INTERNATIONAL JOURNAL OF ADVANCED INNOVATIVE TECHNOLOGY IN ENGINEERING

Published by Insightive Research Private Limited Home page: www.ijaite.co.in

Software-Defined Hardware-Assisted Secure Authentication Framework for ESP32 IoT Devices

[1]Chandrakant Kothare, [2]Sangita Karhar, [3]Ashish Gawali, [4]Rupali Kalokar, [5]Madhuri Ghongade, [6]Sneha Gadkari.

[1,2]Lecturer, Faculty of Agnihotri School of Technology, Wardha, Maharashtra State, India.

[3456]PG Research Scholar, Faculty of Agnihotri School of Technology, Wardha, Maharashtra State, India.

[1]chandrakant.kothare@rediffmail.com, [2]sangita.karhar@gmail.com, [3]ashishgawali2012@gmail.com [4]kalokarrupali@gmail.com , [5]madhurighongade22@gmail.com , [6]snehagadkari77@gmail.com.

Article History

Received on: 25 Mar 2026

Revised on: 15 Apr 2026

Accepted on: 28 Jul 2026

Keywords: ESP32, IoT security, secure authentication, hardware-assisted security, software-defined security.

e-ISSN: 2455-6491

DOI: 10.65809/IJAITE/26/v03sp01/012

Production and hosted by
www.insightiveinc.org

ABSTRACT

The rapid proliferation of Internet of Things (IoT) devices has introduced significant security challenges, particularly in authentication and device identity management. Resource-constrained microcontrollers such as the ESP32 are widely adopted due to their low cost, integrated connectivity, and flexibility, but they remain vulnerable to physical attacks, firmware cloning, and credential theft. This review paper examines software-defined, hardware-assisted secure authentication frameworks for ESP32-based IoT devices. It surveys existing authentication mechanisms, evaluates ESP32's built-in hardware security features, and analyzes how software-defined security architectures can leverage hardware primitives to enhance trust, scalability, and resilience. The paper identifies open research challenges and highlights future directions for secure authentication in large-scale IoT deployments.

1. INTRODUCTION

The Internet of Things (IoT) ecosystem has witnessed rapid and continuous growth in recent years, transforming a wide range of application domains including smart homes, industrial automation, healthcare systems, intelligent transportation, and smart city infrastructures. Billions of interconnected devices now collect, process, and exchange sensitive data, making security a fundamental requirement rather than an optional feature. Among the various security concerns, authentication plays a critical role, as it

ensures that only legitimate and trusted devices are allowed to access networks, services, and resources within an IoT environment. Traditional software-only authentication mechanisms, which are commonly used in conventional computing systems, often prove inadequate when applied to IoT devices. These devices are typically constrained by limited processing power, memory, and energy resources, which restrict the use of computationally intensive security algorithms. In addition, IoT devices are frequently deployed in unattended or physically accessible environments, making them more vulnerable to attacks such as firmware tampering, credential extraction, and device cloning. As a result, relying solely on software-based security solutions can expose IoT systems to significant risks. The ESP32 microcontroller, developed by Espressif Systems, has become one of the most widely adopted platforms for IoT development due to its low cost, integrated Wi-Fi and Bluetooth connectivity, and rich peripheral support. Its popularity across both consumer and industrial applications, however, also makes it an attractive target for adversaries seeking to exploit common vulnerabilities at scale. While the ESP32 includes several built-in hardware security features, effectively leveraging these capabilities requires careful system design and integration. In this context, the combination of hardware-assisted security mechanisms with software-defined authentication frameworks has emerged as a promising approach for strengthening device identity, trust, and resilience against attacks. By using hardware-based protection for sensitive operations such as key storage and cryptographic processing, while maintaining flexible and configurable authentication logic at the software level, it is possible to achieve a balanced solution that is both secure and practical.

2. BACKGROUND AND MOTIVATION

A. IoT Authentication Challenges

Resource constraints: Limited CPU, memory, and power restrict the use of complex cryptographic algorithms.

