System Approach of Smart Home Implementation with Cybersecurity Elements

Ivana Bridova[1], Peter Brida[2], Michal Janovec[1]

Abstract— **This article presents a systematic methodology for the design and secure implementation of smart homes, emphasizing structured steps from planning and analysis to deployment. The proposed methodology prioritizes cybersecurity, system flexibility, and usability to meet evolving user needs. The study demonstrates how centralized management, process automation, and open-source technologies can enhance user comfort, optimize energy consumption, and reduce operational costs. Special attention is given to the integration of various communication protocols, detailed security measures, and testing processes, ensuring reliable, scalable, and resilient smart home solutions.**

I. Introduction

Home automation fundamentally transforms how households are managed and perceived, bringing greater comfort, time savings, and more efficient use of technology. Smart homes offer many possibilities, from simple device control using voice assistants and virtual agents to advanced automation systems that optimize energy efficiency and enhance security.

The growing complexity of intelligent systems drives demand for intuitive and centralized control. Users expect solutions that ensure easy management of multiple devices while minimizing concerns about their usage.

A systematic approach to designing smart homes is crucial, as it considers all aspects of development and supports the creation of an efficient, eco-friendly, and easily operable smart home that meets current user needs.

Security is an integral part of smart homes. Combining a systematic approach with thorough user research enables the creation of personalized and reliable solutions that accurately reflect users' needs and lifestyles. Smart homes thus represent a seamless blend of technology, comfort, and security for modern living.

The goal of our research was to apply a systematic approach and develop a methodology for creating a secure smart home.

Having introduced the subject and defined the objective of this work, the remainder of this paper will address the following points. Section II provides an overview of perspectives and solutions published by other authors. Section III focuses on analyzing smart home system implementations through open-source platforms. Section IV describes the individual components of the automation system in detail, while Section V explores support for various communication protocols. Section VI is dedicated to system testing. Section VII addresses security aspects in the smart home environment. Section VIII presents the proposed methodology for solving the issue. Section IX concludes the article with an overall evaluation of the results.

II. Related Works

The topic of automation has become the focus of numerous studies aimed at implementing intelligent devices across various environments and for diverse purposes [1], [2], [3]. Smart homes, in particular, enhance both security and comfort by utilizing sensors and communication devices [4], [5].

The market currently offers a wide range of commercial and open-source solutions [6], although many commercial options rely on proprietary sensors and communication protocols, which can restrict compatibility with third-party devices [7]. In contrast, open-source platforms typically support a broader array of devices and protocols through plugins or extension modules.

A significant number of projects aim to integrate smart homes with security elements such as motion detectors, cameras, or smoke and gas sensors [1], [4], [5]. The availability of open-source home automation systems has expanded rapidly in recent years, each offering unique features and facing specific limitations, which makes the selection process critical, as inappropriate choices can lead to substantial costs [6].

Within the domain of open-source automation platforms, the frequent adoption of the Home Assistant platform underlines its widespread popularity and recognized potential [1], [4], [6], [7].

An analysis of recent research indicates that many authors are actively investigating the development and deployment of smart home solutions based on open-source platforms.

III. Analysis of smart home implementations through open-source platforms

Numerous open-source platforms for home automation enable the integration and management of various devices and systems. A comparison of the most well-known open-source home automation platforms is presented in Table I. Platforms that are commercially used, difficult to extend, or complex to operate are excluded from the table.

