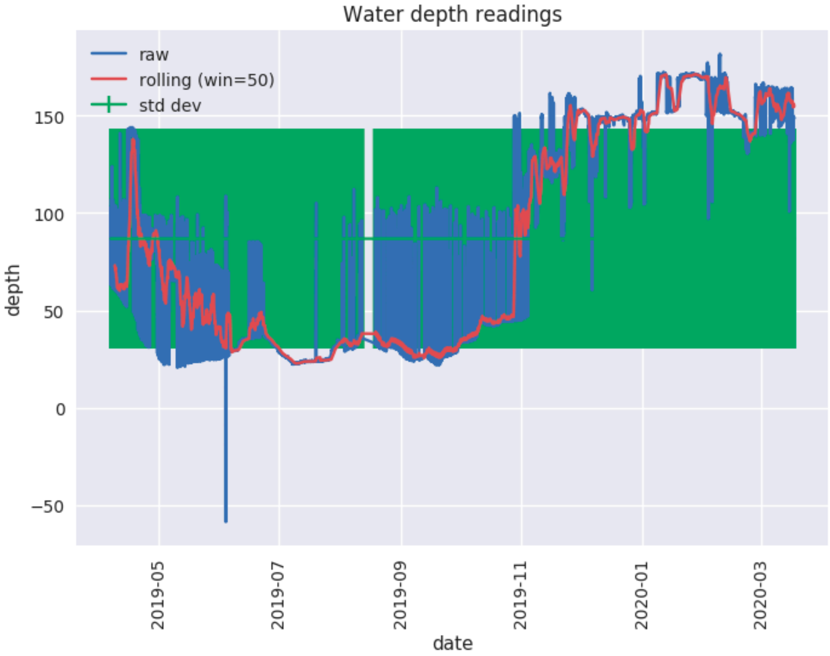


Figure 11. Sample trend in water tank level readings [88].

Our case scenario using a smart water monitoring system mainly illustrates the application of our proposed security model for a scalable IoT deployment as a case study. Any security breach resulting in MITM attacks in such a scenario can affect the integrity of the water readings of the tanks. The attacker could misuse the automated controls leading to disastrous outcomes for the farm. The focus of this paper is not towards addressing the limitations of the VPN protocol, but mainly on the proposal of a novel method to integrate a secure protocol at the embedded controllers, such as Raspberry Pi devices. For this case scenario, we have implemented our proposed IoT security model using VPN as an illustration. Our model is highly flexible and scalable than any security protocol could be applied in the future. Thus, in this paper, we establish the value cocreation process using a smart water tank monitoring case scenario to illustrate the practical application of our proposed IoT security model. Further, our proposed practical and self-managed IoT security model paves the way for future empirical studies for large-scale secure IoT deployments in various other smart environments

7. Discussion and Current Trends

In the real-world, an industrial smart IoT deployment solution requires high levels of scalability to support a large number of heterogeneous entities within a dynamically changing IoT ecosystem. In addition, due to their information exchange among different multiple systems and technologies, current IoT technologies most often use mediators or translators via the cloud that are posing more security and privacy risks. A security breach in a smart IoT environment can result in damage to the information assets, people, and infrastructure—leading to huge financial loss [6,7,20].

Recently, LPWAN poses to be the fast-growing communication technology for IoT, as discussed in Section 2. There are several competing standards and vendors, such as LoRaWAN, NB-IoT, and Sigfox, which allow thousands or millions of sensors to be integrated into an application [26,88]. For instance, DASH7 is a low latency, bi-directional firmware standard that operates over multiple LPWAN radio technologies, including LoRa (Long Range), a proprietary, chirp spread spectrum (CSS) radio modulation technology. Ultra-Narrowband (UNB) is a modulation technology used for LPWAN by various companies, including Sigfox, for specific situations. These are some of the many competing proprietary standards and are not interoperable with all types of IoT devices that are being manufactured every day. Furthermore, recent surveys and research studies comparing such LPWAN technologies have reported various security infiltrations and vulnerabilities [34–36].

