SCCTS Journal of Embedded Systems Design and Applications, ISSN: 3048-8753 Vol. 3, No. 1, 2026 (pp. 66-72)
RESEARCH ARTICLE ECEJOURNALS.IN

A RISC-V Based Embedded Security Architecture for Trusted Execution in Industrial Control Systems

Anuradha K. Madugalla[1]*, Hee-Seob Kim[2]

[1]Department of Electrical Engineering Faculty of Engineering, University of Moratuwa Moratuwa, Sri Lanka
[2]Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, Korea

Keywords:
RISC-V,
Trusted Execution,
Embedded Security,
Industrial Control Systems,
Root of Trust, Secure Boot,
Hardware Security

Author's Email:
k.anuradha@elect.mrt.ac.lk,
h.s.kim@snu.ac.kr

https://doi.org/10.31838/ESA/03.01.08

Received : 12.08.2025
Revised : 22.10.2025
Accepted : 18.11.2025

Abstract

Industrial Control Systems (ICS) have become more vulnerable to cyber threats as they have emerged in the digital world, in particular, embedded environments that face limits to power, cost, and latency. A new embedded security architecture has been presented in this paper based on the open-source RISC-V instruction set architecture (ISA) to support trusted execution within the ICS setting. The first goal is to be able to create a low-overhead, transparent, and scalable system, which is enforcing hardware-based trust but without affecting the real-time performance. The architecture proposed incorporates a lightweight hardware Root of Trust (RoT), secure bootloader, memory protection units (MPUs) and minimal Trusted Execution Environment (TEE) as part of a bespoke RISC-V core. Important architectural updates are register-level TEE isolation, crypto-integrity verification, and hardware-accelerated AES-GCM and SHA-256. The system is run on a 32-bit RV32IM RISC-V and tested against real-time industrial protocols (Modbus, OPC-UA) and code injection, denial-of-service emulations. Through experimental evaluations, it is shown how the architecture is able to attain a security enhancement score of 92.4 % and blocks more than 91 % of the injected threats but without much overhead (execution time overload of 3.9-4.7 %). These results attest the feasibility of code-attestable, hardware-based security assimilation into ICS embedded systems, which makes a potential gateway to open and trusted, scale-able industrial automation.

How to cite this article: Madugalla AK, Kim H (2026). A RISC-V Based Embedded Security Architecture for Trusted Execution in Industrial Control Systems. SCCTS Journal of Embedded Systems Design and Applications, Vol. 3, No. 1, 2026, 66-72

Introduction

These critical infrastructure sectors include energy, manufacturing, water management and transport systems, where Industrial Control Systems (ICS) are installed on the very foundations of those infrastructures. Since these systems are becoming more interconnected more to IoT devices and digital communication networks to enhance edge intelligence, process automation and remote operations, they have become quite vulnerable to advanced cyber threats. It is important to note that the threats like code injection, data tampering, denial-of-service (DoS) and firmware manipulation have been reported in a range of ICS environments where low-level firmware and control interfaces are frequently exploited.[1] Nonetheless,

conventional cyber protection platforms, including software firewalls, all the way to virtualized Trusted execution environments (TEE) do not match well with the real-time low-power requirements of embedded ICS units. Furthermore, such traditional solutions are either vendor-locked in terms of hardware or lacking in the transparency department, or prone to induce both computational and memory overhead, a violation of the design budget of embedded microcontrollers. Although some of the existing researches have investigated ARM TrustZone, TPM-based security, and software TEEs, their implementation lacks open-source extensibility and fine-grained control of resources that deterministic control systems demand.[2] New RISC-V based efforts including Keystone and Sanctum, have added secure enclaves, but due to their general-purpose OS requirements and execution constraints related to non-deterministic execution, they are not deployed to ICS.

In the current paper, these limitations are discussed by proposing an embedded security architecture based on RISC-V supporting a lightweight hardware Root of Trust (RoT), memory protection units (MPUs), secure boot, and a minimal TEE implemented directly into the RISC-V execution pipeline. This design is intended to support trusted execution at very low cost to the performance of the system and is optimized to support real-time industrial workloads.

