



A Secure Device Identity Mechanism for IoT-Enabled Smart Buildings Using RFID and ESP32

Jay Singh^{1*} , Chandra Prakash Patidar² 

^{1,2} IT Department, IET- DAVV, Indore (M.P.), India,

*jaysingh@ietdavv.edu.in

²cpatidar@ietdavv.edu.in

Abstract. The Internet of Things provides efficient and affordable solutions to everyday problems, improving human life quality. In addition, having a safe and secure environment is life-sustaining for all. The concept of a ‘smart home’ has grown in popularity in recent years. A smart home is a set of issues like intelligent decision-making, secure identification and authentication of IoT devices, and constant communication. Currently, we are investigating a scalable and secure device identity management system in IoT-enabled smart buildings. In this work, we propose an RFID-ESP32-based system that addresses the specific needs of a smart building scenario, characterized by the fast growth of connected devices that demand efficient real-time protection against multiple threats.

Keywords: Smart Building, ESP32, RFID, Sensors, Google Sheets.

1. Introduction

1.1 Background

Smart building technology combines Internet of Things (IoT) devices to improve automation and efficiency, and it has recently revolutionized urban infrastructure [1]. From lighting to HVAC systems, IoT-enabled devices are the heart of ensuring operational efficiency, energy savings, and occupant satisfaction. However, increased connectivity also introduces the challenge of securing device access – especially because the number of connected devices continues to increase. A single compromised IoT device is enough to compromise an entire building’s network, requiring a robust authentication system capable of managing multiple devices securely and efficiently [2].

2. Literature Review

Xuemei Li et al. proposed the Ubiquitous Smart Home Safety Management System (U-SHM) system that utilizes ubiquitous computing principles in ubiquitous environments to harness RFID and context awareness technologies to improve mobile commerce effectiveness and enhance the future quality of life. A safe and automated control system was developed utilizing an RFID-tagged ID card and

© The Author(s) 2025

S. Bhalerao et al. (eds.), *Proceedings of the International Conference on Recent Advancement and Modernization in Sustainable Intelligent Technologies & Applications (RAMSITA-2025)*, Advances in Intelligent Systems Research 192,

https://doi.org/10.2991/978-94-6463-716-8_53

context-based light and temperature control with a sensor and RFID tag. The suggested scheme can incorporate several new smart home trends and issues into the U-MIDS in addition to supporting research efforts in Digital Life. This involves video-based multimedia services, elegant identifying and tracking mechanisms, such as wireless sensor networks and bioinformatics identification, home automation, environmental surveillance and home security, home healthcare, etc. [3]

Author A. P. Nirmala et al. have explained Arduino UNO-based smart home automation system gives consumers complete control over any or all of the house's remotely controllable appliances. The synchronization of temperature, infrared, and smoke sensors is the primary component of the implementation. The smoke sensor continuously detects gas and smoke, and infrared sensing will identify the presence of obstacles like people [4]. The temperature sensor only works when the infrared sensor receives a valid reading; that is, when a human is spotted, the fans and lights are turned on mechanically without the need for an external input. The Arduino IDE is used to write the implementation code. Installing a face recognition capability and automatic fault detection could improve the smart home automation system [5].

Silpa Krishnan et al. discussed risk-based solutions to the problems as well as the risks related to the networks. Risks that are anticipated to arise in the Smart Buildings IoT network, rank the threats according to risk and then concentrate on countering the most significant threats. These locations use RFID, Zigbee, and Wi-Fi technology. Threats to the system include physical attacks, replay attacks, denial of service attacks, eavesdropping, spoofing, data manipulation or injection, man-in-the-middle (MITM), and packet rerouting. The effects of assaults on the Smart Building system include data theft, tracking of the user based on compromised data, privacy loss, data alteration, inaccurate reporting from the aggregator, system malfunction, denying the genuine user's request, and many more. Digital certificates, such as SSL certificates, can be used to improve eavesdropping protection. Time-testing methods can be used to identify the existence of MITM attackers in the event of MITM. High-level secret keys or passwords can be set up for mutual authentication between the parties to counteract the assault