An IoT irrigation system was implemented on the 3G network more than a decade back, forming a patent [84]. The 3G IoT irrigation network implemented in the mid-2000s was not secure with the introduction of new malicious network attacks. Recent work considers NB-IoT to be the standard for large-scale IoT deployments [31]. However, as discussed in Section 2, there are security and privacy risks in adopting LPWAN, as well as cloud platforms, particularly for large-scale IoT deployments, such as the smart water monitoring system considered in this study [33,36,45,47]. Our emphasis is on the provision of a low-cost and secure IoT infrastructure that can be self-managed with the least cost, overheads, and complexity.

This paper has demonstrated how a secure IoT network can be implemented using standard VPN protocols over TCP/IP with existing APNs to establish VPN tunnels to VPN servers. Through this method of having the VPN server to authenticate access over the standard infrastructure networks, we can even have an 3G/4G access via mobile phones to be enabled. Hence, recent research focus has shifted in catering to interoperability and scalability of low-cost security solutions for IoT deployments. Using a low-cost router (or Raspberry Pi) to run OpenVPN, we established secure communications among a cluster of IoT devices for a real-time water monitoring system. In this research, we have adopted a proof-of-concept approach, which is quite complementary to existing related studies [24,25,80,81]. Each existing research work has been developed with a different application context, and our paper is the first of its kind to propose a simple, low-cost end-to-end security model configured to the unique context of a scalable smart water monitoring system using IoT infrastructure. In our distinguishing solution, we have addressed the high overheads, and complex configurations of OpenVPN reported in these existing works.

In summary, many IoT network solutions exist—however, many of them are proprietary. It is not the intention of this paper to compare the merits of the many standards and proprietary IoT technologies, but rather to provide a demonstration of what can be done with open-source platforms. Our aim in the proposed solution was to cater to scalability, security, and interoperability for a large-scale IoT deployment. In our solution applied to a large-scale water monitoring system, we used TCP/IP access points available on-site and created VPN tunnels to a remote VPN Server via the on-board VPN Client. These are standards that are well known, and many open-source libraries exist that allow for easy, transparent, and non-proprietary implementations in any operating system, including Linux. Overall, this paper has proposed a unique solution specific to IoT and demonstrates how this is implemented as a convergence of IoT devices, VPN client/server security, and mobile phone apps to configure, monitor, and control an IoT environment in a secure manner. Little work is available that takes advantage of these three readily available technologies in proposing a practically novel approach to address the main security concerns in a large-scale IoT environment. We strongly believe that our proposed IoT security model and its unique implementation in a large IoT infrastructure, such as a smart water monitoring system, would be of practical and academic value for a secure IoT deployment in the present and future smart environments.

8. Conclusions and Future Work

Despite the rapid advancement of IoT technologies, security and privacy threats continue to hamper the benefits of IoT based smart environments ranging from domestic to industrial deployments. Current IoT technologies and device vendors lack insights into the requirement of scalability, interoperability, and end-to-end security of dynamically changing large IoT environments.

Firstly, this paper uncovered the vulnerabilities in the IoT architecture by identifying the security attacks possible in each of the four layers, namely, Device or Perception Layer, Network or Transmission Layer, Middleware or Service Layer, and Application or Business Layer. We identified the security requirements of IoT architecture by differentiating the unique characteristics of IoT networks as compared to internet networks. Secondly, with the aim of meeting the baseline IoT security requirements for smart environments of the future, we proposed a simple, adaptive, and scalable end-to-end security model for a large IoT infrastructure. Our model with low-cost advanced Raspberry Pi controllers provisioned for the convergence of more sophisticated networking protocols embedded at the IoT monitoring interface. We employed a unique configuration of VPN servers and clients with Raspberry Pi as the IoT gateway to establish a low-cost VPN to connect several IoT devices securely. Thirdly, a pilot implementation of the proposed security model for a large IoT infrastructure, was successfully demonstrated with a prototype as a case scenario. We illustrated the seamless integration of a secure IoT infrastructure connecting various sensors of a water tank system to remotely control and monitor the smart environment via user-friendly mobile Apps. We provided the implementation details with sample use case visual illustrations to gain IoT data insights based on water level readings, water usage, and other data analytics.