[1] Authors are with Faculty of Management Science and Informatics, University of Zilina, Univerzitna 8215/1, 010 26 Zilina, Slovakia (corresponding author to provide phone: +421-41-513-7760; e-mail: ivana.bridova@uniza.sk), (e-mail: janovec5@stud.uniza.sk)

[2] Author is with Faculty of Electrical Engineering and Information Technology, University of Zilina, Univerzitna 8215/1, 010 26 Zilina, Slovakia (e-mail: peter.brida@uniza.sk)

TABLE I.
COMPARISON OF SMART HOME PLATFORMS

Platform	Language and Environment	Device and Protocol Support	Flexibility and Expandability	Community and Support	Special Features
Home Assistant	Python, Web Interface	Extensive support (Zigbee, Z-Wave, MQTT, Wi-Fi)	High, customizable with automation options	Strong community, regular updates	AI-based automation, numerous integrations
OpenHAB	Java, Web/Desktop Interface	Extensive support (Zigbee, Z-Wave, MQTT, Modbus)	Highly flexible but complex configuration	Active community, numerous plugins	Advanced user interface (complex configuration)
IoBroker	JavaScript, Web Interface	Extensive protocol and device support	High, easy extension and integration	Growing community, multi-language support	Multi-user and multi-server support
Mozilla WebThings	Node.js, Web Interface	Basic protocol support, primarily Wi-Fi	Less flexibility, suitable for developers	Limited support	Focused on security and privacy
OpenHaus	Node.js, Web Interface	Wi-Fi and MQTT support	Flexible, limited device support	Small community	Modular system, suitable for smaller systems
Calaos	C++, Web and Mobile Interface	Basic protocol support (Zigbee, Z-Wave, EnOcean)	Medium flexibility, limited protocol support	Smaller community	Quick implementation, commercial use

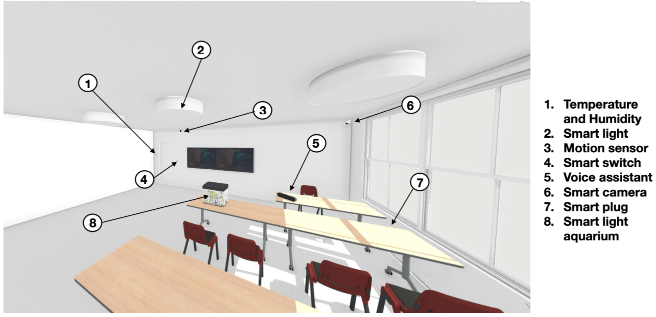

Figure 1. Placement of sensors

For the purpose of our work, we have chosen the Home Assistant platform because of its:

- Simplicity and user-friendliness with a clear web interface for managing devices and automation.
- Broad compatibility with devices and protocols, allowing the integration of various products into a unified platform.
- Regular updates and community support to ensure security and access to the latest features.
- Scripting capabilities for complex automation.
- Local data storage for enhanced security and privacy protection.
- Integration of artificial intelligence for more intuitive control.

When selecting the platform, user needs, device compatibility, and the level of technical expertise were considered.

IV. ELEMENTS OF AN AUTOMATION SYSTEM

The automation system was designed according to the needs of the specific household, which determined the type of IoT sensors. The deployment of the sensors in the home is shown in Fig. 1

A. Temperature and humidity sensor - type: HTU21D

Criteria such as affordability, measurement accuracy, connectivity options, and measurement range were considered for sensor selection. These parameters were compared for DHT11, AHT10, SHT30, HTU21D, and DHT21 sensors.

After analysis, the HTU21D sensor was chosen. Its key features include high humidity measurement accuracy with a deviation of ±2% and temperature measurement accuracy with a deviation of ±0.3°C [34].

The HTU21D sensor communicates with a microcontroller via the I²C protocol. It measures relative humidity using a capacitive hygrometer and temperature using a temperature sensor. The processed values are transmitted to the microcontroller through the I²C bus.

HTU21D is an efficient and stable sensor offering precise humidity and temperature measurements. Its simple I²C interface and low power consumption make it highly suitable for various applications, particularly in smart homes and IoT systems.

B. Smart Lighting Sensor - type: WS2812B

When selecting a sensor, the criteria considered were affordability and compatibility with the 5V operating voltage of the LED strip, enabling power supply via a USB adapter. The analysis compared WS2812B, VS2813, and SK6812. Based on the established criteria, WS2812B was chosen.

