Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8896637, 18 pages
https://doi.org/10.1155/2020/8896637

Research Article

Designing Efficient Smart Home Management with IoT Smart Lighting: A Case Study

Tran Anh Khoa[1], **Le Mai Bao Nhu**[2], **Hoang Hai Son**[3], **Nguyen Minh Trong**[2], **Cao Hoang Phuc**[2], **Nguyen Thi Hoang Phuong**[2], **Nguyen Van Dung**[2], **Nguyen Hoang Nam**[1], **Dong Si Thien Chau**[2] and **Dang Ngoc Minh Duc**[4]

[1]*Modeling Evolutionary Algorithms Simulation and Artificial Intelligence, Faculty of Electrical & Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam*
[2]*Faculty of Electrical & Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam*
[3]*Faculty of Mechanical, Electrical, Electronic and Automotive Engineering, Nguyen Tat Thanh University, Ho Chi Minh City 700000, Vietnam*
[4]*School of Graduate Studies, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam*

Correspondence should be addressed to Hoang Hai Son; hhson@ntt.edu.vn

Received 9 April 2020; Revised 28 September 2020; Accepted 14 October 2020; Published 20 November 2020

Academic Editor: Hui Cheng

Smart homes are an element of developing smart cities. In recent years, countries around the world have spared no effort in promoting smart cities. Smart homes are an interesting technological advancement that can make people's lives much more convenient. The development of smart homes involves multiple technological aspects, which include big data, mobile networks, cloud computing, Internet of Things, and even artificial intelligence. Digital information is the main component of signal control and flow in a smart home, while information security is another important aspect. In the event of equipment failure, the task of safeguarding the system's information is of the utmost importance. Since smart homes are automatically controlled, the problem of mobile network security must be taken seriously. To address these issues, this paper focuses on information security, big data, mobile networks, cloud computing, and the Internet of Things. Security efficiency can be enhanced by using a Secure Hash Algorithm 256 (SHA-256), which is an authentication mechanism that, with the help of the user, can authenticate each interaction of a given device with a WebServer by using an encrypted username, password, and token. This framework could be used for an automated burglar alarm system, guest attendance monitoring, and light switches, all of which are easily integrated with any smart city base. In this way, IoT solutions can allow real-time monitoring and connection with central systems for automated burglar alarms. The monitoring framework is developed on the strength of the web application to obtain real-time display, storage, and warning functions for local or remote monitoring control. The monitoring system is stable and reliable when applying SHA-256.

1. Introduction

A smart city is a future trend solution that uses various electronic Internet of Things (IoT) sensors to collect data. Insights obtained from that data are subsequently used to manage assets, resources, and services efficiently; as such, the data is used to improve operations across the city. It includes data collected from citizens, devices, buildings, and assets that are processed and analyzed to monitor and manage traffic and transportation systems, power plants, utilities, water supply networks, waste, crime detection, information systems, universities, libraries, hospitals, and other community services [1]. Data is at the heart of the IoT, but to make it trustworthy enough for widespread acceptance, the security and privacy of that data must be protected. It is at this focus thing of demand for innovation and the requirements for acceptable data security and privacy [2].

The smart home is one of the most prominent applications in the paradigm of the IoT. While it has added a level of comfort and convenience to users' everyday lives, it also

brings a unique security challenge of mitigating insider threats posed by legitimate users. Such threats primarily arise due to the sharing of IoT devices and the presence of complex social and trust relationships among users. State-of-the-art home IoT platforms manage access control by deploying various multifactor authentication mechanisms. Nevertheless, such hard security measures are inadequate to defend against insider threats, and there is a growing need to integrate user behavior and environmental context in order to make intelligent authorization decisions [3].

Some of the leading home IoT platforms that have emerged over the past few years are Samsung's SmartThings, Apple's HomeKit, and Google's Android Things. These platforms are energy efficient, connect heterogeneous devices and protocols, allow remote control and actuation, and support third-party application development [4].

The popularity of smart home appliances is causing the increasing development of the IoT. For example, most smart home devices—such as smart televisions, fridges, dishwashers, cooling systems, and heating machines, among others—are connected to the Internet to make people's lives more comfortable and easier. Nowadays, the ability of smart home management to combine and control devices has increased and developed significantly. A smart home is like a bespoke home based on personal preferences and individual specifications. It can regulate and control the internal/external features of a house, such as lighting, temperature, doors, and windows. Smart home management can be used to set the brightness and warmth of a room, adjust background music, and even schedule TV programs to be recorded and played, all depending on the homeowner's taste and decisions. With the use of a smartphone, it is also possible to check a home's current status remotely. Furthermore, settings can be adjusted when away from the house. For example, air conditioners that use temperature sensors and remote-controlled lighting systems can be controlled, and the television can be switched on to make it look as though the owner is at home [5–8]. With the aid of specific smart devices, smart home applications and services offer living convenience with consideration of the homeowner's living pattern, living style, and various other preferences. In other words, a homeowner can control and monitor areas of their home with the use of a smartphone [9–11]. As such, the intelligence and versatility of a smart home service make it possible for users to manage their houses regardless of where they are and what they are doing, through the use of a smartphone. However, although the system is convenient and efficient, it can be vulnerable and open to security threats. Unlike smartphones, which have a decent level of security technology due to their various uses (such as business communications, family phone calls, text messaging, and internet surfing), smart home appliances have weak security technology so they are exposed to various attacks [11–13].

This article introduces a private security framework for smart devices, which can be applied to smart home devices as well as smartphones. The suggested security framework uses SHA-256 technology, which protects against several threats/actions, including infiltrating codes. Additionally, this article also explains module functions, which are limited by access control in order to protect the modules of home tools in smart homes.

The idea of building an intelligent lighting system will also be presented in this paper. This is a linkage system that is both a lighting system and a security monitoring system in one. This system not only enables lights to switch on and off when someone is present, but users can also set the color and brightness of lights as desired. As mentioned, users can monitor the status of a home when they are away through the use of a WebServer mobile application and mobile networks. This system is flexible in the sense that it allows one user to control multiple devices, or multiple users to control numerous devices at the same time.

The main contributions of this study can be summarized as follows.

(i) Most smart lighting systems are designed with a simple motion sensor and can detect whether a human being has moved in or out of the sensor's range. However, the novel smart lighting system outlined in this article integrates the main functions of an ESP 8266-12F microcontroller and a two-layer circuit using 24 LED WS set in a spiral-shaped serial configuration with a cooling pad. The purpose is to save the microcontroller gates and to optimize the lamp area. The benefit of the design is that it allows the user to strengthen the security of their smart home. Another added bonus is that it significantly reduces hardware costs

(ii) By employing both a gateway and an Access Point (AP), the wireless device must be allocated to the additional data communications occurring between the two, which unfortunately advances the issue of competition. As a solution, the IoT ESP8266-12F is applied so that the data received from sensors are transmitted to the data server or the data center through the Internet. This is a convenient and functional process, because the gateway and the AP will have been integrated with the IoT ESP8266-12F, thus representing another benefit of the system

(iii) The system can be used even when no Internet or 4G is available, by using WLAN technology APs. This is achievable because the ESP8266-12F-integrated smart lighting device is deployed and can access the Internet anytime and anywhere without having to reinstall the specifications, thus providing another level of convenience to the user

(iv) To improve the security of the system, a compact server system is built that can connect to the network, with a high processing capability. Moreover, the server allows users to install software that gives them the option to request additional services and resources

(v) This paper demonstrates a novel solution for server communication with each device and explains the

process of authenticating encrypted usernames, passwords, tokens, and codes with the use of SHA-256 for added security. The study also describes how a server can engage in multidevice communication, that is, the process of transferring encrypted tokens to several devices using SHA-256 for added security

(vi) A WebServer is built that is capable of tracking and monitoring smart homes in real time, by running on Raspberry Pi 3+, HTML, JavaScript support with WebSocket, Socket.IO2 library, and Python 3

This article is arranged as follows. Section 1 introduces the technology and research trends of smart home security. Then, Section 2 describes safety considerations and relevant research on IoT smart cities. Section 3 proposes an internal security framework for smart lighting systems. Section 4 then presents the initial configuration and demo frameworks used for the system, before Section 5 presents a comparative analysis. Finally, the conclusion is drawn in Section 6.

2. Related Works

2.1. Smart Home Technologies Based on IoT Applications. A number of relevant smart home systems based on IoT applications have been recently developed, with the aim to make human living more convenient and environmentally friendly. However, real-world environments present several challenges. A smart home is controlled remotely; therefore, it is designed to be energy efficient with basic features consisting of lighting and switch modes. Moreover, a low-cost network can be designed based on the use of a gateway composed of Arduino with Ethernet, ZigBee technology, and an Android device that performs as a home environment controller. The disadvantages of this system do not apply to all security technologies, and such a solution is not a novelty in a smart home. Notably, this system does not enable/demonstrate the use of sensors in home monitoring scenarios (e.g., energy consumption, water level, and indoor temperature monitoring) [14, 15].