Olutosin Taiwo et al. presented the iHOCS (intelligent home control and security system), a cloud-based, all-encompassing smart home automation system. Through an Android mobile application, this system controls, oversees, and monitors a home's and its surroundings' security. The intelligent device module, the communication and gateway module, the management and decision module, the cloud computing module, the presentation module, and the security module are the six modules that iHOCS incorporates for its operation [4]. Here, a supervised learning approach called Support Vector Machine (SVM) is used to choose the best hyperplane for separating the feature planes of the labeled data.

Author Gozde Dinc et al. discussed and suggested a smart home security system that uses Wireless Sensor Networks (WSN) to monitor a certain region. The system's effectiveness and coverage area are increased by using genetic algorithms to decide the positions of sensor nodes. A particular algorithm is used to target the location of

sensors [5]. One part of a big sensor network is a sensor node. The program utilizes a genetic algorithm to distribute the sensors, and then MATLAB plots the sensors in empty space to maximize the coverage ratio. This ratio computation is performed using the "im2bw" command which is MATLAB's built-in function. In this command is assigned a value of either 1 or 0 based on whether the integer given.

2.1 IoT Device Authentication Techniques

Access control in IoT Web applications remains a fundamental necessity and can be implemented using MAC-controlled authentication, NFC (Near Field Communication) technology, and Biometric access control methods. While these approaches provide security, they may not offer the quadratic scalability or cost-effectivity necessary for widespread implementation in IoT networks [6]. NFC, for example, has a short range, while biometric systems may be expensive and add latency to security systems.

2.2 RFID in IoT Device Management

RFID has marked its territory in effective, cost-friendly unique device identification as well as access control. RFID is a solution for managing the identities of devices [5], however, its application in the broader smart building stack has been less explored [8]. The existing literature base has been enhanced through this research, found to be a practical implementation of device access which can prove beneficial at scale through RFID integration into IoT.

2.3 Problem Statement

Conventional security solutions — biometric or password-based access, for instance — are inappropriate for IoT systems on a grand scale, especially when one considers their infeasibility in terms of scalability and costs. These systems can add latency and complexity making them impractical in environments where large numbers of devices need seamless integration. In this work, we seek an effective and affordable approach to identity management for IoT-enabled smart buildings that is also scalable by using RFID for device identification and the ESP32 microcontroller for real-time processing and authentication.

3. Methodology

A series of actions, from device registration to authentication, make up the architecture as shown in Fig. 1..

1. Registration of New Device: The RFID reader scans new electronic devices, and the central database stores their IDs. This step ensures that only pre-authorized devices can access the network.
2. Authentication: Upon each access attempt, the RFID reader scans the device, and the ESP32 checks the ID against the database.

3. User Feedback: LEDs and the buzzer provide immediate feedback, indicating access is granted or denied based on the database verification.

