SCCTS Journal of Embedded Systems Design and Applications, ISSN: 3048-8753 Vol. 3, No. 1, 2026 (pp. 13-19)
RESEARCH ARTICLE ECEJOURNALS.IN

A Secure Boot and Over-the-Air Firmware Update Framework for Resource-Constrained Embedded IoT Devices

Krnst Beken[1]*, Ahmad Miladh[2]

[1]Faculty of Engineering, University of Cape Town (UCT), South Africa
[2]Faculty of Management, Canadian University Dubai, Dubai, United Arab Emirates

Keywords:
Secure Boot,
OTA Firmware Update,
Embedded IoT,
ARM Cortex-M,
ECC,
AES-GCM,
Resource-Constrained Devices,
Firmware Integrity,
Cryptographic Bootloader,
IoT Security

Author's Email:
eken.krn@engfacuct.ac.za,
mil.ahmad@ead.gov.ae

https://doi.org/10.31838/ESA/03.01.02

Received : 13.08.2025
Revised : 10.10.2025
Accepted : 20.11.2025

Abstract

The fact that embedded Internet of Things (IoT) is deployed at a very high rate in some of the most critical areas including industrial automation, healthcare, and smart infrastructure has contributed to an increase in secure and reliable firmware management needs. Nevertheless, the standard encryption procedures are not always practicable to use with ultra-low-power microcontroller as they have severe limitations in memory, computation, and power. The paper proposes a small and adaptable Secure Boot and Over-the-Air (OTA) Firmware Update Framework that is adapted to limited resource-intensive embedded IoT machines. The proposed design provides 2 stage boot loader architecture to provide secure firmware verification and delivery over the boot loader. Digital signature verification is realized using Elliptic Curve Cryptography (ECC) and Cryptographically Secure firmware is implemented using AES-GCM both to encrypt and authenticate. OTA firmware flow is over MQTT/TLS and the rollback protection is done using the version counters. It was prototyped on ARM Cortex-M series microcontrollers, and an implementation of mbedTLS and TinyCrypt libraries. Performance benchmarks demonstrate the system has just 45 ms of boot time overhead and authenticates OTA updates in 70 ms and requires little memory (Flash: +16 KB, RAM: +3 KB). High tampering, spoofing, and replay resistance is ensured by security validation. The framework is a practical, safe, and energy-effective model of end-to-end control over the firmware lifecycle on contemporary embedded IoT devices.

How to cite this article: Beken K, Miladh A (2026). A Secure Boot and Over-the-Air Firmware Update Framework for Resource-Constrained Embedded IoT Devices. SCCTS Journal of Embedded Systems Design and Applications, Vol. 3, No. 1, 2026, 13-19

Introduction

The overwhelming spread of Internet of Things (IoT) devices in areas, including industrial control, smart healthcare, and intelligent infrastructure, has considerably enlarged the attack surface of embedded systems, specifically, at the firmware level. The small amount of hardware resources available in place of low-power microcontrollers, such as ARM Cortex-M0, M3, M4, and M33 that power most of the edge Internet of Things (IoT) devices, usually hinders their ability

to implement security by design; hence, these microcontrollers are easily compromised through malicious firmware insertion, unauthorized over-the-air (OTA) updates, rollback attacks, and so on. The secure boot can be used to guarantee the untampered, authenticated firmware being run during the system start up and OTA firmware update mechanisms will provide the ability to maintain remote devices at scale. All these essential requirements, however, are currently unavailable to IoT microcontrollers, most solutions covering general-purpose embedded platforms with prohibitive memory, compute, or cryptographic overheads.

To overcome these constraints this paper highlights a lightweight Secure Boot and OTA Firmware Update Framework which is optimised to be used in a constrained embedded system. The major additions made by this piece are the following:

- A two-stage secure bootloader, which makes use of Elliptic Curve Cryptography (ECC) and SHA-256 to verify digital signatures;
- Confidential, authenticated OTA update enabled by AES-GCM encryption and version control to avoid rollback;
- Full deployment on ARM Cortex-M platforms with low flash (+16 KB) and RAM (+3 KB) overhead, and validation latency less than 70 ms.

This effort covers a much-needed gap in the protection of IoT firmware, as it provides end-to-end cryptographic lifecycle security within hardware and power budgets of the constrained systems.