Papers [16, 17] present ZigBee technology for home automation control, using a PC as a gateway and server through Wi-Fi wireless data communication technology, which can access a home subnet on the Android platform with remote monitoring. Depending on the open-source code and hardware from the other scenario, such a system has different disadvantages. Some modern smart homes utilize Wi-Fi for wireless communication technology [18]. This provides an easy way to integrate a monitoring system with a smart home. However, such low-cost circuit hardware is not easy to implement, as it involves complications with some devices.

Meanwhile, a smart home system uses integrated sensors, actuators, wireless networks, and graphical user interfaces, which have positive, flexible, secure, and cost-effective benefits. A sensor network can convert an original home into a smart home, by introducing sensors for lighting, temperature, pressure, humidity, motion, fire alarms, and dust/air, among others [19]. Such a system uses a combination of Raspberry Pi 2 and ESP8266 microcontrollers as hardware, and an open-source code platform. However, this platform faces many challenges; one is the security and privacy aspect. Similarly, [20] describes a straightforward platform based on open-source code, where the authors present a solution by integrating ESP8266 and MQTT for remote monitoring in a smart home. Furthermore, the authors did not apply security technology for safety purposes, and the system was run on a PC. This reduces the security risk of the IoT system and increases the cost.

A smart home platform was designed and implemented using an effective system for fog computing based on ZigBee and Wi-Fi wireless communication technologies [21], termed ZiWi. It uses an open-source code for an application. The author also enabled technology that is designed for hardware for the IoT node. Other objectives were to create a low-cost platform with easy-to-change settings.

Another smart home system was designed with Raspberry Pi and node MCU as the backend, which can notify the user if someone tries to trespass within the range of the system, and keeps track of money spent each month. Instead of ZigBee or Wi-Fi technologies, the communication link presented in this paper uses a telegram bot. Thus, the system cannot be used in a real-time environment and has low-level security technology [22].

Wireless communication technology has also been integrated with microcontroller technology, which is a hot topic in research on the IoT platform. Based on these technologies, the capabilities of sensing, identification, and communication can be embedded in several smart devices. In [23], the authors designed and implemented an IoT AP that carries the functionalities of coordinating multiple wireless transmissions. However, a high-performance AP is required as computer access, resulting in high equipment costs.

Android applications can remotely control a smart lighting system in a smart home via mobile or tablet devices [24]. With this in mind, smart LEDs have been designed for particular user requirements, such as temperature evaluation and supposed illumination, with the use of ZigBee technology for data transmission. This system is more suitable for use in factories than in small houses or apartments. Table 1 reviews the principal characteristics of the various designs of smart home technologies discussed above and presents the main features of the server, communication link technology, sensors, IoT node, security technology, and application. However, the smart lighting system developed in this study is not designed with the circuit board integrated into the sensors and is not used as an open-source system. This research is aimed at developing a circuit board with two layers and 24 LEDs and at building a web application with a secure IoT-based system. Moreover, the system can collect data securely and efficiently by communicating between the server and the target nodes.

2.2. Security Technology IoTs in the Smart Home. In a smart home, almost all IoT-based home automation systems, including actuators and sensors, are located in the interior of the building. This system and the sensor equipment are connected to the local server via wireless communication

Table 1: Comparison of principal characteristics of the most relevant smart home technologies.

Ref.	Server	Communication	Node hardware	Sensors	Purpose	Prices	Security	Application
[15]	Android	X10, ZigBee	Arduino	Light and switch modules	Energy efficient & remote control home environment	NA	No	Open-source code
[16]	NA	ZigBee	CC2530	NA	Home control	>$1000	NA	Open-source code
[17]	NA	Wi-Fi, ZigBee	A20 SoC	Light, door, temperature	Home control	>$500	NA	Open-source code
[18]	NA	Wi-Fi, IR	NA	Dust sensor	Smart home management	>$500	NA	Open-source code
[19]	Raspberry Pi 2+	Wi-Fi	ESP8266	NA	Remote control home environment	>$500	No	Open-source code
[20]	PC	Wi-Fi	ESP8266	Luminosity, LDR sensors, LED, and buzzer	Remote control home environment	>$1000	No	Open-source code
[21]	Raspberry Pi	Wi-Fi, ZigBee	ESP8266	PIR, humidity, temperature sensors	Remote control home environment	>$500	No	Open-source code
[22]	XP8000	Wi-Fi, ZigBee	NA	Light switch, light dimmer, and light sensors	LED lighting system	>$1000	No	Open-source code
[23]	NA	Wi-Fi, ZigBee	Wi-Fi AP	No	Remote control home environment, and IoT AP for smart home	>$500	No	Open-source code
[24]	Raspberry Pi, NodeMCU	Wi-Fi	ESP8266	Rain, door, PIR, and DHT22 sensors	Home control	<$500	No	Open-source code
Our proposed	Raspberry Pi 3+	Wi-Fi	ESP8266-12F	D203B PIR, humidity, temperature sensor, LED WS2812B, and buzzer	Energy efficient, low-cost, security, remote control home environment, smart light design and develop a server for smart home	<$200	SHA-256	JavaScript, HTML, CSS, C++, and Python 3

for data collection and analysis. However, a significant problem is that of securely transmitting the received data from the sensor nodes to the compatible receiver. This subsection presents the security challenges arising from these platforms and compares the developed framework with other state-of-the-art frameworks that are available for smart homes.

In [25], a secure IoT low-cost edge device authentication is presented. The authors pursued two objectives. The first objective was to propose an ID matching scheme to verify the identity of an edge device, to prevent others from attempting to impersonate an ID. To meet the second objective, the authors applied a communication protocol to authenticate edge devices for an IoT system by using undesirable device IDs. Using a PC as a gateway made it a challenge to secure low costs in this paper, because the PC is not integrated with the hardware. Therefore, equipment design is an essential factor to get right in the IoT field.

In [26], the authors propose a secure platform for IoT smart homes. Three contributions are presented: (1) a hardware platform was designed by applying the Intel board, (2) an energy-efficient security algorithm for data encryption was generated, and (3) the authors compared the energy efficiency of their solution with that of other research. Their developed system is high cost and has only been tested in the laboratory since it was applied to the test board and ThingSpeak application. Thus, the principal disadvantage of this paper pertains to the Intel board and the high cost of the system.

In [27], hash-chain-based authentication for IoT devices and web services is presented. The system can be used to authenticate each interaction of the apparatus with a REST web service by using one-time passwords (OTP).

The security framework in [28] presents a sincerity system that uses self-signing and access control techniques for checking security warnings such as data qualification, leakage, and code fabrication. Moreover, the paper explains some module functions that are defined by access control to protect the modules of home devices in a smart home.

Data security is a critical investigation topic in the context of the IoT, as IoT systems penetrate deeper into users' personalities, and these devices function, process, and store various kinds of data. The paper discusses several challenges faced by security and privacy features, mainly for applications operating on resource-constrained devices. The authors of [29] studied the use of a cryptographic block, hashing algorithms, message authentication codes, signature mechanisms, and critical exchange protocols performed on state-of-the-art resource-constrained devices. The authors found the optimal hash function to add to the constrained application protocol to increase security without lowering performance. The hash function applied in this paper is SHA-224 [30]. The authors found the optimal SHA-224 function to add to the constrained application protocol to increase security without lowering performance. The other security framework proposed in [31] can predict and protect against various possible malicious advances in the ZigBee communication network and respond with a warning to the system administrator.

By combining the bitwise XOR operation and hash function, the authors of [32, 33] propose a security framework for a smart home in order to achieve mutual authentication with security features, such as anonymity and perfect toward security. Another article [34] proposes the use of logic-based security algorithms to improve home security. The study classifies the natural APs of a home as primary or secondary depending on their use. Logic-based sensing is implemented by identifying normal user behavior at these APs and requesting user verification when necessary. User position is also considered when various APs change states. Meanwhile, a framework named SoftAuthZ is presented in [35], which proposed a context-sensitive and behavior-based security framework. It incorporates soft security mechanisms, such as belief and confidence, to support authorization decisions. Meanwhile, [36] designs IoT architecture for a smart home, with hardware and software designed according to the system architecture. The hardware part is mainly analyzed in terms of the image recognition module and speech recognition module, while the software part is mainly the optimized algorithm of the accuracy of the surveillance system.

Many smart platforms now exist for IoT, e.g., [5, 6], and are based on a standard design for IoT devices, cloud services, and a proxy gateway. These platforms support IoT devices from prestigious technology companies such as Apple, Samsung, Google, Amazon, Philips, Hue, Azure, and Lifx. However, one of the most significant disadvantages of these platforms is the price and the fact that more than one device is often required. For instance, to control a home remotely, the user must have one or more Apple devices. If the user has an iPhone and an iPad tablet, this is possible, but with only an iPhone, it is not possible to use all of the features because an intermediary Apple device is required. This demonstrates the limitation of controlling devices on a local network. Meanwhile, if using Google or Xiaomi, for example, all control commands are sent to the company's server for processing. Based on observations, a list follows the fundamental shortcomings of most smart homes, along with some possible solutions:

(i) First, the majority of smart home appliances are readily available for purchase on the market, but some are unable to connect to certain systems, such as Apple, Samsung, or Amazon. Most of these systems only connect with devices produced by the same company. However, this study designs a circuit board that is compatible with any kind of smart device or sensor, regardless of the make. Moreover, some other frameworks and systems require monthly payments, unlike the one proposed here

(ii) Second, the price involved in the deployment and setup of a smart home can be excessive. The higher the number of devices and amount of technology that makes up the system, the more expensive it will be, and the more likely it is to fail. Many aspects affect the price, including the size of the home,

product versions, assistance services, and documentation. A primary package for deployment in small homes typically costs at least $1000

(iii) Third, when building and setting up a smart home, it is essential to distinguish between old and new built houses. In the former, both wired and wireless devices usually require time-consuming modifications to be made to enable installation, and in some cases, it may not be possible to install them. Therefore, the proposed system uses Wi-Fi, which is a technology available for use by any house. Furthermore, by applying the SHA-256 algorithm with 32-bit first to avoid brute-force attack, the presented policy enhances security by preventing outside cyber attacks that could infiltrate the communication system

(iv) Finally, another common weakness found in most other systems is that they require the use of sophisticated and specialist tools, which most people either do not have or do not know how to use. In contrast, the presented system is convenient and user-friendly and requires no specialist knowledge or equipment to setup

Based on Tables 1 and 2, this paper has been devised pending past weaknesses. At the same time, it allows for the functionality generally demanded by a smart home.