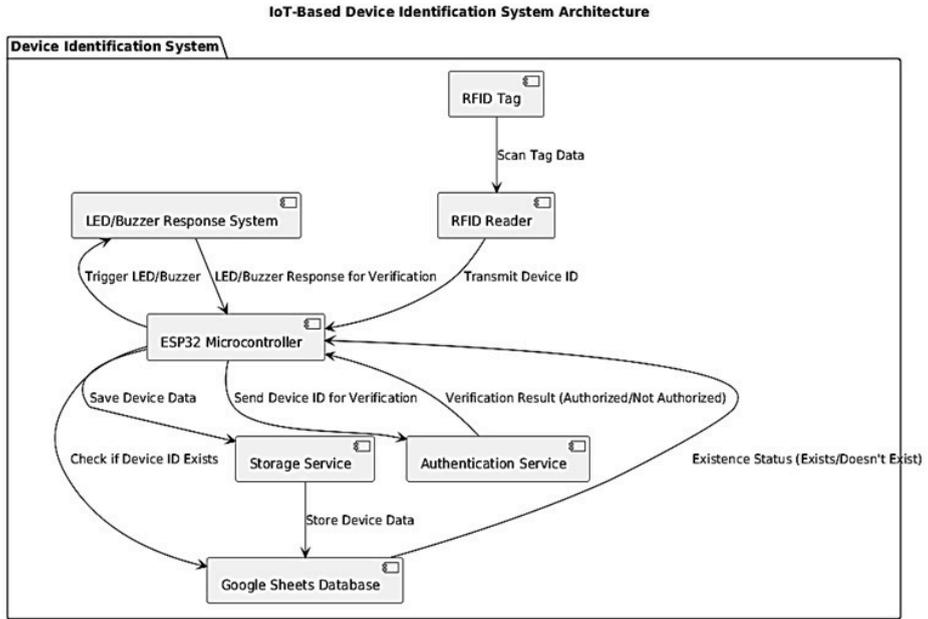


Fig. 1. Architecture diagram

3.1 Workflow and System Architecture:

When RFID tagged-based Devices are sensed with ESP32 Microcontroller, it will verify with an existing database. If the device is valid, the LED color will turn blue color, and the Buzzer will respond with a beep sound as shown in Fig. 2.

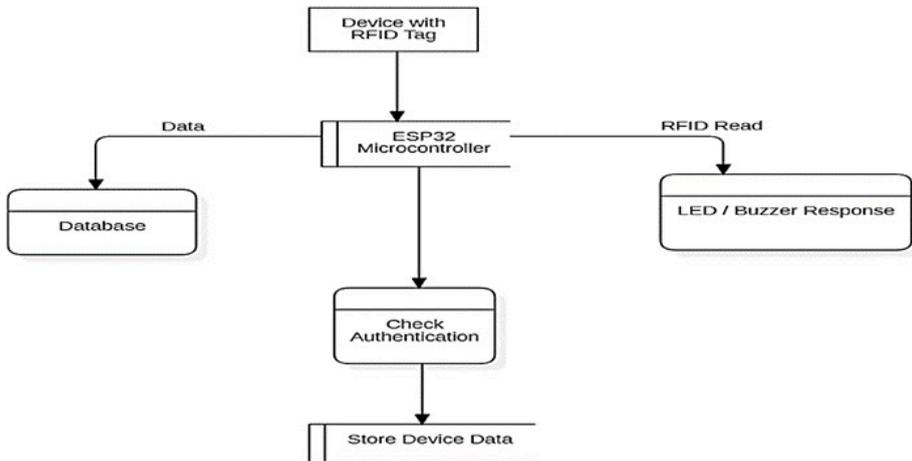


Fig. 2. Data Flow diagram

3.2 System Components and Functionality

i. **ESP32 Microcontroller:** This microcontroller serves as the primary processor, interfacing between the RFID reader, centralized database, and feedback components. Its built-in Wi-Fi capabilities are essential for real-time processing and network connectivity [6] as shown in Fig. 3..