These are digitally addressable RGB LEDs used in smart lighting applications. They can control each diode independently and support various lighting effects and animations. An integrated microcontroller in each diode allows for individual color and brightness settings. The diodes are arranged in strips, ensuring flexible use.

WS2812B diodes are controlled via a serial protocol, where each diode has a unique address. Data is transmitted through a data line, enabling individual color and brightness adjustments.

C. Motion sensor - LD2410C

When selecting a motion sensor, the criteria considered were motion detection at a distance of at least 4 meters, wide coverage angle, and purchase cost. The analysis compared sensors HC-SR501, HY-SRF05, and LD2410C.

The chosen sensor was the passive infrared sensor LD2410C. It features a horizontal coverage angle of approximately 120° and a vertical coverage of 30°. This wide angle allows for effective motion detection over a large area, making it ideal for security systems or lighting automation applications. The detection range extends up to 7 meters, depending on the surrounding environment and obstacles. It operates on a power supply of 3.3 - 5V. The digital output signal provides two states: motion detected or no motion. Its operating temperature ranges from -20°C to +85°C.

The LD2410C motion sensor uses FMCW (Frequency Modulated Continuous Wave) technology. Unlike traditional infrared sensors that react to temperature changes, this sensor detects movement based on variations in the electromagnetic field caused by an object's motion.

D. Smart switch

The BESTER type represents an electromagnetic relay that uses electrical current to open or close switch contacts.

This relay is a component within a broader smart switch system, often integrated into IoT devices or smart home solutions. However, it is not a standalone ”smart switch.” This relay needs to be connected to a control unit, such as a microcontroller or a Wi-Fi/Bluetooth module, for full functionality as a smart switch.

Relays of this type are commonly found in various smart sockets and switches that enable remote control of connected devices. However, they must be integrated into a controlled circuit to provide the ”intelligence” necessary for operation.

As part of our work, we use a smart switch to control the lighting in the aquarium and regulate the oxygenation of the water.

E. Voice assistant

The voice assistant in Home Assistant allows users to control smart devices locally through voice commands [8].

To enable the local voice assistant to function, plugins such as Whisper (for speech-to-text conversion), Piper (for text-to-speech conversion), and OpenWakeWord (an open-source wake word library used for detecting voice commands or activating the voice assistant) need to be added to the automation system.

The voice assistant uses the MAX98357 and INMP441 components integrated into the ESP32 platform. MAX98357 is an integrated audio amplifier chip often used for small audio systems, ensuring powerful and high-quality sound. INMP441 is a MEMS (MicroElectroMechanical Systems) microphone designed for recording sound, offering high sensitivity and low power consumption.

F. IP Camera - IMOU Bullet 2C 4MP

The installation and connection of the IP camera to the system are straightforward, and live streaming can be displayed via mobile apps [9]. For the purposes of this work, based on affordability, resolution, and ONVIF protocol support, the IMOU Bullet 2C 4MP camera was selected from the options IMOU Bullet 2C 4MP, Tenda IC7-PRS-4, and Tenda IT-PRS. The IMOU Bullet 2C 4MP offers night vision, motion detection, 2560 × 1440 px resolution, a 106° field of view, and a range of up to 30 meters. It has a MicroSD card slot (max. 256 GB) and an app for Android and iOS in both Slovak and English. An important specification is ONVIF support, ensuring secure interoperability with other devices and software.

G. Smart Button - Sonoff SNZB-01

The smart button is a compact control device that allows for remote operation of devices. It can turn appliances on and off, launch scenes, or automate [10].

The criteria for selecting the smart button included low cost and the ability to communicate based on the Zigbee protocol. The Sonoff SNZB-01 was selected after comparing it with other smart buttons like the Xiaomi Mi Wireless Switch and Imou ZE1.