3. System Architecture of the IoT Smart Light

To build an IoT smart light framework with enhanced security, the system developed in this study has both lighting and security monitoring features that determine the detailed system structure, as shown in Figure 1. The system is designed for use on a website or an application. Each light bulb consists of an ESP8266-12F microcontroller circuit that receives data from the sensor on the circuit and sends that data to the WebServer. The WebServer then displays information on the light bulbs, and depending on the operation of the user, it transmits commands to the ESP8266-12F to adjust the colors of the lights and turn them on or off. In automatic mode, the WebServer uses the data collected from the motion sensor to transmit a command to turn the light on or off. Subsequently, it transmits a warning or a siren alarm if security mode is on. With the functions of the system that will help users easily manage their lighting system, there will be no electricity wastage or unfortunate accidents when forgetting to turn off the lights. Moreover, the system plays the role of security manager for the house. More specifically, users can integrate additional sensors into the system without having to install or configure the device. This study proposes to design and implement policies with motion sensors into the product range and to enable data control by using a smartphone with a web application. The used components are hardware, a server, the web, and a mobile app, as shown in Figure 1.

3.1. IoT Nodes. Figure 2 shows the prototype of the IoT node, including all of its features on the front and back sides. The prototype has a two-layer circuit using 24 LED WS2812B in a spiral-shaped serial with a cooling pad, microcontroller, DAC unit, ESP8266-12F transmission module, power supply unit, and buzzer behind the circuit. This section discusses the measurement system components, from the sensors to the transmitting module. A further breakdown of the prototype parts is presented in Table 3, including operating voltages, operating currents, operating temperatures, and prices.

3.2. Server Gateway. A server is considered to be a computer that is connected to the network and has high processing capability, as shown in Figure 3. The server allows users to install software which in turn allows them to request services and resources. Currently, many servers are available; these include Google virtual servers like Firebase or other free servers. Characteristically, these systems are simple, public, and easy to use and have a static IP system. However, they also have some drawbacks:

(i) Restricted long-term use; fees must be paid to get the best features

(ii) Restricted ability to customize the server, due to expensive subscription services

(iii) Costly system with expensive bills due to dependency on third-party providers

The benefits of building a private server system are as follows:

(i) No limits in terms of server resources, thus increasing storage space and bandwidth and allowing simultaneous access. Therefore, there is no need to share with other users

(ii) The ability to set up and configure the system according to individual needs

(iii) Security capabilities, custom configuration, and private protocol

(iv) Quick remote access for administrators and easy upgrading when required

(v) The possibility to learn and implement more protocols, thus stimulating creativity, and search capabilities

(vi) Multiple users and devices can be signed in at the same time

As stated in our previous work, the best choice for a server is the Raspberry Pi 3+ [42]. A good server is one of the most important factors. However, the servers and virtual servers currently available on the market are often costly to rent or are free to use but do not work effectively.

(i) Only a Raspberry Pi 3+ is needed to build a server, as it can be used with many users and devices

(ii) Easy to program and access because Raspberry's primary operating system is Linux

Table 2: Comparison of principal characteristics of the most relevant security technologies.

Ref.	Hardware design	Sensors	Purpose	Application	Prices	Security
[25]	Digital fingerprints	No	Reduced cost solution for authenticating edge devices	No	>$1000	No
[26]	Intel Galileo board	Temperature	A secure solution for IoTs smart home	ThinkSpeak	>$1000	TBSA
[27]	No	No	Modified hash-chain authentication mechanisms	No	No	Hash train
[28]	No	No	Developed a security framework	No	No	Hash function
[29]	No	No	Established data protection mechanisms	No	NA	Hash function
[30]	No	No	Find the optimal hash function in order to increase security	Contiki OS	No	SHA function
[31]	No	No	Developed a security framework based on ZigBee protocol	Yes	No	AES-128
[32]	No	No	A secure solution for IoTs smart home	No	No	Hash function & XOR
[33]	No	No	A secure solution for IoTs smart home	No	No	Hash function & XOR
[34]	Yes	Yes	Developed a security framework	No	No	Logical sensing
[35]	No	No	Developed a security framework	Yes	No	Soft-security
[36]	Yes	Yes	Developed a security framework based on IoT	No	Low-cost	Stereo matching
Our proposed	Yes	Yes	The design of a smart light system developed on a security framework for IoT smart home with a low-cost and enhanced security	Yes	<$200	SHA-256

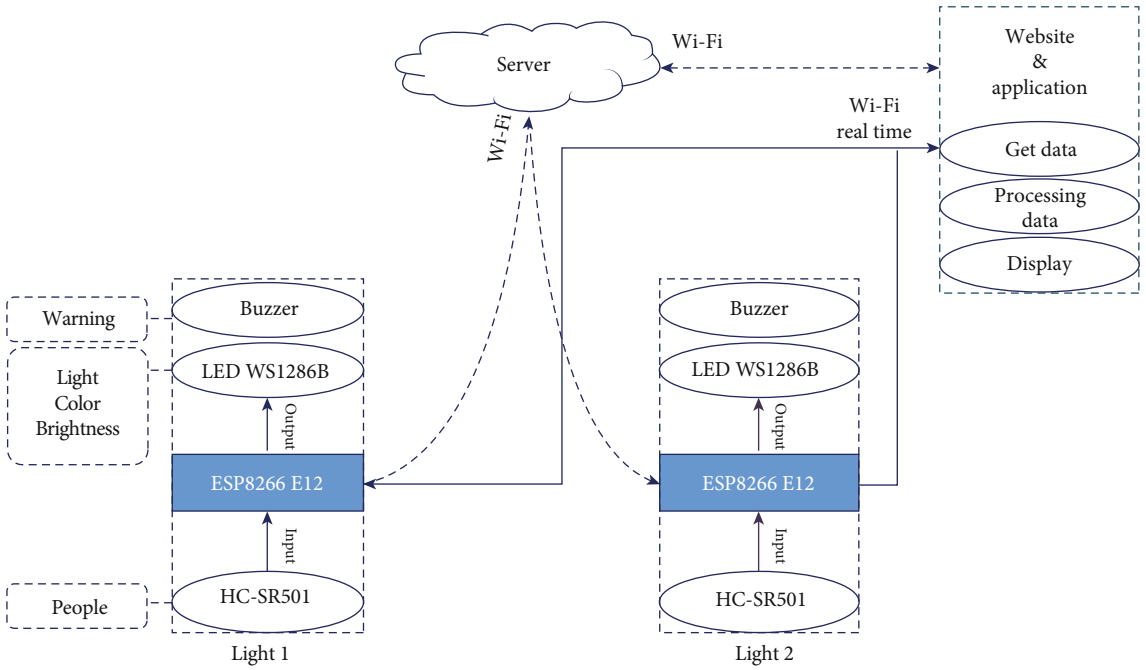

Figure 1: Block diagram of the smart lighting system.

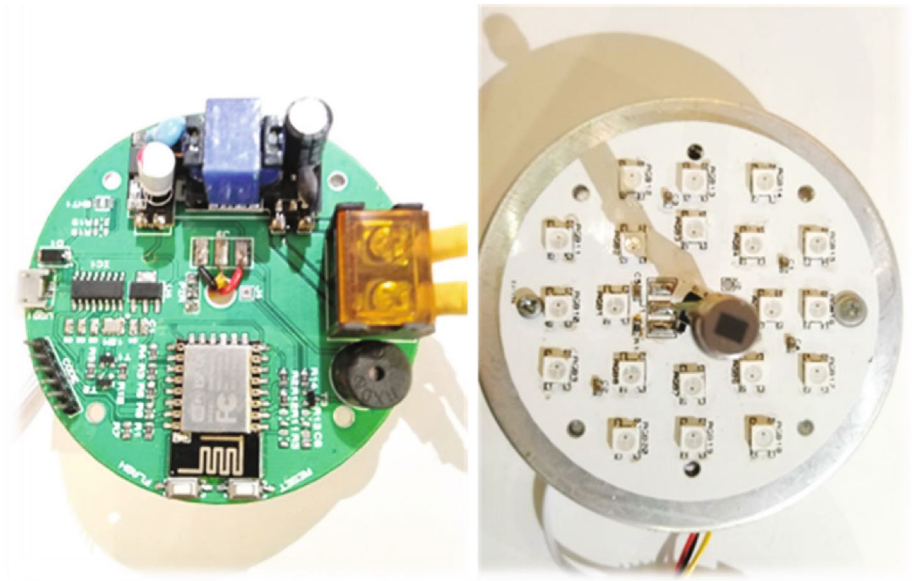

Figure 2: Prototype of the sensor node: (a) the front of the circuit board and (b) the other side of the circuit board.