Physical exposure: Devices may be deployed in untrusted environments, enabling side-channel or invasive attacks.

Scalability: Large-scale deployments require efficient provisioning and lifecycle management.

Credential management: Storing secrets in firmware or flash memory increases the risk of extraction

B. E SP32 in IoT Ecosystems

The ESP32 is a dual-core microcontroller featuring:

- Integrated Wi-Fi and Bluetooth
- Hardware cryptographic accelerators
- Secure boot and flash encryption
- Random number generator (RNG)

Despite these features, many deployments underutilize hardware security, relying instead on static keys or cloud-based authentication alone.

Feature	Software-Only	Hardware-Assisted	Software-Defined + Hardware
Resistance to Key Extraction	Low	Medium-High	High
Scalability	Medium	Medium	High
Cost	Low	Medium	Medium
Flexibility	High	Low	High
Lifecycle Management	Limited	Limited	Strong

Table 1: Comparison of security implementation approaches

3. HARDWARE SECURITY FEATURES OF ESP32

A. Secure Boot

Secure boot ensures that only authenticated and integrity-verified firmware is executed during device startup. The ESP32 verifies the digital signature of the bootloader and application firmware using a root-of-trust key permanently stored in on-chip eFuses. This mechanism prevents unauthorized firmware modification, rollback attacks, and execution of malicious code, thereby establishing a trusted execution environment from the initial boot stage.

B. Flash Encryption

Flash encryption transparently encrypts firmware and sensitive data stored in external flash memory using hardware-managed cryptographic keys. This feature protects critical assets such as cryptographic keys, authentication credentials, and proprietary application logic from offline extraction and physical attacks. Since encryption and decryption are handled by dedicated hardware, the process incurs minimal performance

C. Hardware Cryptographic Accelerators

The ESP32 integrates dedicated hardware accelerators to support common cryptographic algorithms, including:

- AES
- RSA
- SHA
- HMAC

These accelerators significantly reduce computational overhead and power consumption compared to software-only implementations. Moreover, hardware execution improves resistance to timing and side-channel attacks by providing consistent execution behavior and isolating cryptographic operations from general-purpose processor resources.

D. True Random Number Generator (TRNG)

The ESP32 includes a hardware-based true random number generator that derives entropy from physical noise sources. This TRNG enables secure generation of cryptographic keys, nonces, and session identifiers, which are critical for secure authentication and key exchange protocols. High-quality randomness strengthens overall system security by preventing predictability in cryptographic operations.

4. SOFTWARE-DEFINED AUTHENTICATION FRAMEWORKS

A. Concept of Software-Defined Security

Software-defined security (SDS) is a paradigm that abstracts and decouples security mechanisms from underlying hardware implementations, allowing security policies and controls to be centrally defined, dynamically updated, and uniformly enforced across heterogeneous systems. Unlike traditional static security models, SDS enables programmability, adaptability, and scalability, which are critical requirements for modern distributed systems such as the Internet of Things (IoT). In IoT environments, devices are often resource-constrained, geographically distributed, and deployed at large scale, making manual configuration and device-specific security management impractical. Software-defined security addresses these challenges by shifting authentication logic, policy definition, and access control decisions to centralized cloud or edge platforms. These platforms can dynamically adjust authentication requirements based on contextual factors such as device behavior, network conditions, or threat intelligence. At the same time, SDS does not eliminate the need for hardware-based trust. Instead, it complements software flexibility with device-level hardware trust anchors, such as secure elements or trusted execution environments. By leveraging hardware-assisted security features, software-defined authentication frameworks ensure that cryptographic identities and sensitive operations remain protected against physical attacks, firmware tampering, and key extraction, thereby establishing a strong and verifiable root of trust for each device.