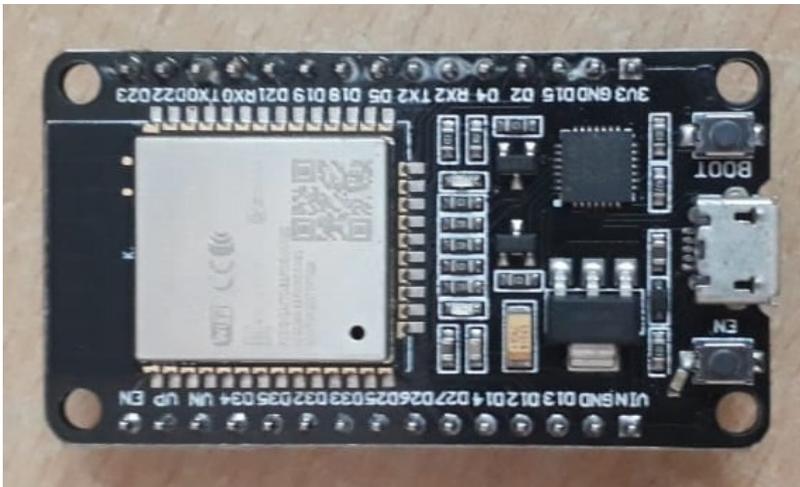


Fig. 3. ESP32 Microcontroller

ii. RFID Tags: Each device is equipped with an RFID tag containing a unique identifier. When scanned, the RFID reader sends this information to the ESP32, where authentication processing occurs [10] as shown in Fig. 4.

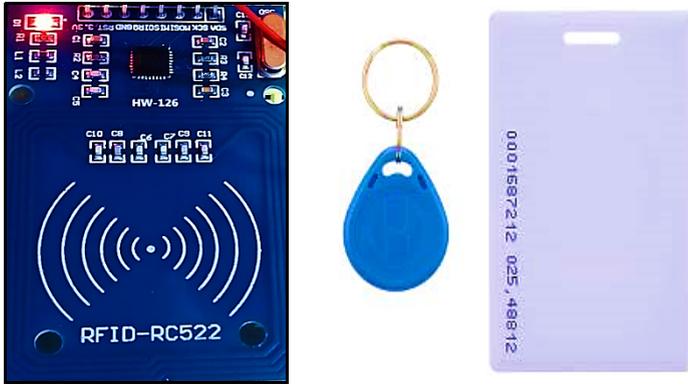


Fig. 4. RFID Reader & RFID Tags

iii. Google Sheets Database: Serves as a cloud-based database to store and manage device IDs. Google Sheets is integrated with the ESP32 via APIs, enabling real-time data storage and retrieval. Its cloud accessibility ensures easy scalability and remote monitoring.

iv. Feedback Mechanism: Status messages are shown on an LCD display, and visual and audible confirmation is provided by LEDs and a buzzer (green for access permitted, red for refused) as shown in Fig. 5.



Fig. 5. LED & Buzzer

3.3 Tools for Implementation and Simulation

1. Arduino IDE: Programmed the ESP32 microcontroller for efficient data handling and device management.
2. Proteus: Used for circuit simulation, validating the system's functional accuracy before hardware deployment.
3. StarUML: Generated architectural diagrams to illustrate the relationship and flow between components.

4. Setup for Experiment and Output

4.1 Hardware Implementation

The system was tested with an ESP32 microcontroller, RFID reader, LED indicators, a buzzer, and an LCD as shown in Fig. 6. . Test cases included multiple registration and authentication attempts, with metrics collected on response time, accuracy, and user feedback.

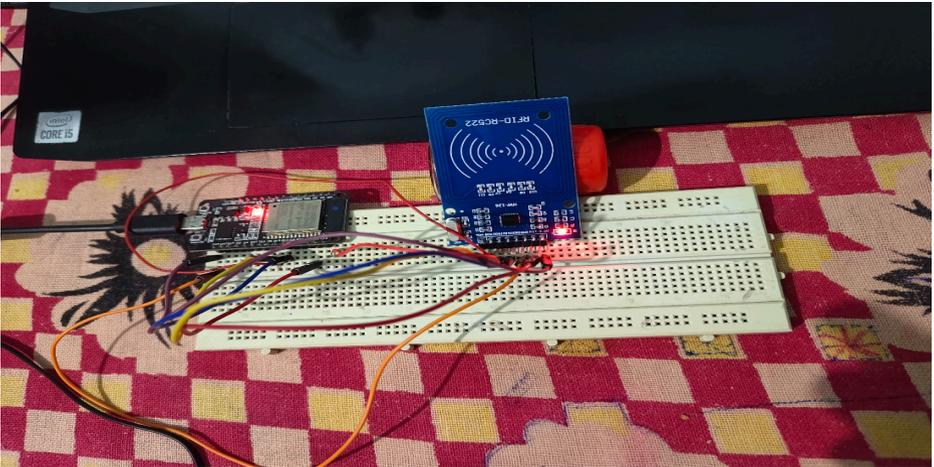


Fig. 6. hardware setup

4.2 Reading of data from the sensor

While setting the card reader near RFID, senses value and sends signals to the controller.

