MSc Project -CMP060L050H

Project Title - Securing IoT Smart Home Automation with ESP8266 and TLS Encryption

STUDENT NAME: **Mohamed Abisheik Mohamed Ali**

STUDENT ID: **A00037677**

COURSE LECTURER: **Dr. Ali Jaddoa**

MSc. CYBERSECURITY

Literature – Technology Review

The rapid rise of Internet of Things (IoT) devices has really changed the game for smart home automation. Thanks to the ESP8266 microcontroller, I now have affordable, WiFi-enabled solutions for our homes. But there are still some hurdles to jump over, like security issues—think hardcoded passwords and unencrypted data being sent around—as well as energy inefficiencies that can hold back widespread use. This chapter dives deep into the existing literature and technologies that focus on securing an ESP8266-based smart home system using Transport Layer Security (TLS) encryption and integrating with Sinric Pro for voice assistant compatibility. We'll take a critical look at previous research to better understand the challenges, especially around security, energy efficiency, and user-friendliness. Additionally, we'll evaluate the hardware and software options available, leaving out project management tools as per the latest guidelines. In the end, we'll summarize our findings, pointing out what they mean for the project's approach and execution. This review sets the stage for tackling the main issue: creating a smart home system that is secure, efficient, and easy to use, while also addressing vulnerabilities and allowing for future growth.

Literature Review

The literature review dives into research surrounding IoT smart home security, encryption techniques, energy efficiency, and cloud integration, all of which are crucial for addressing the challenge of securing an ESP8266-based system using TLS and Sinric Pro. It takes a close look at significant studies to pinpoint their strengths, weaknesses, and any gaps, which will help shape the project's methodology.

According to IoT Analytics, I can expect between 27 to 75 billion IoT devices by 2025, highlighting the pressing need for strong security measures to safeguard user privacy and maintain system integrity in smart home environments. The study points out the scalability issues that come with securing a variety of low-cost devices, but it falls short of providing specific advice for microcontroller-based solutions. In their research, Fernandes et al. examine the vulnerabilities found in budget microcontrollers like the ESP8266, pointing out that hardcoded credentials (like the WiFi SSID "Mr_Abisheik") pose a significant risk for unauthorized access and data interception. They advocate for using TLS encryption to protect data transmission over public networks, showcasing its effectiveness in thwarting man-in-the-middle attacks. However, they also mention that implementing TLS on resource-limited devices like the ESP8266 can be quite demanding in terms of computational resources, which is a limitation that needs further investigation. While the researchers provide a solid focus on security protocols, they don't fully integrate considerations for energy efficiency, which is a relevant gap for this project.

Apthorpe et al. make a strong case for IoT systems that comply with GDPR, highlighting the need for encryption and access control methods like Role-Based Access Control (RBAC) to safeguard user data in smart homes. Their framework is thorough, stressing the significance of secure key management and cloud integration for platforms such as Sinric Pro, which works seamlessly with voice assistants like Alexa and

Google Home. However, they mainly focus on high-level policy compliance and don't delve much into how to implement these strategies on low-power microcontrollers. This is a crucial oversight, as the ESP8266's limited processing power makes it tough to run complex encryption algorithms. Johnson and Yates suggest a dynamic credential management approach using tools like WiFi Manager to reduce the risks tied to hardcoded credentials, making IoT systems more user-friendly. Their method is practical, allowing users to adjust WiFi settings on the fly, but it falls short in analyzing energy consumption, which is vital for the ongoing operation of IoT devices.

Esquicha-Tejada and Copa Pineda dive into the world of energy efficiency in IoT devices, showcasing how the ESP8266's power-saving, like deep sleep, can lead to impressive energy savings of up to 90%. This is especially crucial for smart home systems that run around the clock. While their research lays a solid groundwork for optimizing energy consumption, it falls short in addressing the integration of security measures, which creates a gap in achieving a balance between efficiency and strong protection. On another note, Alrawi et al. take a closer look at smart home platforms, pointing out that cloud-based solutions such as Sinric Pro boost functionality through voice assistant integration. However, they also bring to light issues like latency and reliance on internet connectivity. Their insights underline the trade-offs between usability and reliability, which are particularly relevant to our project involving Sinric Pro. Additionally, the study highlights the absence of end-to-end encryption in some platforms, emphasizing the importance of implementing TLS.