This paper provided the conceptual prototype design and implementation of our proposed model, and for future work, it would be beneficial to assess and validate the model effectiveness with security metrics and simulated malicious attacks from different access points of the IoT network. Future research would also involve studying large-scale secure IoT deployment in other real-world case scenarios.

Author Contributions: Conceptualization, A.O. and S.V.; methodology, S.V.; resources, A.O.; writing—original draft preparation, A.O. and S.V.; writing—review and editing, A.O. and S.V.; visualization, A.O.; supervision, S.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work contains processed information from publicly available data and opensource software components provided by IBM and Github that provide support to the pilot study of this ongoing research work. The authors thank the students (Minh Thong Nguyen, Trent Mackie, Prabin Poudel) and the industry partner (Peter Shackleton) for taking part in one of the project's use cases that helped in the implementation and testing for the case study. The authors also wish to acknowledge the reviewers and editors for their invaluable inputs and support rendered for this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Srirama, S. Mobile web and cloud services enabling internet of things. *CSI Trans. ICT* **2017**, *5*, 109–117. [CrossRef]
2. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A survey on iot security: Application areas, security threats, and solution architectures. *IEEE Access* **2019**, *7*, 82721–82743. [CrossRef]

3. Kang, W.M.; Moon, S.Y.; Park, J.H. An enhanced security framework for home appliances in smart home. *Hum. Cent. Comput. Inf. Sci.* **2017**, *7*, 6. [CrossRef]
4. Venkatraman, S. A Self-Learning Framework for the IoT Security. In *Smart Devices, Applications, and Protocols for the IoT*; IGI Global: Hershey, PA, USA, 2019; pp. 34–53.
5. Ondiege, B.; Clarke, M.; Mapp, G. Exploring a new security framework for remote patient monitoring devices. *Computers* **2017**, *6*, 11. [CrossRef]
6. Fernandes, E.; Jung, J.; Prakash, A. Security analysis of emerging smart home applications. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 636–654.
7. Roy, A.; Datta, A.; Siddiquee, J.; Poddar, B.; Biswas, B.; Saha, S.; Sarkar, P. Energy-efficient Data Centers and smart temperature control system with IoT sensing. In Proceedings of the 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 13–15 October 2016; pp. 1–4.
8. Medwed, M. IoT security challenges and ways forward. In *Proceedings of the 6th International Workshop on Trustworthy Embedded Devices*; ACM: New York, NY, USA, 2016; p. 55.
9. Sivanathan, A.; Gharakheili, H.H.; Sivaraman, V. Managing iot cyber-security using programmable telemetry and machine learning. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 60–74. [CrossRef]