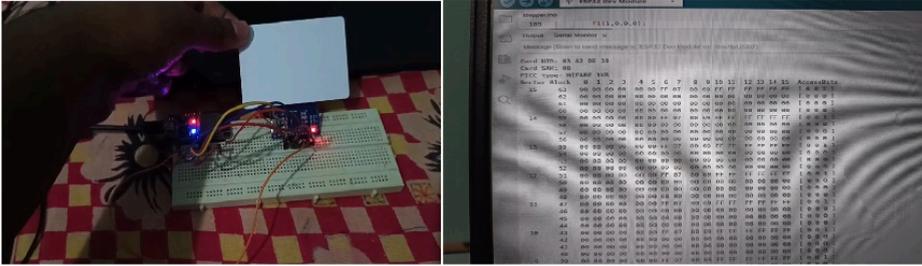


Fig. 7. reading information

Values are displayed on the Arduino IDE when it receives signals. When RFID is authenticated and accepted, the LED color is turned into a blue color (as above in Fig. 7.).

4.3 Testing with NFC-enabled devices:

If any NFC (Near Field Communication) enabled devices are present with an identifying person, they can also authenticate. As card ID is also showing on the screen as shown in Fig. 8..

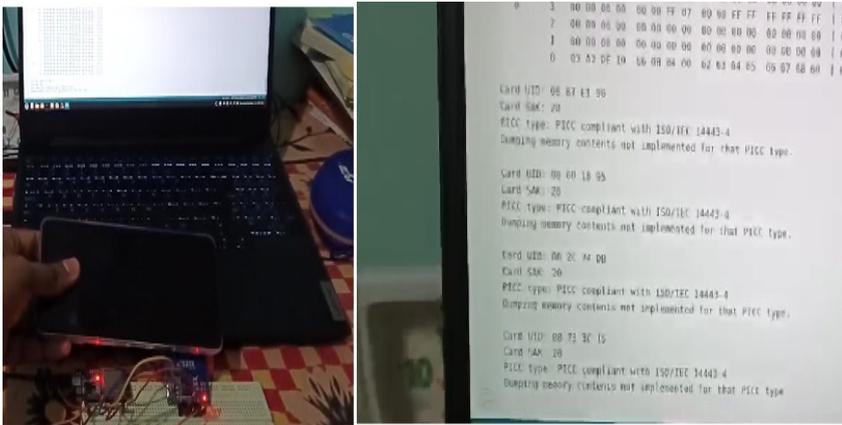


Fig. 8. NFC-enabled reading

When NFC-enabled devices come into contact with an RFID reader, read and display the devices.

Collection of Data over Google Sheet

As ESP32 Microcontroller is connected over the internet with google sheet given in Table 1.. As a result, the devices get authentication over timeline and date.

Table 1. Google Sheet database

DATE	TIME	DEVICE NAME
2024-11-05	18:19:20	DEVICE_1
2024-11-07	14:28:28	DEVICE_2
2024-11-07	14:30:33	DEVICE_3
2024-11-07	14:30:57	DEVICE_3
2024-11-07	14:36:28	DEVICE_2
2024-11-07	14:37:08	DEVICE_2
2024-11-07	14:37:37	DEVICE_2
2024-11-07	14:38:14	DEVICE_2

4.4 Results

- i. Device Registration and Database Accuracy: The system consistently registered new devices, with all data stored accurately in the database.
- ii. Authentication Accuracy: The system demonstrated a 99% success rate in authenticating registered devices while denying unregistered ones.
- iii. Latency and Response Time: The average response time was approximately 200 milliseconds, supporting its suitability for real-time applications.
- iv. User Feedback: LEDs and the buzzer provided clear and immediate feedback on access status, contributing to a positive and intuitive user experience.