H. Smart Socket - TP-Link Tapo P110M

The smart socket is an electrical device that allows easy control of electronic appliances, enabling remote control of home devices via smartphone, tablet, or voice assistant such as Amazon Alexa, Google Home, Apple Siri, or others [11].

The criterion for selecting the smart socket was the support of the Matter communication protocol. After comparison, the TP-Link Tapo P110M socket was selected over the Ee Energy Smart Plug.

V. Protocol Support

The goal of integrating communication protocols into the Home Assistant environment is to allow the system to communicate with devices that use standards efficiently. This step expands the capabilities of the Home Assistant system, providing users with a broader range of smart devices for home automation and control.

Based on a comparison of expanding modules, SONOFF Zigbee 3.0 and SkyConnect, it was found that SkyConnect is the optimal choice for integration into the Home Assistant system. It is designed with compatibility in mind and supports the Thread and Matter protocols.

The Home Assistant SkyConnect USB hub enables the addition of support for the Zigbee protocol and, after firmware updates, provides access to the Thread (Matter) protocol. It is compatible with Raspberry Pi and other Linux devices running Home Assistant, allowing for easy integration of Zigbee sensors and devices into the system [12].

VI. Testing the automation system

The core of the automation system is the Raspberry Pi 4B.

The automation was implemented in the Home Assistant platform and consisted of three parts: trigger, condition, and action. After the automation, it was necessary to verify the correct functioning of the devices connected to the automation system through testing. The work included tests for:

- **Temperature Monitoring.** A testing setup was created – a closed box with a heating pad, which has a temperature sensor that monitors the current temperature of the pad and subsequently regulates the pad's power to achieve the desired temperature of 60 degrees Celsius. Before testing, the box was preheated for two hours to stabilize the temperature. Another temperature sensor was also placed inside the closed box to verify the ambient temperature. The automation integrated into the Home Assistant system monitors the temperature sensor. If the temperature inside the closed box exceeds 27 degrees Celsius, the system automatically sends a notification alerting the user to a potential issue with the increased temperature.

  Test Scenario:

  - Placing the sensor inside the closed box to monitor the temperature.
  - Monitoring the temperature and notifications: The temperature from the sensor is monitored in the automation system, and a notification appears on the information panel.
  - Stabilization of the temperature is achieved by leaving the sensor in the box for 15 minutes.
  - Notifying notifications in an Excel table when notifications arrive.
  - Recording temperature: After the 15-minute interval, the temperature from both the monitored and control sensors is recorded.
  - Cooling of the sensor: The sensor is placed outside the box until the temperature drops to the room temperature.
  - Repeated measurements: To validate the results, the test was repeated 10 times, with each test lasting 30 minutes.

  Sensor 1 represents the monitored sensor from the HTU21D module. Sensor 2 represents the control sensor with a temperature measurement accuracy of ±1°C. Based on the temperature measurement data from sensor one and sensor 2, it was found that the temperatures from the sensor mostly matched those from the control sensor, with minimal differences ranging from 0.1°C to 0.3°C. The statistical deviation of sensor 1 is 0.11764, and of sensor two is 0.11936. Despite these deviations, the sensor provided reliable temperature measurements compared to the control sensor. Moreover, it was proven that the notification always arrived after the set temperature was exceeded.
- **Motion Sensor Monitoring** During the testing of the motion sensor, movement, presence, and distance were detected. The motion sensor was placed at the end of a hallway, 7 meters away and at a height of 85 cm. Markers were placed on the floor at distances ranging from one to seven meters from the sensor. The tests aimed to collect data for a thorough analysis of the sensor's functionality at different distances. In the motion detection test, the test subject stood sideways to each marker and raised their hand forward and back to the body after the sensor detected motion. If the sensor correctly detected the movement, the value one was recorded; otherwise, 0. This procedure was repeated for each marker to analyze the sensor's response based on the distance. The test was repeated 20 times. The motion detection test showed that the sensor achieves the highest efficiency at distances from one to five meters, with a success rate of 85%. The effectiveness dropped significantly at distances beyond five meters; at six meters, the detection was successful only 45% of the time, and at seven meters, the sensor failed to detect any movement. This indicates that the sensor reliably detects motion within five meters.