(iii) Fast network speed supports the latest two bandwidths

(iv) The high speed of the 1.4 GHz quad-core CPU ensures excellent performance for small and medium projects

3.3. Configuration of Devices. The diagram presented in Figure 4 summarizes the main interactions that may be performed step by step in the smart lighting system. Before devices connect to each other, they connect to a server (named self-services soft IP 192.168.4.1—this can be changed) to ask whether they can connect to each other, consequently enabling management of the network capacity. Additionally, the communication parameters are also controlled by the server.

3.4. IoT Smart Light Functionality. Figure 5 shows the application outline of the IoT smart light for a real system in which three components are used: the first being a server, which allows users to connect to the network; the second being devices, characterized by the modules presented in Table 3, which are integrated into the circuit board; the third being users, which control the devices for real-time monitoring and motion detection via Wi-Fi or the Internet. Each

Table 3: The main active components of the smart lighting system.

Ref.	Modules	Name	Amount	Operating current	Operating current	Operating temperature	Prices
[37]	Microcontroller	ESP8266-12F	1	3.3~3.6 V	80 μA	−40°~125°	$2.0
[38]	LED	WS1286b	24	1.8~3.3 V	20 μA	−40°~85°	$1.0
[39]	PIR sensor	D203B PIR	1	3.0~15 V	60 μA	−30°~70°	$1.0
[40]	Sound	Buzzer	1	3.0~8.0 V	22 μA	−40°~85°	$0.1
[41]	Humidity and temperature	SHT31	1	2.4~5.5 V	800 μA	−40°~90°	$2.0

Figure 3: Raspberry Pi 3+ used in the proposed smart lighting system.

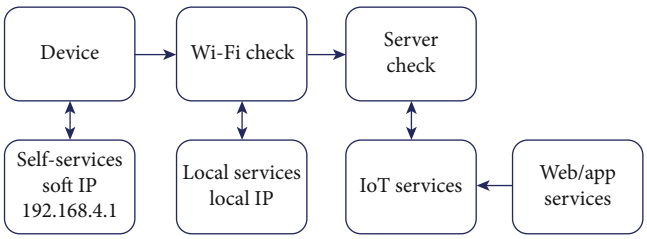

Figure 4: Example of a device configuration.

device can communicate with all of the other devices within an IoT smart lighting system. Seven detailed steps needed for one user to connect to one device are elaborated as follows:

(i) *Initial Configuration*. The server must be configured in such a way that all nodes in the specified network can communicate. This means that it requires an address with a gateway that is always open. Then, a demo initial server configuration can include the following elements, as shown Box 1:

Step 1 (S1). A device connected to the network starts connecting to the server. The server then sends an identifier or token to the device to authenticate the link between the server and the device. In this step, SHA-256 encrypts the ID and passwords.

{'Type': 'EspGreeting',
'Name': ' f0d08d2501e41e2743b057fab38ac7782d4f8652d99c31a04342fece03402fa4', 'Pass': '57d7284f44
32d7b40f03990d8777c34925bdca95183a967a7f50726d83edd5a7'}

Step 2 (S2). The server receives the packet and checks the type, then extracts the database, and encrypts the SHA-256 ID and password to compare the username and password of the received packet. If the packet is correct, the server returns the token to the device.

{'Message': 'Greeting accepted', 'Type': 'EspGreeting', 'Token': '91d9365f88344c1d86546f1d67be083e', 'Update': 0, 'Server': 'Python'}

Step 3 (S3). About 30 seconds after completing the pairing and transferring the token device, the device updates the status on the server to notify it of its operational status.

{'Name': f0d08d2501e41e2743b057fab38ac7782d4f8652d99c31a04342fece03402fa4', 'ToSid': “, 'Token':
'91d9365f88344c1d86546f1d67be083e', 'Type': 'EspPolling', 'EMC': '0',

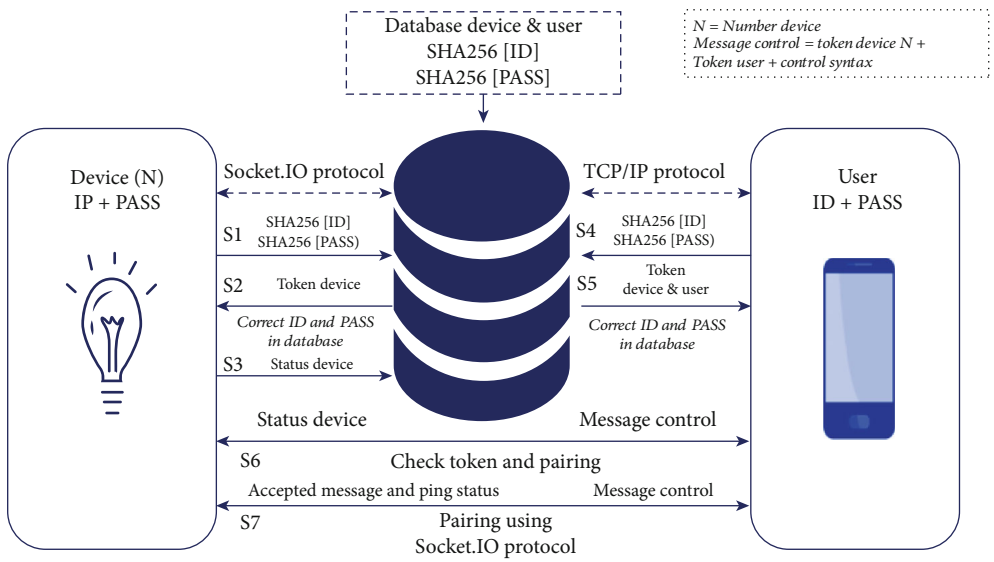

Figure 5: Basic process of a smart lighting system with SHA-256.

```
wsgi starting up on http://0.0.0.0:80
accepted ('14.169.106.97', 52430)
connect 2ee1a6ed939f4d8485441353508d5d19
14.169.106.97 - - [19/Dec/2019 11:38:48] "GET/socket.io/?EIO=3&transport=polling&t=MyTy_ha
HTTP/1.1" 200 349 0.003137
accepted ('14.169.106.97', 52434)
connect d8c1cfd5a7a345309c388e24cee3de64
```

Box 1:

'Humidity': '74.28 \%', 'Temperature': '28.60 ˚C',
'Pir Sensor S1': 'Connect', 'Pir Sensor A1': 'ON',
'Now': 412386, 'Date': 191219, 'SSID': 'P9', 'IP': '192.168.137.135'}

Step 4 (S4). Every time a user accesses the server, the server verifies information with a reconstructed database, and all data transmitted from the user's device to the server is encrypted with SHA-256 security.

{"Type":"DevGreeting",
"Name":" 0a041b9462caa4a31bac3567e0b6e6fd9100787 db2ab433d96f6d178cabfce90",
"PassWord":"8d969eef6ecad3c29a3a629280e686cf0c3f5d 5a86aff3ca12020c923adc6c92"}

Step 5 (S5). After undergoing the authentication process, the server sends the user an identifier called a token user, which is token device *n*, and encloses the token device so that the device can link to the device directly. This experiment uses a two-handled device such as SmartLight_3AF3 and SmartLight_3AF4.

{'Message': 'Server accepted', 'Type': 'DevGreeting', 'Token': 'd8c1cfd5a7a345309c388e24cee3de64
', 'Update': 207, 'Server': 'Python', 'EspSid1': '91d9365f88 344c1d86546f1d67be083e', 'EspName':'SmartLight_3AF3', 'EspSid2': '2ee1a6ed939f4d8485441353508d5d19', 'EspName': 'SmartLight_3AF4'}

Step 6 (S6). The user selects the device and sends a file formatted as follows: "token user+token device *n*+message." It is imperative that the JSON packet includes the above information so that it can be allowed into the real-time environment, where the package is then sent straight to the device.

(i) "Token user+token device 1+message" or

(ii) "Token user+token device 2+message" or

(iii) ""Token user+token device 3+message"

{'Name':'0a041b9462caa4a31bac3567e0b6e6fd9100787d b2ab433d96f6d178cabfce90',
'Token':'d8c1cfd5a7a345309c388e24cee3de64',
'ToEsp':'f0d08d2501e41e2743b057fab38ac7782d4f8652d 99c31a04342fece03402fa4',
'Token':'91d9365f88344c1d86546f1d67be083e',
'Type': 'DevCommand', 'Message': 'LMode', 'EName': 'DirectMes', 'Update': 0, 'V0': 50, 'V1': 50, 'V2': 50,'V3': 50, 'V4': 1, 'V5': 0, 'V6': 0, 'V7': 0, 'V8': 0, 'V9': 0, 'V10': 0, 'V11': 0, 'V12': 0, 'V13': 0, 'V14': 0, 'V15':0}

Step 7 (S7). The device sends back the message from the user. The result is then sent in the JSON file format, starting with

"token user+token device" to the device SmartLight_3AF3 and SmartLight_3AF4.