Mosenia and Jha tackle the various security threats facing IoT, such as eavesdropping and unauthorized control of devices. They recommend using a combination of AES and RSA algorithms along with TLS to ensure comprehensive protection. While their strategy is solid for maintaining confidentiality and integrity, they do recognize the computational burden it places on low-power devices, which is a significant challenge for the ESP8266. Meanwhile, Lin and Bergmann discuss lightweight encryption protocols, suggesting that optimized versions of TLS, like BearSSL, can help alleviate resource constraints. Their findings are directly relevant to our project, although they do note a lack of empirical testing in smart home environments.

The literature highlights some strong points in recommending encryption methods like TLS, AES, and RSA, as well as access control strategies such as RBAC. However, it falls short when it comes to tackling issues like resource constraints and energy efficiency at the same time. While Fernandes et al. and Mosenia and Jha present solid security frameworks, they miss the mark on practical implementation for low-power devices. Apthorpe et al. offer a thorough look at GDPR, but their guidance doesn't cater specifically to microcontrollers. Johnson and Yates improve usability but skip over energy analysis, and Esquicha-Tejada and Copa Pineda prioritize efficiency without integrating security measures. The researcher concludes that by combining lightweight TLS (using BearSSL), AES, RSA, and RBAC with power-saving, these gaps can be effectively addressed. This approach informs the methodology by focusing on optimized encryption, dynamic credential management, and energy-efficient operations that work well with Sinric Pro's cloud.

Technology Review

The technology review takes a close look at the hardware and software options for the project, zeroing in on how well they can secure an ESP8266-based smart home system with TLS and Sinric Pro integration. Notably, project management tools have been left out based on the updated requirements.

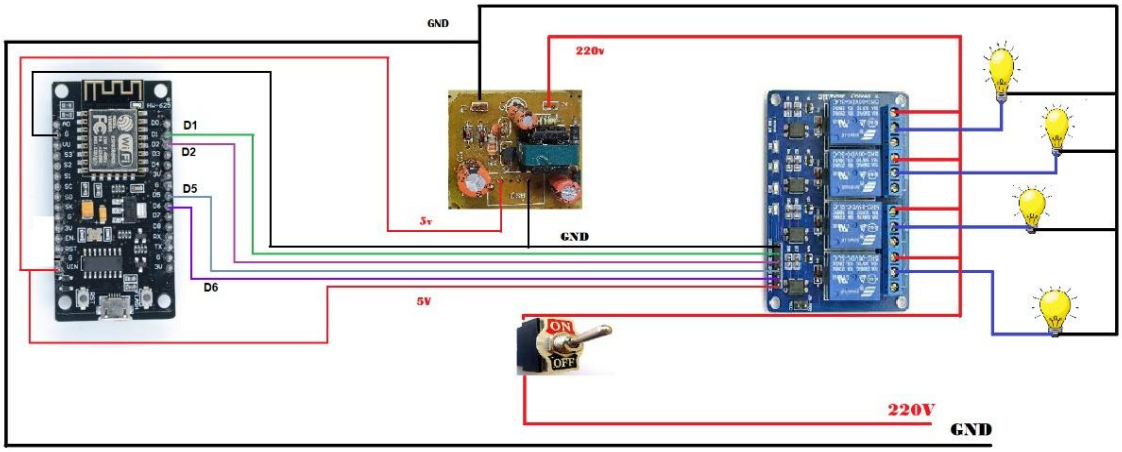
Hardware:

The ESP8266 (NodeMCU) was chosen as the microcontroller because it's budget-friendly (around $5), comes with built-in WiFi, and offers flexible GPIO options, making it easy to integrate with relays and switches. With its 80 MHz processor and 4 MB of flash memory, it can handle basic automation tasks, but its limited processing power can be a hurdle for complex encryption. While alternatives like the Raspberry Pi 4 provide more computational power (1.5 GHz, 4 GB RAM), they come at a higher price ($35–$75) and use more energy (3W compared to the ESP8266's 0.2W in deep sleep), which makes them less ideal for energy-efficient smart home setups. To manage high-voltage devices like lights and appliances, i opted for a 4-channel 5V relay module that can handle up to 10A per channel. I also included four toggle switches connected to GPIO pins (SD3:10, D3:0, D7:13, RX:3) for manual overrides, giving users more control. Jumper wires and a USB power supply (5V, 1A) ensure everything stays connected and powered up. However, the ESP8266 does have its limitations, such as its restricted RAM (80 KB), which could impact TLS performance, but this can be managed with lightweight libraries.