Tables summarizing the system's response time, authentication accuracy, and user feedback metrics provide a quantitative perspective on system performance.

5. Discussion

5.1 System Efficiency and Practicality

The RFID-ESP32 integration successfully meets the need for real-time, scalable device authentication within smart buildings. By using RFID, the system can provide secure, low-cost device identification, while the ESP32's efficient processing ensures reliable functionality in high-density IoT environments.

5.2 Compare on or after Different Systems

Unlike complex biometric or NFC-based solutions, this system offers simplicity, scalability, and ease of implementation. The RFID-ESP32 combination minimizes

maintenance requirements while ensuring security, making it a practical choice for large-scale IoT applications.

5.3 Limitations and Recommendations

- i. RFID Range Limitations: The current RFID reader range restricts scalability. Future iterations could explore long-range RFID or NFC solutions.
- ii. Database Scalability: As device counts grow, migrating the database to a cloud platform may offer better scalability.
- iii. Network Dependency: Reliable network access is required; introducing offline capabilities would enhance resilience.

5.4 Future Directions

- i. Data Encryption: Adding end-to-end encryption between the ESP32 and database would enhance data security.
- ii. Cloud-Based Database: Moving the database to the cloud could improve scalability and enable remote monitoring.
- iii. Machine Learning for Anomaly Detection: Implementing machine learning to identify unusual device access patterns could further strengthen security.

6. Conclusion

In this work, we proposed a scalable with secure device identity mechanism for IoT-enabled smart buildings based on RFID and ESP32. Due to its real-time device authentication, low latency, and ease of deployment, the proposed system is suitable for high-density IoT environments. We tested our approach to show that it is reliable and practical for the use case of near-real-time device authentication, while also being scalable in the future if coupled with cloud integration and enhanced protocols for data protection. Future enhancements like encryption and cloud integration, this solution holds promise for wide-scale application in advanced smart building infrastructures.

References

1. Daqiang Zhang, Laurence Tianruo Yang, Min Chen, Shengjie Zhao, Minyi Guo, and Yin Zhang, "Real-Time Locating Systems Using the Active RFID for the Internet of Things", IEEE Systems Journal, Vol. 10, No. 3, September 2016.
2. A. P. Nirmala, V. Asha, Paramita Chandra. IoT based Secure Smart Home Automation System. 2022 IEEE Delhi Section Conference (DELCON), 978-1-6654-5883-2, 2022, IEEE,2022.
3. Xuemei Li et al., RFID Based Smart Home Architecture for improving lives, China Postdoctor Foundation with No. 20070410827, Guangdong Sci. & Tech. Plan with No. 20050201060 / no. 2007B080701002, SZ Sci. & Tech. Plan with No. QK 20060121 and Opening Foundation for Shenzhen key lab of mould advanced ma