{'Name': 'f0d08d2501e41e2743b057fab38ac7782d4f8652d99c31a04342fece03402fa4',
'Token':'57d7284f4432d7b40f03990d8777c34925bdca95183a967a7f50726d83edd5a7',
'ToSid':'0a041b9462caa4a31bac3567e0b6e6fd9100787db2ab433d96f6d178cabfce90',
'Token':'8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92',
'Type': 'EspPolling', 'EMC': '0','Humidity': '74.28 \%', 'Temperature': '28.60 °C','Pir Sensor S1': 'Connect', 'Pir Sensor A1': 'ON','V0': 50, 'V1': 50, 'V2': 50, 'V3': 50, 'V4': 1, 'V5': 0, 'V6': 0, 'V7': 0, 'V8': 0, 'V9':0, 'V10': 0, 'V11': 0, 'V12': 0, 'V13': 0, 'V14': 0, 'V15': 0,'Update': 400}
{'Name': 'ded4f806888f56a3a948a16faace4471af877a02f68687fb50471b395bc90301',
'Token':'2df35259cbb0b4075023fe429e30455888dfadb14065baadeedeef06a08db93a ',
'ToSid':'0a041b9462caa4a31bac3567e0b6e6fd9100787d-b2ab433d96f6d178cabfce90',
'Token':'8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92',
'Type': 'EspPolling', 'EMC': '0','Humidity': '74.28 \%', 'Temperature': '28.60 °C','Pir Sensor S1': 'Connect','Pir Sensor A1':'ON','V0': 50,'V1': 50, 'V2': 50, 'V3': 50, 'V4': 1, 'V5': 0, 'V6': 0, 'V7': 0, 'V8': 0, 'V9': 0, 'V10': 0, 'V11': 0, 'V12': 0, 'V13': 0, 'V14': 0, 'V15': 0,'Update': 400}

This study makes the interactions between multiple users and multiple devices more secure by using a server based on SHA-256. Figure 6 describes the system structure of the improved process, including the *database*, the *user*, the *invention*, the *services*, and the *real time*. The features of the module functionalities are elaborated in the following.

(i) *Server*. The general model of the server is included (user, service, real time, and database)

(1) The user block consists of a user data validation package, which the server trusts and uses for authentication, receiving, and recording of user tokens and token devices

(2) The service division provides data transfer services, web services, and an app for users

(3) The real-time unit is a data transmission environment, which acts as the bridge from the device to the user

(4) The database block contains the user's information and the password with which the server accesses and retrieves data for authentication

(ii) *Database*. The IoT creates large amounts of data, including streaming data, time-series data, user ID, passwords, and sensory data. Efficient management of this data requires the use of a database. However, the powerful generation of IoT data requires a separate variety of database. This research applies TinyDB, which is a lightweight NoSQL engine that can be used to store structured data. It also supports storing data as JSON files on the server or stores the JSON data in its memory for faster access times

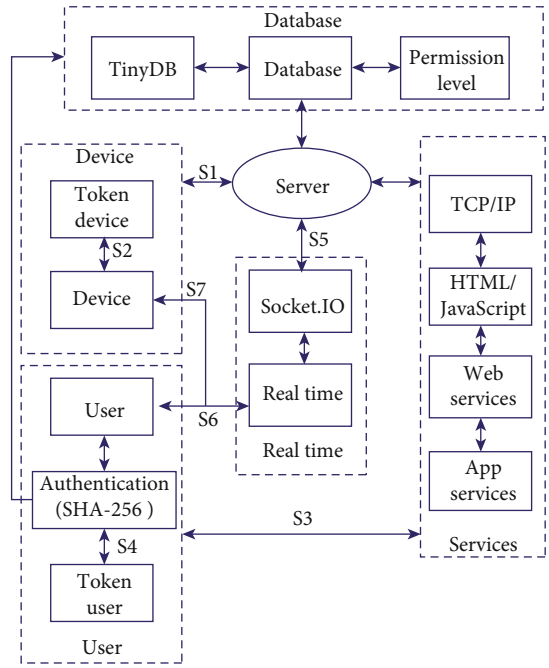

Figure 6: Security framework for the smart lighting system using SHA-256.

(iii) *Users*. The name SHA-256 stands for "256-bit Secure Hash Algorithm" and is applied for cryptographic security

(1) The SHA-256 algorithm is a foolproof algorithm [43, 44]

(2) SHA-256 creates bitcoin addresses to improve safety and security

(3) The cryptographic hash algorithm creates unique and irreversible hashes. The higher the number of possible hash functions, the lower the probability that two values will produce the same hash value. A 32-bit SHA-256 is first applied to avoid brute-force attacks

(4) Hashing with SHA-256 applied in this article is presented in Algorithm 1 and described in Table 4

(iv) *Services*. The service block includes data transfer and uses a web application platform for users

(v) *Real time*. Modern web applications have evolved and now differ considerably from when they first appeared, with many new techniques and abilities that deliver fresh, exciting experiences. They are also handy for users. Real-time web technology is becoming increasingly popular. Some techniques and

Input $\mathcal{I} = \{M, L, P\}$
Output $\mathcal{O} = \{H_t\}$
  for i $i = 1$ to N **do**
    Find $\mathcal{BW}(t)$
      if $t = 0$ to 63 **do**
        if $(t < 15)$ **then**
          $\mathcal{BW}(t) = \mathcal{M}(i(t))$
        else if (t >16) **then**
          Calculate $\sigma_0 = ROTR^7(t) \oplus ROTR^{18}(t) \oplus SHR^3(t)$
          Calculate $\sigma_1 = ROTR^{17}(t) \oplus ROTR^{19}(t) \oplus SHR^{10}(t)$
          Calculate $\mathcal{BW}(t) = \sigma_1(\mathcal{BW}(t-2)) + \mathcal{BW}(t-7) + \sigma_0(\mathcal{BW}(t-15)) + \mathcal{BW}(t-16)$
        end if
        Set $\{a = H_0(i-1), b = H_1(i-1), c = H_2(i-1), d = H_3(i-1), e = H_4(i-1), f = H_5(i-1), g = H_6(i-1), h = H_7(i-1)\}$ in [43]
        for $t = 0$ to 63 **do**
          Calculate $\mathcal{T}_1 = \text{h} + \sum_1(e) + ((e \wedge f) \oplus (e \wedge g)) + \text{C(t)} + \text{BW(t)}$
          Calculate $\mathcal{T}_2 = \text{h} + \sum_0(a) + ((a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c))$
          Set $\{h = g, g = f, f = e, e = d + T_1, d = c, c = b, b = a, a = \mathcal{T}_1 + \mathcal{T}_2\}$
        end for
        Calculate the i^{th} intermediate $H(i)$
        for $t = 0$ to 7 **do**
          $\mathcal{A}(a, b, c, d, e, f, g, h)$
          $\iota = \mathcal{A}(t)$
          $H_t(i) = \iota + H_t(i-1)$
      end for
    end for

Algorithm 1: Hashing with SHA-256.

Table 4: Simulation/numerical parameters.

Variable	Description
$A = \{a, b, c, d, e, f, g, h\}$	Number of bit words is used to compute the hash values
$\mathcal{H}_i = \{\mathcal{H}_i \mid i \in (1, N)\}$	Hash value i
$\mathcal{H}_j = \{\mathcal{H}_j \mid i \in (1, N)\}$	j word of hash value i
t	Iteration t of the hash computation
$\mathcal{C}_i = \{C_i \mid i \in (1, N)\}$	Value to be used for the iteration t of the hash computation i
k	Number of appended to a message
l	Length of the message M
M	Message to be hashed i
m	Number of bits in a message block
$\mathcal{M}_i = \{\mathcal{M}_i \mid i \in (1, N)\}$	Set of message block i, with a size of m bits i
$\mathcal{M}_j = \{\mathcal{M}_j \mid j \in (1, N)\}$	The j word of the i message block
n	Number of bits to be rotated or shifted when a word is operated upon
N	Number of blocks in the padded message
T	Temporary w-bit word used in the hash computation
$\mathcal{BW}$	Number of bits in a word
$\mathcal{BW}_i = \{\mathcal{BW}_i \mid i \in (1, N)\}$	Number of bits word for the message schedule

methods help to build real-time applications such as Ajax long, polling, server-sent events (SSE), Comet, and WebSocket. It should be mentioned that WebSocket, with the support of HTML, is becoming dominant. The research team is using WebSocket with the Socket.IO library. WebSocket is a protocol that supports two-way transfer data between the server and the client over a single TCP connection. Moreover,