10. Miettinen, M.; Marchal, S.; Hafeez, I.; Asokan, N.; Sadeghi, A.R.; Tarkoma, S. IoT sentinel: Automated device-type identification for security enforcement in IoT. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 2177–2184.
11. Lochab, K.; Yadav, D.K.; Singh, M.; Sharmab, A. Internet of things in cloud environment: Services and challenges. *Int. J. Database Theory Appl.* **2017**, *10*, 23–32. [CrossRef]
12. Guarnizo, J.D.; Tambe, A.; Bhunia, S.S.; Ochoa, M.; Tippenhauer, N.O.; Shabtai, A.; Elovici, Y. SIPHON: Towards scalable high-interaction physical honeypots. In *Proceedings of the ACM Workshop on Cyber-Physical System Security*; ACM: New York, NY, USA, 2017; pp. 57–68.
13. Venkatraman, S.; Overmars, A. New Method of Prime Factorisation-Based Attacks on RSA Authentication in IoT. *Cryptography* **2019**, *3*, 20. [CrossRef]
14. Diaz Lopez, D.; Blanco Uribe, M.; Santiago Cely, C.; Vega Torres, A.; Moreno Guataquira, N.; Moron Castro, S.; Nespoli, P.; Gomez Marmol, F. Shielding IoT against Cyber-Attacks: An Event-Based Approach Using SIEM. *Wirel. Commun. Mob. Comput.* **2018**, *2018*. [CrossRef]
15. Knieriem, B.; Zhang, X.; Levine, P.; Breitinger, F.; Baggili, I. An overview of the usage of default passwords. In *Digital Forensics and Cyber Crime, ICDF2C 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*; Matoušek, P., Schmiedecker, M., Eds.; Springer: Cham, Germany, 2018; Volume 216, pp. 195–203.
16. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT Security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2702–2733. [CrossRef]
17. Makhdoom, I.; Abolhasan, M.; Lipman, J.; Liu, R.P.; Ni, W. Anatomy of threats to the Internet of things. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1636–1675. [CrossRef]
18. Alaba, F.A.; Othman, M.; Hashem, I.A.T.; Alotaibi, F. Internet of Things security: A survey. *J. Netw. Comput. Appl.* **2017**, *88*, 10–28. [CrossRef]
19. Rizal, R.; Riadi, I.; Prayudi, Y. Network forensics for detecting flooding attack on internet of things (IoT) device. *Int. J. Cyber Secur. Digit. Forensics* **2018**, *7*, 382–390.
20. Stellios, I.; Kotzanikolaou, P.; Psarakis, M.; Alcaraz, C.; Lopez, J. A survey of IoT-enabled cyberattacks: Assessing attack paths to critical infrastructures and services. *IEEE Commun. Surv. Tuts.* **2018**, *20*, 3453–3495. [CrossRef]
21. Moosavi, S.; Gia, T.; Nigussie, E.; Rahmani, A.; Virtanen, S.; Tenhunen, H.; Isoaho, J. End-to-end security scheme for mobility enabled healthcare internet of things. *Future Gener. Comput. Syst.* **2016**, *64*, 108–124. [CrossRef]
22. Ngu, A.H.; Gutierrez, M.; Metsis, V.; Nepal, S.; Sheng, Q.Z. IoT middleware: A survey on issues and enabling technologies. *IEEE Internet Things J.* **2017**, *4*, 1–20. [CrossRef]
23. Das, P.K.; Narayanan, S.; Sharma, N.K.; Joshi, A.; Joshi, K.; Finin, T. Context-sensitive policy based security in internet of things. In Proceedings of the 2016 IEEE International Conference on Smart Computing (SMARTCOMP), St. Louis, MO, USA, 18–20 May 2016; pp. 1–6.