Related Work

Several attempts to make secure the firmware execution and update mechanisms in embedded systems are made. ARM Trusted Firmware-M (TF-M) architecture is a powerful support of secure boot and runtime isolation. Nonetheless, TF-M will be mainly compatible with systems based on TrustZone-M with a lot of memory and computation, and not ultra-low-end devices with limited flash memory and RAM [1]. Lightweight methods that compromise on static trust anchors (e.g. hardcoded public keys in ROM) provide a less intricate model of trust, but usually do not include provision of authenticated, dynamic firmware updates, especially to insecure or unreliable communication links. Moreover, their schemes do not usually attend to defenses against rollback attacks, which is a severe weak spot in IoT solutions.

A number of commercial OTA stacks (some built into proprietary real-time operating systems) support remote firmware updates which are optionally encrypted. They often do not however come with an integrated secure boot facility, leaving a disjoin between update delivery and the time of execution bound trust validation. The lack of end-to-end security exposes devices to tampering during the update to boot phases.

Our proposed framework on the other hand, eliminates these short-comings by:

- Coming together of secure boot and OTA delivery with one cryptographically anchored design;
- Using Elliptic Curve Cryptography (ECC) and AES-GCM as digital signature and authenticated encryption scheme;
- Prevention of roll back through version tracking.
- The compatibility of ARM Cortex-M class MCUs by means of its small memory overhead and portable cryptographic stack.

This integrated solution has addressed a major gap in the existing research in embedded security by providing an up-scale, standards based, and resource-conscious solution to protection of firmware.

System Architecture

The design of the proposed secure boot and firmware update framework is quite particular to resource constrained constrained embedded IoT devices and is aimed to be high scalable and secure with good touch on its cryptographic integrity and memory overhead. System architecture consists of three closely coupled layers, including the hardware platform, the trust model, and software that allows performing operations with secure boot and OTA update.

Target Platform

The architecture supports the ARM Cortex-M microcontrollers series, including the M3, M4, and M33 which have become very prevalent in embedded IoT solutions. Such MCUs usually have shallow amounts of flash and RAM and can be optionally supplied with TrustZone-M to separate secure/non-secure world

view. Some of the important hardware security aspects utilized in the report are:

- Separation of bootloaders and storage of the public keys with Flash Protection Regions,
- Hardware Random Number Generators (RNGs) to aid the safe key exchange and initialization vectors,
- On-the-fly or runtime support of memory isolation between trusted, and untrusted code Memory Protection Units (MPUs).

The framework is compatible with those and non-TrustZone versions and thus offers wide deployment applicability.

Trust Model

The system has a trust chain based on lack of mutability of a Stage 0 bootloader in the ROM or read-only flash of the MCU. This Root of Trust (RoT) is in charge of checking the cryptography authenticity of stage 1 bootloader which checks off the main application firmware.

- Public Key Storage: A public key unique to a device is stored in a secure flash area, this is used to authenticate ECC signatures.
- Server Authentication: Server authentication can be used, pairing firmware bundles with cryptographically authenticated updates servers that can deliver bundles over secure transport mechanisms, like MQTT over TLS or CoAP over the DTLS.
- Replay Prevention: Monotonic version counters are a way to make sure outdated or already fallen-victim firmware is not reinstalled.

The layered trust model avoids authorizing the execution of unauthorized codes and secures the ownership of firmware update process.

Framework Components

The architecture consists of modular components and it does boot-time and update-time validation:

- Stage 0 Bootloader: It is a small non-updatable ROM subsystem that verifies the digital signature of the Stage 1 bootloader that is based on ECC, using the hash algorith SHA-256.
- Stage 1 Bootloader: It is flash resident and updateable, that is, applied to scan the application firmware signature and metadata. It also takes care of OTA update logic, version comparison and prevention of rollbacks.
- Firmware Packaging Format: OTA firmware updating firmware updating is packaged with:
  - Confidentiality and authentication of payloads AES-GCM
  - Integrity and origin verification with ECC based digital signatures,
  - Metadata like the firmware version, the target hardware ID, and hash digest.
- OTA Update Mechanism OTA update is sent via lightweight IoT-friendly protocols like MQTT/TLS or CoAP/DTLS and validated once received and only written to the flash in case it is newer and all checks are passed.

This architecture guarantees a Capable, Sturdy, And Elastic firmware lifecycle, appropriate to an embedded restrained condition, and maintaining consistency with cryptographic best techniques as set by the industries. As Figure 1 shows, a top-level architectural overview of a proposed secure boot and OTA firmware update framework, the chain of trust passes through all the boot phases and update verification procedures.