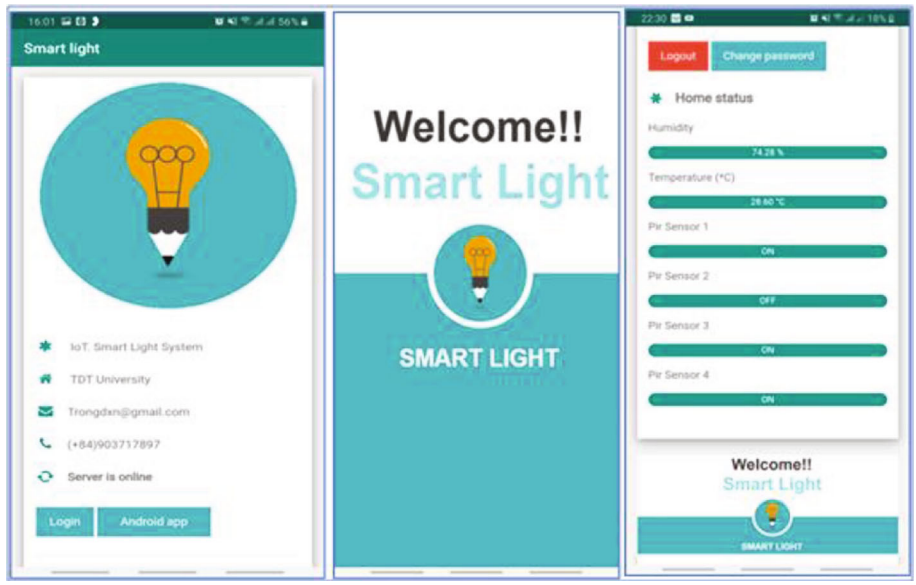

Figure 7: Screenshot of a mobile application that controls the IoT Smart Lighting system.

Table 5: Simulation parameter settings.

#	Parameters	Values
1	ESP8266 protocol	SPI protocol (CS, SCLK, MOSI, MISO)
2	Flash memory	64 MB
3	Frequency MCU	80-160 MHz, 32-bit micro MCU
4	Channel mode	I2SCONF CHAN
5	SRAM size	36 kB
6	Baud rate	110-460800 bps
7	Security standard	WPA/WPA2
8	Multiplexed	UART
9	SPI	30 pins (PWM, I2C, 1-wire)

WebSocket is designed to transfer data using port 80 and port 443 and is part of HTML5. Because WebSocket can operate on regular web gates, there is no need to go through the hassle of opening gates for applications or to worry about being blocked by firewalls or proxy servers. Socket.IO is a library used for mobile and web applications to develop real-time applications. With its robust and easy-to-use features, Socket.IO is increasingly used for social networking sites that require high interaction with blogs or e-commerce websites. With this library, working with WebSocket becomes much more straightforward

4. Building a Smart Home by Using the Smart Lighting System

This section presents the configuration setup for the initial configuration and demo frameworks used for the system.

Table 6: Comparison of the packet delivery success rate based on distance (m) and RSSI (dB).

#	Distance	RSSI	Packets sent	Packets loss	Error rate
1	$d < 5$ m	-58 dB	100	0	0%
2	5 m $< d <$ 10 m	-65 dB	100	0	0%
3	10 m $< d <$ 5 m	-74 dB	100	5	0%
4	$d > 15$ m	-80 dB	100	40	40%~50%

4.1. Initial Configuration. Various modules must be configured before the IoT smart light begins operating. Firstly, the IoT nodes with five modules and one lamp need to be connected to the circuit board carefully. Then, they must be connected with the local network throughout the server. The network parameters of each IoT node then need to be configured manually through a web application to connect to WebSocket. For this purpose, the Raspberry Pi 3+ is set to the server gateway, so that it can be accessed with a regular web application through the self-service soft IP address 192.168.4.1. Following the self-service, the IoT nodes can be connected to Wi-Fi, where a web interface allows definition of the Wi-Fi status. For simplicity, the server is configured to assign port forwarding+DNSS to Web HTML. Port forwarding is the process of forwarding a specific port from one network to another. This allows external users to easily access the internal intranet via a dynamic DNS router (dynamic DNS) that provides a particular program which runs on the user's computer translating service. This program monitors IP address changes on the host's computer and contacts the DNS system whenever the host's IP address (which is provided by the ISP by the dynamic method) changes. The program then updates the DNS database with information about the change of that address. In this way, even though the server is continuously changing its address, the domain name is appointed to the new IP address by the

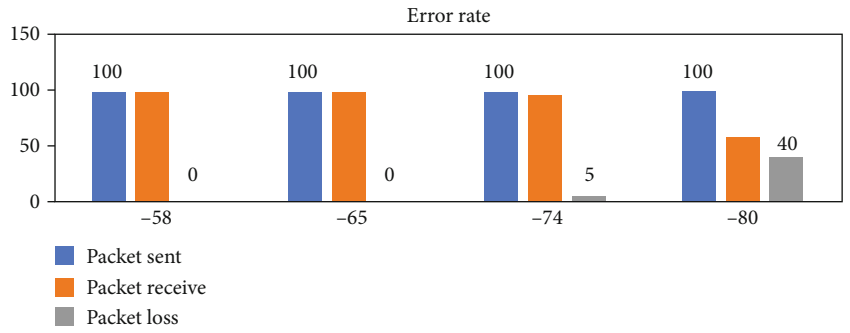

Figure 8: A bar chart of the packet versus the RSSI (dB).

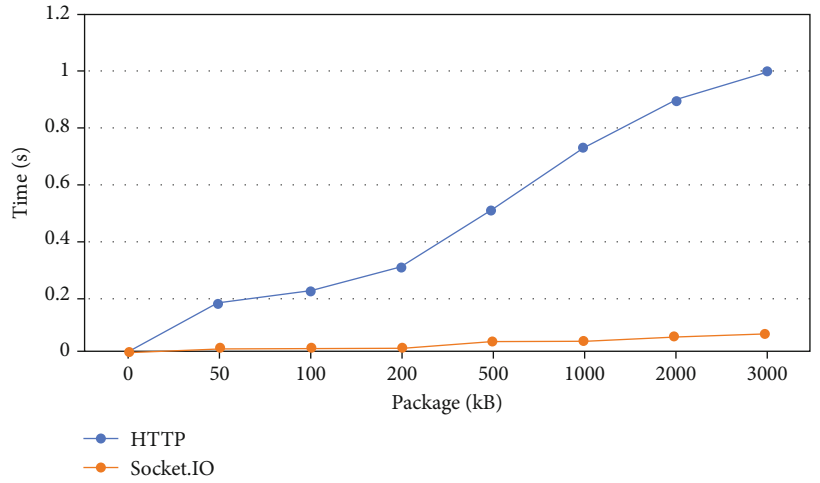

Figure 9: The impact of Wi-Fi throughput variation on the command execution response time.

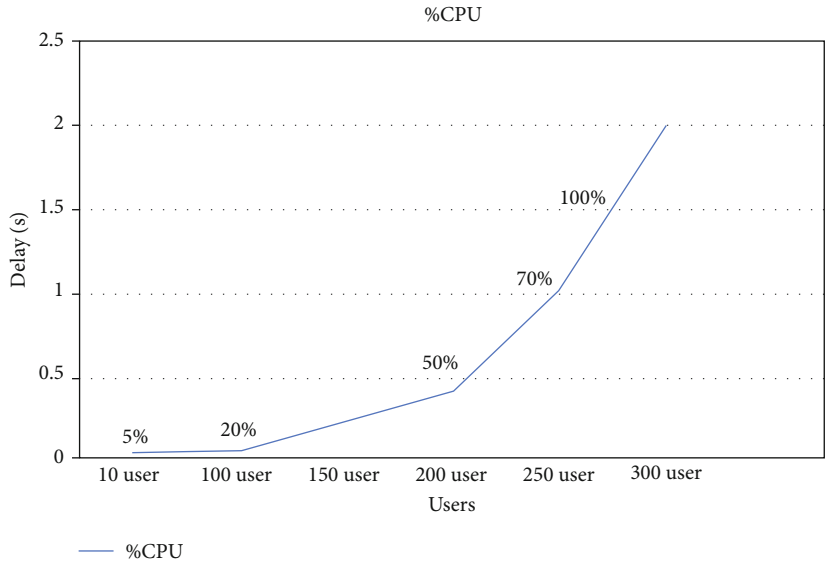

Figure 10: A chart comparing the number of users with the CPU performance (%) and time delay (s).

1DNS server system. A dynamic domain address, DDNS, leads to the home internet modem and is programmed to continuously update the modem's dynamic IP address. Finally, it is worth considering that sensors should be calibrated before operation (and periodically during use) to maintain the accuracy of data:

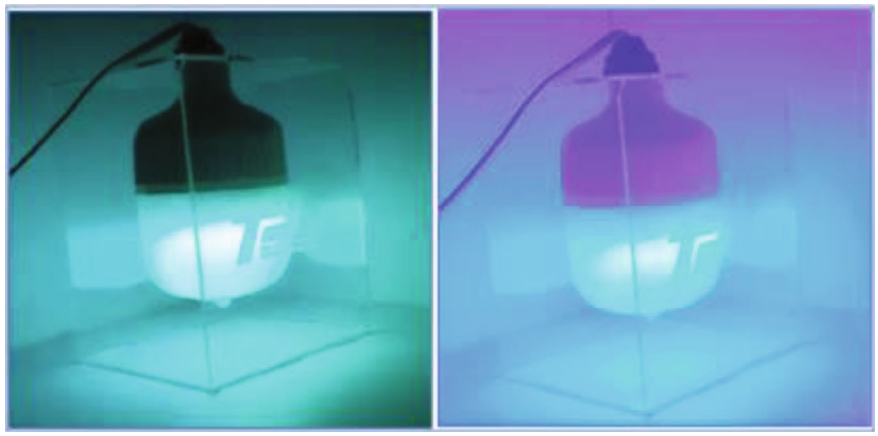

Figure 11: A final prototype of the IoT smart light system.