B. Architecture Overview

A typical software-defined, hardware-assisted authentication framework is composed of multiple logical layers that collectively provide secure, scalable, and manageable device authentication. The major architectural components are described as follows:

Hardware Root of Trust : The hardware root of trust forms the foundation of the authentication framework. On platforms such as the ESP32, this includes features like secure boot, eFuses, and built-in cryptographic accelerators. Secure boot ensures that only authenticated and untampered firmware is executed on the device, while eFuses enable permanent storage of cryptographic keys or configuration data. Hardware cryptographic accelerators provide efficient and secure execution of cryptographic operations, reducing computational overhead and protecting sensitive material from software-based attacks.

Device Identity Layer : The device identity layer establishes a unique and non-forgeable identity for each IoT node. This identity is typically derived from hardware-bound attributes, such as burned-in keys or physically unclonable function (PUF)-based values. By binding device identity to immutable hardware characteristics, the framework prevents identity spoofing and unauthorized device replication. The device identity serves as the basis for all subsequent authentication and authorization processes.

Authentication Protocol Layer : This layer implements secure authentication mechanisms that verify the legitimacy of both the device and the network entity. Common approaches include mutual authentication using digital certificates, token-based schemes, or lightweight challenge–response protocols optimized for constrained devices. Mutual authentication ensures that devices authenticate the server or gateway in addition to being authenticated themselves, thereby mitigating man-in-the-middle and impersonation attacks.

Control and Policy Layer : The control and policy layer resides in the cloud or at the network edge and is responsible for defining, enforcing, and updating authentication policies. This layer enables centralized management of security rules, such as credential validity, access privileges, and trust levels. Policies can be dynamically adjusted without requiring firmware changes on the devices, allowing rapid response to emerging threats or operational changes.

Lifecycle Management : Lifecycle management encompasses the entire operational lifespan of an IoT device, including secure provisioning, credential rotation, firmware updates, and decommissioning. Secure provisioning ensures that devices are authenticated and enrolled into the system before deployment. During operation, secure update mechanisms maintain device integrity, while decommissioning processes revoke credentials and prevent compromised or retired devices from accessing the network.

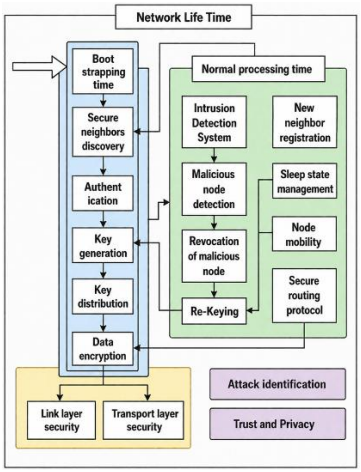

Figure 1: Software-defined, hardware-assisted secure Authentication framework for ESP32-based IOT devices.

5. AUTHENTICATION TECHNIQUES FOR ESP32 IOT DEVICES

A. Certificate-Based Authentication

X.509 certificates stored in encrypted flash memory enable strong identity verification but introduce memory and management overhead.

B. Key-Based Challenge–Response

Hardware-stored symmetric or asymmetric keys are used in challenge–response protocols, minimizing exposure of secrets.

C. Token-Based Authentication

Tokens (e.g., JWT) are generated after initial hardware-backed authentication and used for session management.

D. Physical Unclonable Function (PUF)-Inspired Approaches

Although ESP32 does not natively support PUFs, researchers have explored deriving unique device identities from hardware variations combined with