  Presence detection testing began outside the sensor's range. The test subject then stood at the corresponding marker and waited for the sensor to stop detecting movement. After 30 seconds, it was verified whether the sensor correctly responded to presence. A value of 1 was recorded for correct detection and 0 for incorrect detection. The test was repeated 10 times. The sensor reliably detected presence within five meters, with the success rate dropping to 90% at 6 meters and zero at 7 meters. These results confirm the manufacturer's claimed values but suggest the need for optimization for larger distances.
- **Voice Assistant** Verifying the voice assistant functionality in an enclosed environment includes testing its ability to respond to wake phrases reliably. The voice assistant was tested in a room with an area of 11m². The voice assistant who spoke the wake phrase was placed one, two, or three meters from the test subject. The automation system was observed to see if the voice assistant detected the wake phrase and was able to process the command. The test was repeated 10 times. The test results indicate that the voice assistant effectively responded to wake phrases at distances ranging from one to three meters.

  Response to command: The goal was to verify whether the voice assistant could successfully process and execute the spoken command. The voice assistant received the command "Alexa, bedroom light on/off". The success of the command execution was assessed by whether the light turned on. Testing was conducted at distances of 1-3 meters. The automation system responded in various ways to the command: turning on the light (success), displaying an error message (failure), or not responding at all. The test was repeated 10 times for each distance. The voice control test showed the highest success rate (70%) at 1 meter. As the distance increased, the success rate decreased significantly (40% at 2 meters, 30% at 3 meters). The low success rate could be due to the use of a basic model for speech recognition, microphone quality, or the surrounding environment. Despite these limitations, the voice assistant offers a promising solution for smart homes, particularly due to privacy protection and fast response times.

  Response time of the voice assistant to a command: The testing involved issuing the voice command ”Alexa, bedroom light on/off” and measuring the response time of the smart lamp. Commands that were not responded to by the lamp were ignored. The test was repeated 15 times for each distance. The testing revealed that the average response time of the voice assistant was 3.61 seconds.
- **Smart Button** The functionality of the smart button and the smart lamp was verified. The smart button operates on the Zigbee wireless protocol. The smart button was placed at various distances from the Zigbee coordinator, specifically at distances of 1, 3, 5, and 7 meters. The smart lamp is controlled via Wi-Fi, and the Wi-Fi router is located 7 meters away. The smart button demonstrated reliable smart lighting control at the defined distances.

A. *Control of the automation system*

The Home Assistant automation system can be controlled in various ways. It can be used through smartphones, web browsers, smartwatches, etc. [8], [13]. In our case, a web browser was used, which provides full access to its functions. It allows for the creation of complex automation and the integration of various devices. However, for some advanced features, it is necessary to ensure encrypted communication.

VII. Smart Home Security

Home Assistant is an open-source home automation platform. It offers a wide range of integrations and extension options, making it very flexible. However, this flexibility also comes with certain risks. Specifically, installing add-ons from community sources, such as the Home Assistant Community Store (HACS), can pose security threats. Vulnerabilities have been reported in these add-ons in the past, which could be exploited to gain unauthorized access to the system [14], [15]. It is important to select and decide which add-ons to use carefully, regularly update the system and add-ons, and use strong and unique passwords to minimize the risk.

Security in smart homes is critical because these systems are connected to the internet and contain personal data. In this work, smart home security was implemented through:

- Strong and unique passwords.
- Software updates.
- Secure Wi-Fi network (strong password, encryption, guest network).
- Two-factor authentication for all accounts associated with smart home devices.
- Physical security - devices are placed out of physical reach from unauthorized individuals, especially those with administrative rights.
- Choosing reliable manufacturers - security updates are regularly released for their devices.
- Regular monitoring - consistently checking activity to understand what is happening within the network.
- Antivirus software on PCs or mobile devices used to control the smart home.
- Network segmentation - where the network is divided into multiple segments to limit the potential impact of an attack on a single device.
- Regular backups of important data - so it can be restored in case of loss.
- Continuous learning - staying updated on new threats and security practices.
- Testing custom solutions - implementing and verifying your own security solutions.