(i) *PIR Sensor D203B PIR*. The pyroelectric passive infrared (PIR) sensor is digital and regularly used in daily life. Moreover, its measurement procedure is very straightforward, as active sensors can detect a change in the environment when a transmitted signal is disturbed

(ii) *Humidity and Temperature SHT31*. This system uses an industrial temperature thermometer. It is not necessary to calibrate the entire operating range, and temperatures between 10°C and 40°C will not activate any alarm. However, if the temperature falls below 10°C or rises above 40°C, an alarm will be activated

Once such a configuration is finished, the IoT node connects to the server gateway and begins to interact with the network. The status of the connection is displayed on the web application.

4.2. Software Configuration. After preparing the module shown in Table 3, all of the parts embedded in the circuit board are placed in a real environment to perform different experiments, as described in Figure 2. The other essential feature is the software configuration. The use of open source is not necessary to guarantee that the framework can be proactive and encrypt the design of the structure. The use of open-source code results in our system, which is easy to manage through the attractive application shown in Figure 7, which uses multiple plugins to make home automation tasks fully autonomous. The web application is designed by HTML, JavaScript, and CSS programming with integrated real-time technology, which allows users to control the system anytime and anywhere. Almost no related research into IoT smart homes has considered the use of security technologies, even in commercial operations. This is because the resource-constrained device is usually not powerful enough to handle secure communication protocols. Therefore, in the case of this system, we applied SHA-256 to enhance security for an IoT smart home, as shown in Figure 6.

5. IoT Smart Light System Evaluation

5.1. Packet Delivery Success Rates. To verify and confirm the reliability and stability of the IoT smart light system, this section of the article focuses on the ESP 8266-12F packet delivery success rates over various distances. The wireless communication link response time is determined as the time needed for the IoT smart light to complete the task of sending a command to ESP 8266-12F devices and receive a response from the corresponding device. Table 5 lists various ways to verify an existing Wi-Fi network. All commands are sent and received using the TCP method to ensure that the controls are transferred to the correct place. The first value shown in Table 6 represents the distance from the server to the IoT node varying the RSSI value from -58 dB to -80 dB and displays the error rate of packets sent and packets lost. Therefore, if the distance is greater, the error rates increase. It is also worth considering that the outcomes and results presented in Figure 8 were collected from regular traffic in an IoT smart home system. The results indicate that the operating frequency should be carefully considered to enhance and maximize the packet delivery success rate.

5.2. Wi-Fi Throughput Variation on the Command Execution Response Time. The chart presented in Figure 9 shows the response time based on the size of the packet transmitted, which is sent from a handheld device to the IoT AP through the Wi-Fi network with a variation of Wi-Fi throughputs. In this figure, the Wi-Fi throughput is varied from 0 to 3000 kB/s. The horizontal axis indicates the variation of Wi-Fi throughputs, while the vertical axis represents the sum of HTTP response time and Socket.IO response time. With Wi-Fi throughput ranging between 0 kBps and 3000 kBps, the total Wi-Fi response time has minimal impact. Because of the system in use, front end: using HTTP protocol on TCP/IP platform, since there is transmission error checking between transmitting packets, high latency will occur back end: use the Socket.IO protocol to transmit data extremely fast, because between no transmission packet error check was sent during transmission.

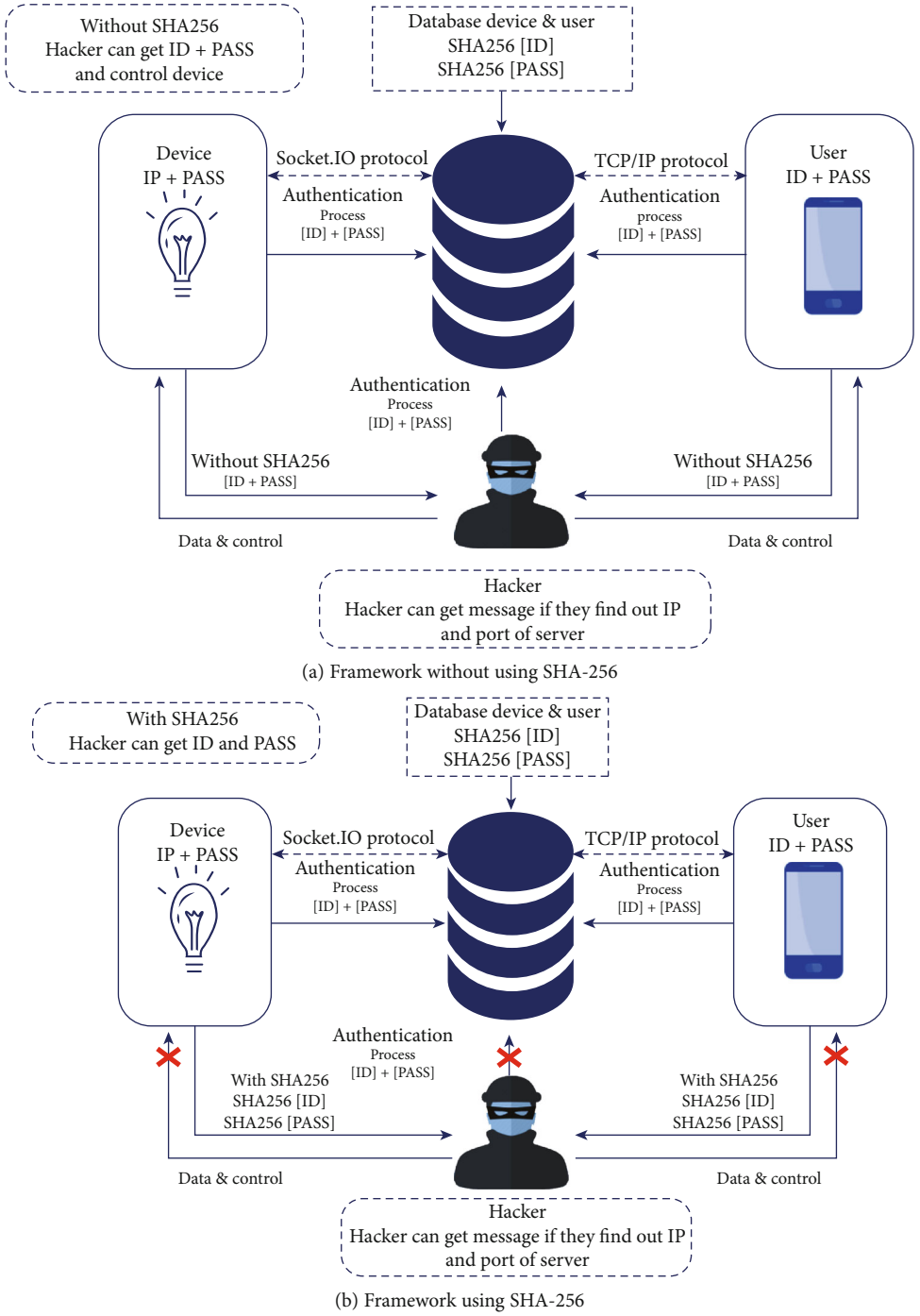

Figure 12: Comparison of the authentication process with and without SHA-256.

5.3. Variation in the Percentage of CPU (%CPU). Figure 10 depicts user variation with the percentage of CPU (%CPU), where the %CPU chart displays virtual machine CPU usage and values. The system can also be used with multiple users at one time. Evaluation of the system performance has a significant function in the IoT system, since dozens or even hundreds of devices can be connected to a server system at the same time. Therefore, it is necessary to evaluate the performance in practice to determine stability and reliability.

5.4. Design and Implementation of an IoT Smart Light. This subsection describes the implementation of an interface with the IoT connection. The mobile application is used to manage switching on and off of the device by users. This application has two modes: automatic and manual. The automatic mode is activated when IoT devices detect a change in the coverage of the sensors. Moreover, users can take over control of the device and turn the system on or off with the mobile application, as shown in Figure 11.

5.5. Comparison of the Authentication Process with and without SHA-256. In a smart home environment like the ones discussed in this article, smart devices transmit data across a wireless network. A variety of data is transferred between the devices, and such data can leak if an unauthorized device accesses, or hacks into, the smart home environment. Due to this potential breach of security, smart devices in the environment implement an authentication rule throughout an authentication module. To highlight the effectiveness of using SHA-256, this article presents the advantages of using it, compared to not using it. As shown in Figure 12, hackers can obtain information regarding the network configuration if they are successful in finding the IP address of the server. This could be very dangerous for all concerned parties, because they could take control of smart devices in the house, and the house would become insecure. The use of SHA-256 brings many advantages when attempting to make a smart home as safe and secure as possible. This is beneficial, because we aim to provide a strong security framework for all devices.