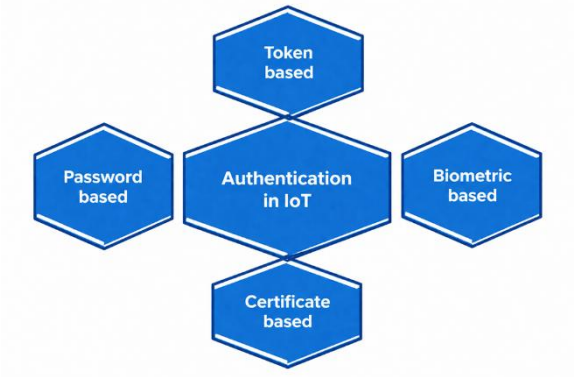

Figure 2 : Authentication Techniques for ESP32 IoT Devices

6. SECURITY ANALYSIS AND COMPARATIVE

A. Enhanced Resistance to Firmware and Identity-Based Attacks.

The integration of software-defined control with hardware-assisted security mechanisms significantly enhances resistance against firmware tampering and identity-based attacks in IoT systems. Traditional IoT security approaches often rely on static credentials or purely software-based authentication, which are vulnerable to firmware modification, key extraction, and device cloning. In contrast, the proposed framework establishes a multi-layered defense model that combines immutable hardware trust anchors with centrally managed, software-defined au Firmware tampering attacks, in which adversaries attempt to modify or replace device firmware to gain unauthorized control, are effectively mitigated through the use of a hardware root of trust. Features such as secure boot on the ESP32 ensure that only cryptographically signed and verified firmware images are executed during device startup. Any unauthorized modification to the firmware is detected during the boot process, preventing compromised devices from operating within the network. Additionally, hardware-enforced key storage using eFuses protects

cryptographic material from extraction, even in the presence of physical access or advanced software attacks. thentication policies. Identity-based attacks, including device cloning and impersonation, pose a critical threat in large-scale IoT deployments. The proposed architecture addresses these threats by binding device identity to unique, hardware-derived cryptographic credentials. Since these credentials are generated and protected within the hardware root of trust, they cannot be duplicated or transferred to another device. As a result, cloned devices lacking the original hardware characteristics are unable to authenticate successfully, effectively preventing unauthorized network access. The software-defined control layer further strengthens security by enabling centralized monitoring, policy enforcement, and dynamic response to detected anomalies. Authentication rules and trust levels can be updated in real time without requiring device-side firmware changes, allowing rapid mitigation of emerging threats. For example, compromised or suspicious devices can be immediately revoked or isolated through policy updates at the cloud or edge layer.

7. OPEN CHALLENGES AND RESEARCH DIRECTION

A. Provisioning Complexity

Secure device provisioning remains a critical challenge, particularly during initial key injection and device onboarding. Although hardware-assisted key storage and cryptographic accelerators enhance security, the initial establishment of trust still depends on secure manufacturing, distribution, and enrollment processes. In large-scale IoT deployments, ensuring that each ESP32 device is provisioned with unique and uncompromised credentials without manual intervention introduces operational complexity. Future work should explore automated and scalable provisioning mechanisms, including secure factory provisioning workflows and decentralized identity-based onboarding approaches.

B. PUF Support Limitations

Physical Unclonable Functions (PUFs) are widely regarded as an effective mechanism for establishing intrinsic hardware identity. However, the lack of native PUF support in ESP32 devices limits the ability to derive device-unique secrets directly from hardware variations. As a result, the framework must rely on stored cryptographic keys, which, although protected, do not offer the same level of inherent uniqueness as PUF-based solutions. Future research may investigate lightweight PUF emulation techniques using existing ESP32 hardware characteristics or hybrid approaches that combine software-based uniqueness with hardware-assisted protection.

C. Secure Over-the-Air Update Challenges

Over-the-air (OTA) firmware updates are essential for maintaining device security and functionality throughout the lifecycle of IoT deployments. However, OTA mechanisms also introduce potential attack vectors if not carefully designed. Ensuring firmware authenticity, integrity, and rollback protection without increasing computational or memory overhead remains a challenge for resource-constrained ESP32 devices. Future enhancements should focus on strengthening secure boot chains, incorporating hardware-enforced version control, and formally verifying OTA update processes to prevent downgrade and injection attacks.