4. Silpa Krishnan, Anjana M. S., Sethuraman N. Rao. Security Considerations for IoT in Smart Buildings. Amrita Center for Wireless Networks & Applications (AmritaWNA), 978-1-5090-6621-6, 2017 IEEE
5. Olutosin Taiwo , Absalom E. Ezugwu , Olaide N. Oyelade, Enhanced Intelligent Smart Home Control and Security System Based on Deep Learning Model. Hindawi Wireless Communications and Mobile Computing Volume, Article ID 9307961, 22 pages. <https://doi.org/10.1155/2022/9307961.2022>.
6. Gozde Dinc, Ozgur Koray Sahingoz, Smart Home Security with the use of WSNs on Future Intelligent Cities. 978-1-7281-1315-9, 2019, IEEE
7. Sara Amendola, Rossella Lodato, Sabina Manzari, Cecilia Occhiuzzi, and Gaetano Marrocco, "RFID Technology for IoT-Based Personal Healthcare in Smart Spaces", IEEE Internet of Things Journal, Vol. 1, No. 2, April 2014.
8. Yuvraj Agarwal and Anind K. Dey, "Toward Building a Safe, Secure, and Easy-to-Use Internet of Things Infrastructure", IEEE Computer Society, April 2016.
9. Loebbecke, C.; Huyskens, C.; Gogan, J. Emerging technologies in the service sector: An early exploration of item-level RFID on the fashion sales floor. In Proceedings of the International Working Conference on Information Technology in the Service Economy—Challenges and Possibilities for the 21st Century, Toronto, ON, Canada, 10–13 August 2008; Springer: Berlin/Heidelberg, Germany; Volume 267, pp. 189–198, 2008.
10. Sharma, D.; Mahto, R.; Harper, C.; Alqattan, S. Role of RFID technologies in transportation projects: A review. *Int. J. Technol. Intell. Plan*, 12, 349–377, 2020.
11. Li, Q.S.; Xu, X.L.; Chen, Z. PUF-based RFID Ownership Transfer Protocol in an Open Environment. In Proceedings of the 15th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Hong Kong, China, IEEE: Piscataway, NJ, USA, 2014; pp. 131–137, 9–11 December 2014.
12. Cheng, S.; Varadharajan, V.; Mu, Y.; Susilo, W. A secure elliptic curve-based RFID ownership transfer scheme with controlled delegation. *Cryptol. Inf. Secur. Ser*, 11, 31–43, 2013.
13. Chien, H.Y. De-synchronization Attack on Quadratic Residues-based RFID Ownership Transfer. In Proceedings of the 10th Asia Joint Conference on Information Security (AsiaJICIS 2015), Kaohsiung City, Taiwan; IEEE: Piscataway, NJ, USA, 2015; pp. 42–47, 24–26 May 2015.
14. Wang, H.; Yang, X.; Huang, Q.; Long, K. A Novel Authentication Protocol Enabling RFID Tags Ownership Transfer. In Proceedings of the 14th IEEE International Conference on Communication Technology (ICCT), Chengdu, China; IEEE: Piscataway, NJ, USA; pp. 855–860, 09–11 November 2012.
15. Chien, H.Y.; Huang, C.W. Security of ultra-lightweight RFID authentication protocols and their improvements. *Acm SIGOPS Oper. Syst. Rev*, 41, 83–86, 2007.
16. Huggins, J. S. RFID Handbook: Applications, Technology, Security, and Privacy. Springer, 2017.
17. Allan, A. Programming the ESP32: Getting Started with the Espressif ESP32 Development Board. O'Reilly Media, 2019.
18. Kamal, R. Internet of Things: Architecture and Design Principles. McGraw Hill Education, 2020.
19. Srivastava, S., & Sinha, A., "A Study on IoT Device Identity Management and Security," International Journal of Computer Applications, Vol. 180, Issue 47, 2023
20. Espressif Systems, "ESP32 Technical Reference Manual," <https://www.espressif.com>, Accessed October 2024.
21. Xin, W.; Guan, Z.; Yang, T.; Sun, H.; Chen, Z. An efficient privacy-preserving RFID ownership transfer protocol. In Proceedings of the 15th Asia-Pacific Web Conference

- on Web Technologies and Applications, Sydney, Australia; Volume 7808 LNCS, pp. 538–549, 4–6 April 2013.
22. Niu, H.; Taqieddin, E.; Jagannathan, S. EPC Gen2v2 RFID Standard Authentication and Ownership Management Protocol. *IEEE Trans. Mob. Comput.*, 15, 137–149, 2016.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