6. Conclusions

Numerous IoT techniques have been installed in smart homes to improve homeowners' quality of life. In this context, an excellent asset for a smart home is proposed in this study. We have designed and implemented a system to control the home, which has three parts: hardware, a server with high security, and a web application. The IoT node hardware was designed for real-life testing, and to receive IoT information from any device. A server was designed and implemented to control the IoT nodes in the system. Finally, an application for use anytime or anywhere on a smartphone or web browser via a Wi-Fi communication connection link was built to control the IoT smart system in real time. This application permits both automatic and manual functional control, which is flexible for the user. The advanced IoT system was installed at Ton Duc Thang University, Vietnam. The results showed clear potential benefits for a smart home, including robust security and low cost. Above all, this study is aimed at demonstrating the great potential that all digital technology holds for smart homes.

Abbreviations

IoT: Internet of Things
SHA: Secure Hash Algorithm
WLAN: Wireless Local Area Network
4G: Fourth generation
HTML: HyperText Markup Language
PC: Personal computer
Wi-Fi: Wireless Fidelity
MQTT: MQ Telemetry Transport
AP: Access Point
ID: Identifier
OTP: One-time password
IP: Internet Protocol
CPU: Central processing unit
NoSQL: Not only SQL
TCP: Transmission control protocol
DDNS: Dynamic DNS
ISP: Internet service provider
PIR: Pyroelectric passive infrared
CSS: Cascading style sheets
RSSI: Received signal strength indicator.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Faculty of Electrical and Electronics Engineering, Ton Duc Thang University.

References

[1] "IoT big data security and privacy versus innovation," https://en.wikipedia.org/wiki/Smartcity.

[2] K. R. Sollins, "IoT big data security and privacy versus innovation," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1628–1635, 2019.

[3] R. Petrolo, V. Loscrì, and N. Mitton, "Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 1, p. e2931, 2017.

[4] Z. B. Celik, E. Fernandes, E. Pauley, G. Tan, and P. McDaniel, "Program analysis of commodity IoT applications for security and Privacy," *ACM Computing Surveys*, vol. 52, no. 4, pp. 1–30, 2019.

[5] Y. Meng, W. Zhang, H. Zhu, and X. S. Shen, "Securing consumer IoT in the smart home: architecture, challenges, and countermeasures," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 53–59, 2018.

[6] N. Panwar, S. Sharma, S. Mehrotra, Ł. Krzywiecki, and N. Venkatasubramanian, "Smart Home Survey on Security and Privacy," https://arxiv.org/abs/1904.05476.

[7] "Security and resilience of smart home environments," *European union agency for network and information security*https://www.enisa.europa.eu/publications/security-resilience-good-practices.

[8] P. Rajiv, R. Raj, and M. Chandra, "Email based remote access and surveillance system for smart home infrastructure," *Perspectives in Science*, vol. 8, pp. 459–461, 2016.

[9] E. Fernandes, J. Jung, and A. Prakash, "Analysis of emerging smart home applications," in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 636–654, San Jose, CA, USA, 2016.

[10] A. Jacobsson, M. Boldt, and B. Carlsson, "A risk analysis of a smart home automation system," *Future Generation Computer Systems*, vol. 56, pp. 719–733, 2016.

[11] K. Markantonakis, R. N. Akram, and M. G. Msgna, "Secure and trusted application execution on embedded devices," in *Innovative Security Solutions for Information Technology and Communications*, vol. 9522, Springer, 2015.

[12] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and secure session-key establishment scheme in smart home environments," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 254–264, 2016.

[13] S. Madakam and H. Date, "Security mechanisms for connectivity of smart devices in the internet of things," in *Connectivity Frameworks for Smart Devices*, vol. 1 of Computer Communications and Networks, , pp. 23–41, Springer, 2016.

[14] H. Lee, C. R. Ahn, N. Choi, T. Kim, and H. Lee, "The Effects of Housing Environments on the Performance of Activity-Recognition Systems Using Wi-Fi Channel State Information: An Exploratory Study," *Sensors*, vol. 19, no. 5, p. 983, 2019.

[15] S. Zhihua, "Design of smart home system based on ZigBee," in *2016 International Conference on Robots & Intelligent System (ICRIS)*, pp. 167–170, Zhangjiajie, China, 2016.

[16] M. Bassoli, V. Bianchi, and I. De Munari, "A plug and play IoT Wi-Fi smart home system for human monitoring," *Electronics*, vol. 7, no. 9, p. 200, 2018.

[17] G. V. Vivek and M. P. Sunil, "Enabling IOT services using WIFI - ZigBee gateway for a home automation system," in *2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pp. 77–80, Kolkata, India, 2015.

[18] Y. A. N. Wenbo, W. A. N. G. Quanyu, and G. A. O. Zhenwei, "Smart home implementation based on internet and WiFi technology," in *2015 34th Chinese Control Conference (CCC)*, pp. 9072–9077, Hangzhou, China, 2015.

[19] A. Bhatt and J. Patoliya, "Cost effective digitization of home appliances for home automation with low-power WiFi devices," in *2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pp. 643–648, Chennai, India, 2016.

[20] R. Kodali and S. R. Soratkal, "MQTT based home automation system using ESP8266," in *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pp. 1–5, Agra, India, 2016.

[21] I. Froiz-Míguez, T. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, "Design, implementation and practical evaluation of an IoT home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes," *Sensors*, vol. 18, no. 8, p. 2660, 2018.

[22] W.-T. Sung and J.-S. Lin, "Design and implementation of a smart LED lighting system using a self adaptive weighted data fusion algorithm," *Sensors*, vol. 13, no. 12, pp. 16915–16939, 2013.

[23] C.-Y. Chang, C.-H. Kuo, J.-C. Chen, and T.-C. Wang, "Design and implementation of an IoT access point for smart home," *Applied Sciences*, vol. 5, no. 4, pp. 1882–1903, 2015.

[24] Y. Amri and M. A. Setiawan, "Design and improving smart home concept with the internet of things concept using RaspberryPi and NodeMCU," *IOP Conference Series: Materials Science and Engineering*, vol. 325, article 012021, 2018.

[25] U. Guin, A. Singh, M. Alam, J. Canedo, and A. Skjellum, "A secure low-cost edge device authentication scheme for the internet of things," in *31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID)*, pp. 85–90, Pune, India, 2018.

[26] S. Pirbhulal, H. Zhang, M. E Alahi et al., "A novel secure IoT-based smart home automation system using a wireless sensor network," *Sensors*, vol. 17, no. 12, p. 69, 2017.

[27] A. Pinto and R. Costa, "Hash-chain based authentication for IoT devices and RESTWeb-services," in *Ambient Intelligence-Software and Applications – 7th International Symposium on Ambient Intelligence (ISAmI 2016)*, pp. 189–196, 2016.

[28] W. M. Kang, S. Y. Moon, and J. H. Park, "An enhanced security framework for home appliances in smart home," *Human-centric Computing and Information Sciences*, vol. 7, no. 1, 2017.

[29] C. Lachner and S. Dustdar, "A performance evaluation of data protection mechanisms for resource constrained IoT devices," in *2019 IEEE International Conference on Fog Computing (ICFC)*, pp. 47–52, Prague, Czech Republic, Czech Republic, 2019.

[30] D. Halabi, S. Hamdan, and S. Almajali, "Enhance the security in smart home applications based on IOT-CoAP protocol," in *2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, pp. 81–85, Beirut, Lebanon, 2018.

[31] S. M. S. Rana, M. A. Halim, and M. H. Kabir, "Design and implementation of a security improvement framework of Zigbee network for intelligent monitoring in IoT platform," *Applied Sciences*, vol. 8, no. 11, p. 2305, 2018.

[32] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 391–406, 2020.

[33] Q. Lyu, N. Zheng, H. Liu, C. Gao, S. Chen, and J. Liu, "Remotely access "my" smart home in private: an anti-tracking authentication and key agreement scheme," *IEEE Access*, vol. 7, pp. 41835–41851, 2019.

[34] A. C. Jose and R. Malekian, "Improving Smart Home Security: Integrating Logical Sensing Into Smart Home," *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4269–4286, 2017.

[35] N. Ghosh, S. Chandra, V. Sachidananda, and Y. Elovici, "SoftAuthZ: A Context-Aware, Behavior-Based Authorization Framework for Home IoT," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10773–10785, 2019.

[36] A. Yang, C. Zhang, Y. Chen, Y. Zhuansun, and H. Liu, "Security and privacy of smart home systems based on the internet of things and stereo matching algorithms," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2521–2530, 2020.

[37] http://www.ai-thinker.com.

[38] http://www1.futureelectronics.com/doc/EVERLIGHT%C2%A0/334-15__T1C1-4WYA.pdf.

[39] http://www.alselectro.com/files/PIR.pdf.

[40] https://download.mikroe.com/documents/add-on-boards/click/buzz/buzz-click-manual-v100.pdf.

[41] https://www.mouser.com/datasheet/2/682/Sensirion_Humidity_Sensors_SHT3x_Datasheet_digital-971521.pdf.

[42] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[43] https://medium.com/biffures/part-5-hashing-with-sha-256-4c2afc191c40.

[44] https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.