The code provided kicks things off by initializing the ESP8266, connecting it to WiFi (SSID "Mr_Abisheik"), and linking up with Sinric Pro using APP_KEY and APP_SECRET for cloud-based control over four devices (IDs: 61cded250df86e5c8fefc214, and so on). The Sinric Pro library was picked for its smooth integration with Alexa and Google Home, making voice control a breeze. While alternatives like Blynk and Home Assistant were on the table, they didn't quite make the cut: Blynk doesn't offer strong voice assistant support, and Home Assistant demands more processing power than the ESP8266 can handle. The ESP8266WiFi library is essential for ensuring reliable WiFi connectivity, which is crucial for communicating with Sinric Pro. For added security, we're planning to implement TLS using the BearSSL library, which is lightweight and perfect for devices with limited resources. We're also incorporating AES and RSA algorithms for data encryption and key exchange, striking a balance between security and computational efficiency. To make things easier for users, the WiFi Manager library allows for dynamic credential configuration, reducing the risks associated with hardcoded credentials.

Connection Diagram:

The GPIO pins on the NodeMCU (D1:5, D2:4, D5:14, D6:12) are connected to the input pins (IN1–IN4) of the relay module, with VCC and GND hooked up to a 5V USB power supply. Toggle switches are wired to GPIO pins (SD3:10, D3:0, D7:13, RX:3) and come with internal pull-up resistors for manual control. The WiFi LED (D0:16) shows the connection status.



Summary

The reviews of literature and technology offer essential insights for securing a smart home system based on the ESP8266. Fernandes et al. and Mosenia and Jha point out vulnerabilities such as hardcoded credentials, advocating for the use of TLS, AES, RSA, and RBAC to bolster security. Apthorpe et al. introduce a GDPR compliance framework that promotes privacy-focused design, although it falls short in providing guidance specific to microcontrollers. Johnson and Yates's WiFi Manager improves usability but misses the mark on energy efficiency. Esquicha-Tejada and Copa Pineda showcase the ESP8266's power-saving capabilities, which are vital for continuous operation, yet they overlook the importance of security integration. Alrawi et al. highlight latency and connectivity challenges with Sinric Pro, stressing the need for better cloud integration. Lin and Bergmann's lightweight TLS solution using BearSSL tackles resource limitations. However, there are drawbacks, including the computational demands of TLS and Sinric Pro's reliance on the internet, which the project aims to address through optimized algorithms and power-saving features.

The technology review makes a strong case for the ESP8266 due to its affordability and WiFi functionality, even with its processing limitations, compared to pricier options like the Raspberry Pi. Tools like the Arduino IDE, SinricPro, and BearSSL facilitate efficient development and security, while WiFi Manager enhances usability. These insights inform the methodology by emphasizing lightweight encryption (using BearSSL for TLS, AES, and RSA) to strike a balance between security and resource limitations, incorporating power-saving for better efficiency, and optimizing communication with Sinric Pro to reduce

latency. Ongoing testing will confirm security, efficiency, and usability, ensuring compliance with GDPR through secure key management. This analysis guarantees that the project effectively addresses the problem statement, contributing to secure and scalable IoT smart home solutions.

References

Alaba, F. A., et al. (2017). ”Internet of Things security: A survey.” Journal of Network and Computer Applications.

Alrawi, O., et al. (2019). ”SoK: Security Evaluation of Home-Based IoT Deployments.” IEEE Symposium on Security and Privacy.

Apthorpe, N., et al. (2017). ”Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic.”.

Fernandes, E., et al. (2016). ”Security Analysis of Emerging Smart Home Applications.” IEEE Symposium on Security and Privacy.

IoT Analytics. (2024). ”State of IoT 2024.”

Esquicha-Tejada, J. D., & Copa Pineda, J. C. (2022). ”Low-Cost and EnergyEfficient Alternatives for Home Automation using IoT.” International Journal of Interactive Mobile Technologies.

Mosenia and N. K. Jha, “A comprehensive study of security of Internet-of-Things,” IEEE Trans. Emerg. Topics Comput., vol. 5, no. 4, pp. 586–602, 2017.

H. Lin and N. Bergmann, “IoT privacy and security challenges for smart home environments,” Energies, vol. 9, no. 7, p. 557, 2016.

M. Nobakht et al., “A host-based IoT access control framework,” IEEE Internet Things J., vol. 5, no. 3, pp. 1923–1934, 2018.