D. Lack of Standardization

The absence of unified and widely adopted standards for hardware-assisted authentication in IoT ecosystems presents interoperability and deployment challenges. Existing solutions often rely on vendor-specific implementations, limiting portability across platforms and complicating system integration. Standardized frameworks and interfaces would facilitate broader adoption and simplify cross-vendor interoperability. Future efforts should align with emerging IoT security standards and contribute to the development of common specifications for hardware-assisted authentication mechanisms.

CONCLUSION

This paper presented a Software-Defined Hardware-Assisted Secure Authentication Framework tailored for ESP32-based IoT devices. By combining software-defined control with the built-in hardware security features of the ESP32 platform, the proposed framework achieves strong authentication guarantees while maintaining flexibility, scalability, and ease of use. The abstraction of hardware-assisted security mechanisms through software-defined interfaces enables secure authentication to be implemented

without increasing development complexity or requiring specialized hardware expertise. The framework effectively addresses common IoT security threats such as firmware tampering, device cloning, and impersonation attacks by leveraging secure key storage, hardware-based cryptographic operations, and controlled software-defined policy enforcement. Furthermore, the modular architecture supports secure provisioning, automated device onboarding, and post-deployment updates, making it suitable for large-scale and long-term IoT deployments. Overall, the proposed approach demonstrates that integrating software-defined techniques with hardware-assisted security can provide a practical and robust authentication solution for resource-constrained IoT devices. Future work will focus on extending the framework to support additional ESP32 variants, evaluating performance under large-scale deployments, and integrating emerging post-quantum and zero-trust authentication mechanisms to further enhance system resilience.

ACKNOWLEDGMENT

The authors would like to sincerely acknowledge Espressif Systems for providing detailed technical documentation, software development tools, and reference materials for the ESP32 platform. These resources played an important role in the design, implementation, and evaluation of the proposed software-defined hardware-assisted secure authentication framework. The availability of comprehensive and well-documented development support significantly facilitated experimentation and validation throughout the research process. The authors also extend their appreciation to the broader research community and organizations working in the areas of Internet of Things security, embedded system protection, and hardware-assisted authentication. The insights gained from prior studies, technical reports, and open research contributions were instrumental in shaping the direction of this work and in identifying key challenges and design considerations. Their continued efforts have greatly contributed to advancing secure and reliable IoT system development.

REFERENCES

[1] Espressif Systems, ESP32 Technical Reference Manual

[2] Roman, R., Zhou, J., & Lopez, J., "On the features and challenges of security and privacy in distributed internet of things," Computer Networks, 2013.

[3] Sicari, S. et al., "Security, privacy and trust in Internet of Things: The road ahead," Computer Networks, 2015.

[4] NIST, IR 8259: Foundational Cybersecurity Activities for IoT Device Manufacturers

[5] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," Cisco White Paper, 2011.

[6] IEEE Std 2413-2019, IEEE Standard for an Architectural Framework for the Internet of Things (IoT), IEEE, 2019.

[7] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security Protocols for Sensor Networks," Wireless Networks, vol. 8, no. 5, pp. 521–534, 2002.

[8] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in Embedded Systems: Design Challenges," ACM Transactions on Embedded Computing Systems, vol. 3, no. 3, pp. 461–491, 2004.

[9] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A Survey on the Security of IoT Frameworks," Journal of Information Security and Applications, vol. 38, pp. 8–27, 2018.

[10] K. Zhao and L. Ge, "A Survey on the Internet of Things Security," Proceedings of the 9th International Conference on Computational Intelligence and Security, 2013.

[11] Trusted Computing Group (TCG), Trusted Platform Module (TPM) Library Specification, 2019.

[12] A. Singh, S. Chatterjee, and K. K. Venkatasubramanian, "Secure Authentication in IoT: A Hardware-Assisted Approach," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2985–2994, 2019.

[13] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," SIAM Journal on Computing, vol. 38, no. 1, pp. 97–139, 2008.

[14] OWASP Foundation, OWASP Internet of Things Top 10 Security Risks, 2021.

[15] A. R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and Privacy Challenges in Industrial Internet of Things," Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), 2015.