24. Iqbal, M.; Riadi, I. Analysis of security virtual private network (VPN) using openVPN. *Int. J. Cyber Secur. Digit. Forensics* **2019**, *8*, 58–65.
25. Nundloll, V.; Porter, B.; Blair, G.S.; Emmett, B.; Cosby, J.; Jones, D.L.; Chadwick, D.; Winterbourn, B.; Beattie, P.; Dean, G.; et al. The Design and Deployment of an End-To-End IoT Infrastructure for the Natural Environment. *Future Internet* **2019**, *11*, 129.
26. Lee, C.; Fumagalli, A. Internet of Things Security-Multilayered Method for End to End Data Communications Over Cellular Networks. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 24–28. [CrossRef]
27. Alrawi, O.; Lever, C.; Antonakakis, M.; Monrose, F. Sok: Security Evaluation of Home-based IoT Deployments. In Proceedings of the IEEE Symposium on Security and Privacy (S&P), San Francisco, CA, USA, 20–22 May 2019.
28. Fang, H.; Qi, A.; Wang, X. Fast authentication and progressive authorization in large-scale IoT: How to leverage ai for security enhancement. *IEEE Netw.* **2020**, *34*, 24–29. [CrossRef]
29. Can, O.; Sahingoz, O.K. A survey of intrusion detection systems in wireless sensor networks. In Proceedings of the 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), Istanbul, Turkey, 27–29 May 2015; pp. 1–6.
30. Hsu, C.; Lin, J.C. An empirical examination of consumer adoption of internet of things services: Network externalities and concern for information privacy perspectives. *Comput. Hum. Behav.* **2016**, *62*, 516–527. [CrossRef]
31. Abomhara, M.; Koien, G. Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks. *J. Cyber Secur.* **2015**, *4*, 65–88. [CrossRef]
32. Sinha, R.S.; Wei, Y.; Hwang, S.H. A survey on LPWA Technology: LoRa and NB-IoT. *ICT Express* **2017**, *3*, 1–21. [CrossRef]
33. Mekkia, K.; Bajica, E.; Chaxela, F.; Meyerb, F. A Comparative Study of LPWAN Technologies for Large-Scale IoT Deployment. *ICT Express* **2019**, *5*, 1–7. [CrossRef]
34. Basu, D.; Gu, T.; Mohapatra, P. Security issues of low power wide area networks in the context of LoRa networks. *arXiv* **2020**, arXiv:abs/2006.16554.
35. Aras, E.; Ramachandran, G.S.; Lawrence, P.; Hunghes, D. Exploring the Security Vulnerabilities of LoRa. In Proceedings of the 3rd IEEE International Conference on Cybernetics, Exeter, UK, 21–23 June 2017.
36. Butun, I.; Pereira, N.; Gidlund, M. Security risk analysis of LoRaWAN and future directions. *Future Internet* **2019**, *11*, 3. [CrossRef]
37. Pathak, G.; Gutierrez, J.; Rehman, S.U. Security in low powered wide area networks: Opportunities for software defined network-supported solutions. *Electronics* **2020**, *9*, 1195. [CrossRef]
38. Lee, W.; Kim, N. Security Policy Scheme for an Efficient Security Architecture in Software-Defined Networking. *Information* **2017**, *8*, 65. [CrossRef]
39. Jose, A.C.; Malekian, R.; Ye, N. Improving home automation security; integrating device fingerprinting into smart home. *IEEE Access* **2016**, *4*, 5776–5787. [CrossRef]
40. Chu, G.; Apthorpe, N.; Feamster, N. Security and privacy analyses of internet of things children's toys. *IEEE Internet Things J.* **2019**, *6*, 1978–1985. [CrossRef]
41. Apthorpe, N.; Reisman, D.; Sundaresan, S.; Narayanan, A.; Feamster, N. Spying on the smart home: Privacy attacks and defenses on encrypted IoT traffic. *arXiv* **2017**, arXiv:1708.05044.
42. Hamza, A.; Ranathunga, D.; Gharakheili, H.H.; Benson, T.A.; Roughan, M.; Sivaraman, V. Verifying and monitoring IoTs network behavior using MUD profiles. *arXiv* **2019**, arXiv:1902.02484.
43. Thangavelu, V.; Divakaran, D.M.; Sairam, R.; Bhunia, S.S.; Gurusamy, M. DEFT: A distributed IoT fingerprinting technique. *IEEE Internet Things J.* **2019**, *6*, 940–952. [CrossRef]
44. Ntuli, N.; Abu-Mahfouz, A. A simple security architecture for smart water management system. *Procedia Comput. Sci.* **2016**, *83*, 1164–1169. [CrossRef]
45. Singh, J.; Thomas, F.J.-M.; Pasquier, J.B.; Ko, H.; Eyers, D.M. Twenty security considerations for cloud-supported internet of things. *IEEE Internet Things J.* **2016**, *3*, 269–284. [CrossRef]
46. Tawalbeh, L.; Muheidat, F.; Tawalbeh, M.; Quwaider, M. IoT privacy and security: Challenges and solutions. *Appl. Sci.* **2020**, *10*, 4102. [CrossRef]