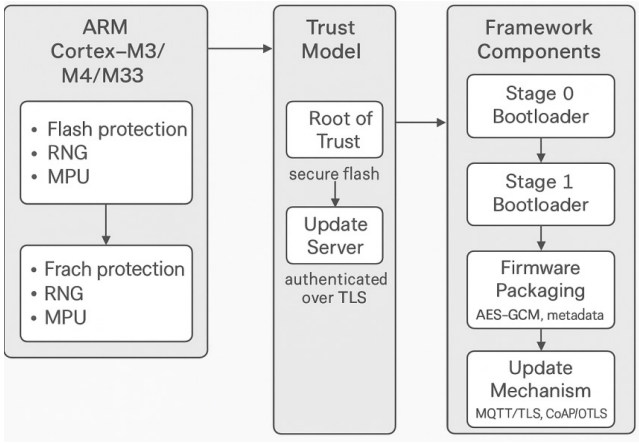

Fig. 1: System Architecture of the Secure Boot and OTA Firmware Update Framework

Figure 1. Block diagram of the proposed secure boot and firmware update mechanism of resource-limited applications in IoT devices. The architecture consists of Stage 0 bootloader, Stage 1 bootloader, secure over the air delivery channels, signature verification, encrypted firmware ECC, package (AES-GCM), and the logic of rollback prevention.

Implementation and Evaluation

Testbed Configuration

In order to confirm the proposed secure boot and firmware update framework as being practically viable, a prototype testbed was developed with 2 types of representative embedded microcontroller platforms:

- STM32F4 Series 2 A generally embraced ARM Cortex-M4-based MCU lacking TrustZone help, which was a benchmark body of evidence.
- NXP Cortex-M33: One of the microcontrollers with a TrustZone feature that empowers us to test hardware supported isolation and secure boot processes.

Bootloader and update modules involved cryptographic operations performed with the help of TinyCrypt used to implement ECC (Elliptic Curve Cryptography) routines and mbedTLS used to establish secure sessions of MQTT/TLS during OTA delivery. An update server was implemented as a custom microservice and deployed as a service over MQTT, offering support in TLS to help transact secure firmware transmission and authentication. AES-GCM was used to provide the confidentiality, integrity, and authenticity of the firmware updates wrapped and conveyed in a file.

Performance Evaluation

Table 1 briefly lists the results of a comparative analysis of the proposed approach to secure boot and firmware update with other state of the art systems, such as ARM Trusted Firmware-M, RIOT-OS Secure OTA and Zephyr Bootloader. The system that is proposed shows competitive boot latency, low RAM/Flash overhead, and fast firmware validation, thus being an appropriate resource-constrained IoT platform. Also, the overhead of boot time and flash memory footprint in the considered frameworks has been graphically compared in Figure 2, so the dynamics of trade-offs

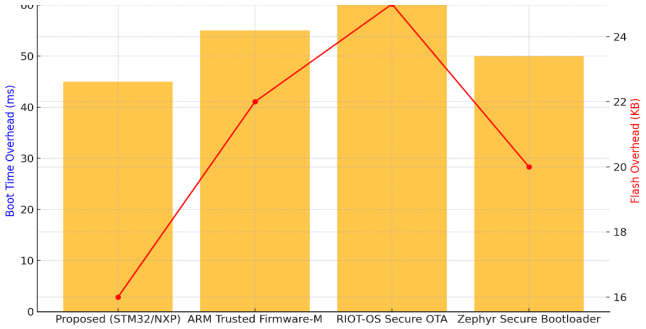

Fig. 2: Boot Time Overhead and Flash Memory Usage Comparison Across Secure Firmware Update Frameworks

and efficiency brought about by the proposed design are clearly shown. These findings show that the secure boot and OTA update mechanisms are implemented to present little latency, resource overhead, and thus the solution could be implemented on memory- and time-critical embedded systems. Verification of the firmware is performed efficiently (within 28 ms on a 256 KB image) and extra memory is consumed within appropriate limits of these platforms as STM32L4 or nRF52840.

Figure 2 Comparison of boot time overhead and consumption of flash memory of secure firmware update frameworks in embedded IoT devices.