Security is an ongoing process. Regularly reassessing and updating security measures can significantly reduce the risk of attacks on smart homes.

Home Assistant is a powerful tool, but its security depends on the user. Strict adherence to security measures is essential for the smooth operation and protection of the smart home.

VIII. Methodology for the Design and Development of a Safe Smart Home

In the development of a smart home using a systems approach, a methodology was developed based on the following steps:

- Definition of goals and requirements
  - User needs analysis - what users expect from the system.
  - Consideration of security requirements - identification of threats (cyberattacks, physical threats) and defining requirements for their minimization.
  - Setting the budget and constraints - summarizing technical and financial limitations.
- System architecture design
  - Selection of technologies - suitable (sensors, IoT protocols, open-source platforms).
  - Architectural decisions - system model, selection of communication protocols.
  - Security framework - implementation of security measures (encryption, authentication, intrusion protection, etc.).
- System implementation
  - Creation of a prototype, installation of devices, deployment of software, and ensuring interoperability between system components.
- Testing and validation
  - Security testing - resilience to cyberattacks and manipulation.
  - Functionality testing - verifying if all functions work as expected.

- Operation and maintenance
  - Monitoring and updating - establishing regular checks and software updates.
  - Incident response - creating procedures for handling incidents/outages.
  - System expansion - adding new elements or functions.
- Documentation
  - Creating a user guide and standardizing processes.

IX. Conclusion

Smart homes represent a significant advancement in modern housing, offering higher levels of comfort, efficiency, and security. This work highlights the critical role of a systems approach in designing smart homes, enabling the effective integration of various devices and technologies into a unified, user-friendly environment. We proposed a comprehensive methodology for smart home development, covering systematic steps from initial planning and analysis through to design and secure implementation, with a strong focus on security, usability, and solution flexibility.

The combination of open-source tools with thoughtful planning and targeted sensor deployment demonstrates that cost-effective and adaptable solutions can be developed to meet the varying needs of users. Our results underline the key benefits of centralized control and process automation, which improve user comfort, optimize energy usage, and reduce operating costs. A particular emphasis was placed on cybersecurity, recognizing it as a fundamental component of modern smart home systems. Thorough risk identification and the implementation of tailored security measures are essential to ensure reliable and uninterrupted system operation.

Beyond presenting a practical framework for smart home development, this work opens avenues for future research. In particular, we aim to explore the application of artificial intelligence for predictive analysis of user behavior, enabling a new level of personalization, proactive system adaptation, and advanced automation. Furthermore, the creation of a dedicated testing network will support extensive stress testing, crucial for the development and validation of novel cybersecurity solutions tailored to smart environments. Our future efforts will focus intensively on strengthening the resilience, adaptability, and intelligence of smart home ecosystems.

ACKNOWLEDGMENT

This work was supported in part by the Slovak Scientific Grant Agency (VEGA) Grant Agency through the Research of Effective Localization and Communication Solutions in Heterogenous Wireless Networks 1/0580/25 and funded by the European Union NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No.17R05-04-V01-00005. We also thank Marek Ziacek, who helped create this paper.

References

[1] M. C. Șuvar, L. Munteanu, and C. Cioară, “Optimal Monitoring of Server Rooms with Home Assistant Platform,” MATEC Web of Conferences, vol. 373, p. 00044, 2022, doi: 10.1051/MATECCONF/202237300044.