47. Singh, V.; Pandey, S.K. Revisiting Cloud Security Threats: Replay attack. In Proceedings of the 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 14–15 December 2018; pp. 1–6.
48. Liyanage, M.; Braeken, A.; Kumar, P.; Ylianttila, M. *IoT Security: Advances in Authentication*; John Wiley &Sons: West Sussex, UK, 2020.
49. Creswell, J.W. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*; Sage Publications: Thousand Oaks, CA, USA, 2013.
50. Rossman, G.B.; Wilson, B.L. Numbers and words: Combining quantitative and qualitative methods in a single large-scale evaluation study. *Eval. Rev.* **1985**, *9*, 627–643. [CrossRef]
51. Baxter, P.; Jack, S. Qualitative case study methodology: Study design and implementation for novice researchers. *Qual. Rep.* **2008**, *13*, 544–559.
52. Merriam, S.B.; Tisdell, E.J. *Qualitative Research: A Guide to Design and Implementation*; John Wiley: San Francisco, CA, USA, 2015.
53. Strauss, A.L.; Corbin, J.M. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*; Sage: Thousand Oaks, CA, USA, 1998.
54. Rashid, Y.; Rashid, A.; Warraich, M.; Sabir, S.; Waseem, A. Case study method: A step-by-step guide for business researchers. *Int. J. Qual. Methods* **2019**, *18*, 160940691986242. [CrossRef]
55. Denzin, N.K.; Lincoln, Y.S. *Collecting and Interpreting Qualitative Materials*; Sage: London, UK, 1998.
56. Scotland, J. Exploring the philosophical underpinnings of research: Relating ontology and epistemology to the methodology and methods of the scientific, interpretive, and critical research paradigms. *Engl. Lang. Teach.* **2012**, *5*, 9–16. [CrossRef]
57. Wilson, J. *Essentials of Business Research: A Guide to Doing Your Research Project*; Sage: Thousand Oaks, CA, USA, 2014.
58. Brynard, D.J.; Hanekom, S.X.; Brynard, P. *Introduction to Research*, 3rd ed.; Van Schaik: Pretoria, South Africa, 2014.
59. Orlikowski, W.J.; Baroudi, J.J. Studying information technology in organizations: Research approaches and assumptions. *Inf. Syst. Res.* **1991**, *2*, 1–28. [CrossRef]
60. Verdouwab, C.; Sundmaeker, H.; Tekinerdogana, B.; Conzon, D.; Montanaro, T. Architecture framework of IoT-based food and farm systems: A multiple case study. *Comput. Electron. Agric.* **2019**, *165*, 104939. [CrossRef]
61. Baskarada, S. Qualitative case study guidelines. *Qual. Rep.* **2014**, *19*, 1–18.
62. Bansal, S.; Kumar, D. IoT ecosystem: A Survey on devices, gateways, operating systems, middleware and communication. *Int. J. Wirel. Inf. Netw.* **2020**, *27*, 340–364.
63. Ferdowsi, A.; Saad, W. Deep learning for signal authentication and security in massive Internet-of-Things systems. *IEEE Trans. Commun.* **2019**, *67*, 1371–1387. [CrossRef]
64. Farris, I.; Taleb, T.; Khettab, Y.; Song, J. A survey on emerging SDN and NFV security mechanisms for IoT systems. *IEEE Commun. Surv. Tuts.* **2019**, *21*, 812–837. [CrossRef]
65. Parvin, S.; Venkatraman, S.; de Souza-Daw, T.; Fahd, K.; Jackson, J.; Kaspi, S.; Cooley, N.; Saleem, K.; Gawanmeh, A. Smart Food Security System Using IoT and Big Data Analytics. In *Proceedings of the 16th International Conference on Information Technology-New Generations (ITNG 2019)*; Advances in Intelligent Systems and Computing; Latifi, S., Ed.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 800, pp. 253–258.
66. Karmakar, K.K.; Varadharajan, V.; Tupakula, U.; Hitchens, M. Policy based security architecture for software defined networks. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*; ACM: New York, NY, USA, 2016; pp. 658–663.
67. Pal, S.; Hitchens, M.; Varadharajan, V. Towards A Secure Access Control Architecture for the Internet of Things. In Proceedings of the IEEE 42nd Conference on Local Computer Networks (LCN), Singapore, 9–12 October 2017.
68. Capellupo, M.; Liranzo, J.; Bhuiyan, M.Z.A.; Hayajneh, T.; Wang, G. Security and attack vector analysis of IoT devices. In *Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 593–606.
69. Fang, H.; Wang, X.; Hanzo, L. Learning-aided physical layer authentication as an intelligent process. *IEEE Trans. Commun.* **2019**, *67*, 2260–2273. [CrossRef]
70. Xu, T.; Gao, D.; Dong, P.; Zhang, H.; Foh, C.H.; Chao, H.C. Defending against new-flow attack in sdn-based internet of things. *IEEE Access* **2017**, *5*, 3431–3443. [CrossRef]