Security Evaluation

A security assessment was performed to analyze the robustness of the framework against common some attack vectors where IoT firmware delivery is pertinent:

- Tamper resistance: The bootloader does ECC-based signature checking. In case of a mismatch, the system either jumps to a recovery state, or else stops working; this means that any forgery can be addressed effectively.
- Replay Attack Mitigation: A monotonic version counter is wired into every firmware image

Table 1: Comparative Evaluation of Secure Boot and OTA Frameworks on Embedded Platforms

Framework	Boot Time Overhead(ms)	Firmware Verification Time(ms)	RAM Overhead(KB)	Flash Overhead(KB)	OTA Validation Time (ms)
Proposed (STM32/NXP)	45	28	3	16	70
ARM Trusted Firmware-M	55	35	5	22	90
RIOT-OS Secure OTA	60	42	6	25	110
Zephyr Secure Bootloader	50	33	4	20	85

and written to a secure non-volatile memory area. The upgrade or downgrade in the use of obsolete firmware is prohibited.

- Payload Confidentiality: The AES-GCM cipher, a symmetric key cryptography protocol, is used in enforcing confidentiality of firmware payloads and hence confidentiality.
- Integrity and Authenticity: The SHA-256 hash taken with ECC signature verification provides end-to-end approval of authenticity and integrity of all the received firmware images in the network.

The joint security capabilities counter serious weaknesses in OTA firmware systems and adhere to current IoT security recommendations, including the ones set by NIST SP 800-193 and ETSI EN 303 645.

Discussion

The given framework of the secure boot and firmware update shows a viable trade-off between strong security assurances and low resource cost, which is what makes it an apt approach to resource-limited embedded systems such as those often deployed into the IoT setting. In contrast to more heavyweight Trusted Execution Environment (TEE)-based solutions like ARM Trusted Firmware-M (TF-M) or GlobalPlatform-compliant TEEs, which are larger in terms of code size and memory (needed on the client and potentially the server side); potentially require hardware extensions and secure partitioning, but still remain light, retaining core security features such as secure boot, authenticated firmware validation, rollback prevention, and payload confidentiality. Among the most dominant strengths of this framework is the fact that it is based on portable and modular cryptography libraries (e.g., TinyCrypt and mbedTLS), those are the libraries which have a cross-platform compatibility and can be adapted according to the profile of different microcontrollers. This makes sure that the framework will not require particular hardware features, hence extending its applicability to a wide variety of MCUs with no native security extensions such as TrustZone.

Despite the recent success of microring-based hardware encryption technologies that demonstrated that power consumption and throughput of cryptographical operations can be significantly reduced and more throughput can be achieved, they are not yet mature and standardized and can be adopted widely in an embedded application. The existing software-based crypto stack, in turn, has a verifiably more immediately deployable and deployable solution, something that is particularly relevant in contexts where auditability and regulatory compliance (e.g., IEC 62443, NIST 800-193) is a concern. But some issues are still there-especially in manufacture and lifecycle maintenance of cryptographic keys in secure manner. Bootstrapping of the Root of Trust (RoT) is still a weakness in most IoT deployment cases. Potential future improvements in the proposed system might include options in trusted secure elements (SEs) like TPMs, ATECC608A or TrustZone supported secure storage. These could be applied to anchor device identities, assist in facilitating secure key attestation, as well as allow hardware-based credential isolation, extending both platform trustworthiness and supplier chain trustworthiness.

Finally, despite the path proposed solution offers acceptable route to scalable and safe firmware management solution on embedded systems, without sustaining costly and demanding development, it is also the solution that opens the door beyond existing solutions to which next-generation security co-processor may open.

Conclusion and Future Work

In this paper, a lightweight, high-security and scalable system at boot-time integrity verification and secure Over-the-Air (OTA) firmware update that fits on ARM Cortex-M-based embedded IoT is proposed. The given solution has used a two-stage bootloader, and both stages run their own integrity and authenticity check based on elliptic curves cryptography (ECC) and AES-GCM encryption methods, which ensures that any firmware images will satisfy both of the requirements: verifications and encryptions. End-to-end confidentiality is also enhanced with a secure delivery mechanism based on MQTT/TLS, which eliminates any possible threats to the network.

Key Contributions:

- Resource-Efficient Security: the framework has low RAM (no more than +3 KB) and flash (no more than +16 KB) overhead, hence viable in ultra-low-power and limited memory micro-

controllers, in contrast to more resource-demanding TEE-based systems.