[2] B. Akhmetzhanov, B. Akhmetzhanov, S. Ozdemir, and N. Zhakiyev, “Advancing affordable IoT solutions in smart homes to enhance independence and autonomy of the elderly,” Journal of Infrastructure, Policy and Development, vol. 8, no. 3, 2024, doi: 10.24294/JIPD.V8I3.2899.

[3] O. Izquierdo-Monge, P. Peña-Carro, R. Villafafila-Robles, O. Duque-Perez, A. Zorita-Lamadrid, and L. Hernandez-Callejo, “Conversion of a network section with loads, storage systems and renewable generation sources into a smart microgrid,” Applied Sciences (Switzerland), vol. 11, no. 11, Jun. 2021, doi: 10.3390/APP11115012.

[4] L. Munteanu, M. C. Suvar, and G. D. Florea, “Residential security through the Home Assistant Platform,” MATEC Web of Conferences, vol. 354, p. 00008, 2022, doi: 10.1051/MATECCONF/202235400008.

[5] K. Akhmetzhanov, O. A. Gazizuly, Z. Nurlan, and N. Zhakiyev, “Integration of a Video Surveillance System Into a Smart Home Using the Home Assistant Platform,” SIST 2022 - 2022 International Conference on Smart Information Systems and Technologies, Proceedings, 2022, doi: 10.1109/SIST54437.2022.9945718.

[6] B. Setz, S. Graef, D. Ivanova, A. Tiessen, and M. Aiello, “A Comparison of Open-Source Home Automation Systems,” IEEE Access, vol. 9, pp. 167332–167352, 2021, doi: 10.1109/ACCESS.2021.3136025.

[7] J. A. Cujilema Paguay, G. A. Hidalgo Brito, D. L. Hernandez Rojas, and J. J. Cartuche Calva, “Secure home automation system based on ESP-NOW mesh network, MQTT and Home Assistant platform,” IEEE Latin America Transactions, vol. 21, no. 7, pp. 829–838, Jul. 2023, doi: 10.1109/TLA.2023.10244182

[8] “Assist - Talking to Home Assistant - Home Assistant.” Accessed: Apr. 17, 2024. [Online]. Available: https://www.home-assistant.io/voice control/

[9] L. J. Fennelly and M. A. Perry, Physical Security: 150 Things You Should Know: Second Edition. Elsevier Inc., 2016. Accessed: Apr. 17, 2024. [Online]. Available: http://www.sciencedirect.com:5070/book/9780128094877/physical-security-150-things-you-should-know

[10] “SONOFF Zigbee Button - a Zigbee remote controller button (SNZB-01) - eWelink Store.” Accessed: Apr. 17, 2024. [Online]. Available: https://ewelinkstore.com/product/sonoff-zigbee-button-a-zigbee-remote-controller-button-znzb-01/?v=13dd621f2711

[11] P. Mtshali and F. Khubia, “A smart home energy management system using smart plugs,” 2019 Conference on Information Communications Technology and Society, ICTAS 2019, Apr. 2019, doi: 10.1109/ICTAS.2019.8703522.

[12] S. N. Tayus, A. K. M. Kamrul Alam Kakon, and M. Ullah, “IoT Based Web Controlled Multiple Home Automation and Monitoring with Raspberry Pi,” 2022 3rd International Conference for Emerging Technology, INCET 2022, 2022, doi: 10.1109/INCET54531.2022.9825087.

[13] F. Corno and L. Mannella, “A Threat Model for Extensible Smart Home Gateways,” 2022 7th International Conference on Smart and Sustainable Technologies, SpliTech 2022, 2022, doi: 10.23919/SPLITECH55088.2022.9854235.

[14] S. Jaafar. (2024). Smart Home Security Designing An Effective IoT. DOI:10.13140/RG.2.2.17570.85445.

[15] G. Vardakis, G. Hatzivasilis, E. Koutsaki, N. Papadakis. (2024). Review of Smart-Home Security Using the Internet of Things. Electronics. 13. 3343. 10.3390/electronics13163343.