71. Parvin, S.; Gawanmeh, A.; Venkatraman, S.; Alwadi, A.; Al-Karak, J. Efficient Lightweight Mechanism for Node Authentication in WBSN. In Proceedings of the Advances in Engineering Technology & Sciences Multi-Conferences (ASET 2018), Dubai, UAE, 6–7 February 2018.
72. Pongle, P.; Chavan, G. A survey: Attacks on RPL and 6LoWPAN in IoT. In Proceedings of the IEEE International Conference on Pervasive Computing (ICPC), Pune, India, 8–10 January 2015; pp. 1–6.
73. Khan, F.I.; Hameed, S. Understanding security requirements and challenges in internet of things (IoTs): A review. *J. Comp. Netw. Communic* **2019**, 9629381:1–9629381:14.
74. Anirudh, M.; Thileeban, S.A.; Nallathambi, D.J. Use of honeypots for mitigating DoS attacks targeted on IoT networks. In Proceedings of the 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai, India, 10–11 January 2017; pp. 1–4.
75. Lyu, M.; Sherratt, D.; Sivanathan, A.; Gharakheili, H.H.; Radford, A.; Sivaraman, V. Quantifying the reflective DDoS attack capability of household IoT devices. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*; ACM: New York, NY, USA, 2017; pp. 46–51.
76. Pal, S.; Hitchens, M.; Varadharajan, V. Modeling Identity for the Internet of Things: Survey, Classification and Trends. In Proceedings of the 12th International Conference on Sensing Technology (ICST), Limerick, Ireland, 4–6 December 2018.
77. Hesham, A.; Sardis, F.; Wong, S.; Mahmoodi, T.; Tatipamula, M. A simplified network access control design and implementation for m2m communication using sdn. In Proceedings of the Wireless Communications and Networking Conference Workshops (WCNCW), San Francisco, CA, USA, 19–22 March 2017; pp. 1–5.
78. Lu, Y.; Ling, Z.; Zhu, S.; Tang, L. Sdtcp: Towards datacenter TCP congestion control with SDN for IoT applications. *Sensors* **2017**, *17*, 109. [CrossRef]
79. Sanchez-Iborra, R.; Sánchez-Gómez, J.; Pérez, S.; Fernández, P.J.; Santa, J.; Hernández-Ramos, J.L.; Skarmeta, A.F. Enhancing lorawan security through a lightweight and authenticated key management approach. *Sensors* **2018**, *18*, 1833. [CrossRef]
80. Bist, P.K.; Mekade, A.S.; Nair, A.M.; Chatterjee, M. Secure VPN server deployed on raspberry pi. *J. Netw. Commun. Emerg. Technol. (Jncet.)* **2018**, *8*, 27–31.
81. Caldas-Calle, L.; Jara, J.; Huerta, M.; Gallegos, P. QoS evaluation of VPN in a Raspberry Pi devices over wireless network. In Proceedings of the 2017 International Caribbean Conference on Devices, Circuits and Systems (ICCDCS), Cozumel Roo, Mexico, 5–7 June 2017; pp. 125–128. [CrossRef]
82. Feilner, M.; Graf, N. *Beginning OpenVPN 2.0.9. Build and Integrate Virtual Private Networks Using OpenVPN*; Packt Publishing: Birmingham, UK, 2009.
83. Qiu, W.; Saleem, K.; Pham, M.; Halpern, M.; Beresford-Smith, B.; Overmars, A.; Dassanayake, K.; Thoms, G. Robust multipath links for wireless sensor networks in irrigation applications. In Proceedings of the 2007 3rd International Conference on Intelligent Sensors, Melbourne, Australia, 3–6 December 2007; pp. 95–100.
84. Overmars, A. Communications Apparatus, System and Method. International Patent Publication No. WO/2010/132929, 2020. Available online: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2010132929 (accessed on 10 January 2020).
85. Moreau, L. Sump Pump Water Level. Available online: http://instructables.com/id/Sump-pump-water-level-The-software (accessed on 15 January 2020).
86. Vishwasrao. *SMART Water Tank Monitoring System. IBM Developer Recipes*; IBM: Armonk, NY, USA, 2017.
87. Github. Available online: https://github.com/paulknewton/pi-tank-watcher (accessed on 15 January 2020).
88. Sanchez-Iborra, R.; Maria-Dolores, C. State of the art in LP-WAN solutions for industrial IoT services. *Sensors* **2016**, *16*, 708. [CrossRef] [PubMed]