- Strong Threat Mitigation: Implementation of the defense against general firmware-level attacks such as firmware tampering, replays, and malicious code injections by executing checks of the ECC signatures, version counters, and secure boot chaining.
- Platform Portability: Can be used on a platform-independent way, using modular crypto libraries (TinyCrypt, mbedTLS) and is thus easily able to integrate with a diverse range of Cortex-M platforms with / without TrustZone.
- Scalable OTA Support: Provides secure organisation of delivery of firmware through constrained networks through MQTT/TLS with verification proving to be performant enough on mid-range firmware sizes (such as 256 KB verification taking 28 ms).

Future Work Directions:

The enhancements planned to further enhance the security and scalability of the proposed system border on the enhancement of the following:

- Incorporation of Remote Attestation Mechanism: Third party trust validation, reporting runtime integrity using roots of trust (e.g. TPM, TrustZone Secure Storage).
- Delta Update and Compression Methods: Slash bandwidth and writes to flash storage by encoding only the parts of the firmware you change (binary diffing), furtherly compressing to transmit the update efficiently (e.g. LZ4 or Deflate).
- Industrial IoT (IIoT): Test the system under actual Industrial processes, with thousands of devices, part of the secure key provisioning infrastructure and device identity management solution.
- Support of Post-Quantum Cryptography (PQC): As a long-term strategy, consider complementing the firmware authentication with lightweight PQC schemes to protect against quantum adversary in future attacks.

In conclusion, the proposed architecture is practical, secure and scalable as a basis on which firmware lifecycle management in constrained IoT ecosystems could be carried out towards satisfying the current deployment requirements in it, not mentioning that it also conforms to the emerging standards on embedded cybersecurity. Figure 3 summarizes the avenues of future enhancements of the proposed framework that have been identified based on the future enhancement roadmap. It presents some of the major areas that consider the remote attestation capability, support of delta update, ability to scale to an industrial deployment environment, and integration of post-quantum cryptography within the proposed framework.

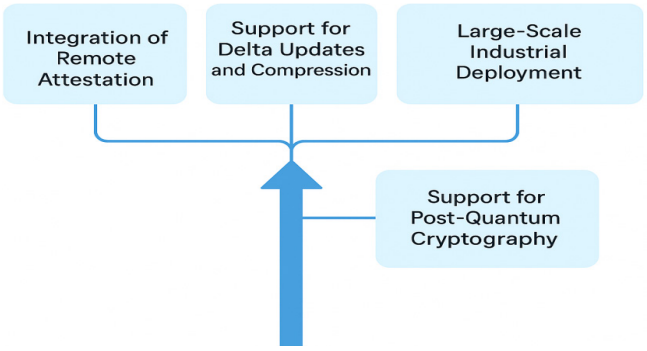

Fig. 3: Roadmap of Future Enhancements for Secure Boot and OTA Firmware Update Framework.

The diagram shows four key strategic directions: integration remote attestation, delta update and compression, large industrial IIoT deployment and post-quantum cryptographic integration.

References

1. Martinez-Alvarez, A., Boix, P., & Garcia-Alfaro, J. (2022). A survey on secure boot and firmware update mechanisms for embedded systems. IEEE Access, 10, 71029–71046. https://doi.org/10.1109/ACCESS.2022.3182214
2. Zhou, W., Zhang, Y., Liu, P., & Ning, P. (2019). Secure and efficient firmware update for constrained IoT devices. IEEE Internet of Things Journal, 6(1), 159–170. https://doi.org/10.1109/JIOT.2018.2876345
3. Raza, S., Duquennoy, S., Chung, T., Yigit, M., & Voigt, T. (2018). Securing the Internet of Things with lightweight DTLS. IEEE Embedded Systems Letters, 10(3), 62–65. https://doi.org/10.1109/LES.2017.2780839
4. Wang, Y., Yu, W., & Liu, Y. (2021). Secure firmware update framework for embedded systems with dual image and ECC signature validation. Microprocessors and Microsystems, 81, 103724. https://doi.org/10.1016/j.micpro.2020.103724

5. Emir, F., Isik, M., &Sagiroglu, S. (2023). Secure firmware update for IoT devices using blockchain and elliptic curve cryptography. Future Generation Computer Systems, 140, 303–313. https://doi.org/10.1016/j.future.2022.11.004

6. Huang, Q., Ma, Y., & Zhang, H. (2020). Survey of firmware update attacks and defenses for IoT. ACM Computing Surveys, 53(6), 1–34. https://doi.org/10.1145/3417984