Related Work

The embedded security in Industrial Control Systems (ICS) has undergone a significant change as cryptography modules,0 secure execution environment and techniques of validating firmware are also incorporated. ARM TrustZone, Trusted Platform Module (TPM) and software based Trusted Execution Environment (TEE) have gained large adoption in conventional system architectures to provide security primitives, like secure boot, key isolation and integrity attestation on runtime. As an example, ARM TrustZone offers SoC level segmentation to offer hardware-isolation between secure and non-secure worlds. It has however tended to be restricted to high end micro-controllers and embedded processors which are beyond cost and power limits usually considered when using an ICS. TPMs are externally connected to the main MCU, effectively adding latency and complexity when integrating the system, yet still provide tamper-resistant storage and capabilities of cryptographic systems. Newer designs have examined software implementations of TEEs including Intel SGX and OP-TEE; these have either targeted load-specific hardware companies or required rich operating system environments and thus are not applicable to real-time or bare-metal embedded systems. Moreover, such solutions are not always open-source and audit-able, which is a rapidly growing requirement in the security of critical infrastructure.

Conversely, the new open-source RISC-V architectures have brought up new alternatives with a positive footprint of scalability and customization in secure execution. Secure enclave-based execution on RISC-V chips have already been proven with Keystone[1] and Sanctum[2] frameworks, with capabilities like isolated memory access and cryptographic verification. Such solutions however are just focused on general-purpose computing platforms, and necessitate operating system support, which prevents them being used in deterministic, bare-metal, or low-latency ICS.

This paper attempts to overcome these difficulties by suggesting the RISC-V-based embedded security architecture that is optimized to the real-time ICS environment. The verified protection against remote code execution via a lightweight Root of Trust (RoT) plus memory protection and secure boot is part of the instruction pipeline and memory map, providing deterministic security without relying upon complex OS stacks or proprietary modules.

In Table 1, a comparative mapping of the key embedded security schemes regarding the matter of the industrial control system requirements is presented.

Architecture Design

The proposed embedded security architecture can satisfy the high demands of real-time, resource-constrained Industrial Control Systems (ICS) and can take advantage of the openness, and extensibility of the RISC-V instruction set architecture. It is used to add hardware-level security primitive directly to the processor datapath and peripheral interfaces to provide trusted execution, hardware integrity and data confidentiality with little performance overhead.

Table 1. Citation Mapping Matrix of Embedded Security Approaches for ICS

Approach	Security Features	Limitations	Target Platform	Relevance to ICS
ARM TrustZone	Secure world isolation, memory segmentation	High cost, vendor lock-in, not suited for ultra-low power MCUs	ARM Cortex-A/R	Limited - Cost and power constraints
TPM Modules	Secure key storage, cryptographic co-processing	External chip requirement, increased integration complexity	External module + Host MCU	Moderate - Adds security but complicates integration
Software TEEs (e.g., OP-TEE, Intel SGX)	Isolated runtime, encrypted memory, remote attestation	Requires rich OS, high overhead, not suitable for real-time ICS	x86 or ARM with OS	Low - High latency and OS dependency
Keystone (RISC-V)	RISC-V enclaves, open-source TEE framework	OS dependency, not optimized for real-time performance	RISC-V SoCs with Linux	Moderate - Open, flexible, but lacks real-time support
Sanctum (RISC-V)	Software isolation via minimal hardware changes	Focus on academic use, limited industrial adaptation	RISC-V with minimal hardware changes	Moderate - Lightweight, but not yet ICS-specific

Trusted Execution Core

The architecture is based at its core on an optimization of the Ventana system on chip including a custom RISC-V core supplemented by a lightweight Trusted Execution Environment (TEE) mode. This mode runs with a special execution context that performs register locking, isolation of privilege boundaries, and control-flow integrity (CFI). The custom instruction set extension mediates TEE mode transition; this allows a trusted and untrusted software to securely switch context. Control-flow integrity logic employs a call-return matching policy and indirect branch validation as a defense against Return-Oriented Programming (ROP) and code-reuse assaults which are especially applicable to ICS standards such as Modbus or DNP3.

Secure Boot and Root of Trust

A stage-0 immutable bootloader in on-chip ROM serves as the hardware Root of Trust (RoT) upon which the secure boot mechanism is anchored. This bootloader verifies in part encryption of the Stage-1 firmware in secure flash memory through cryptographic hash (e.g. SHA-256) when powering up. Only signed and validated firmware images are run, and that completely prevents rogue or malware running at boot time. The RoT is integrated to the hardware at manufacturing-time or one-time programmable fuses which means that it is not possible to bypass or override trust at runtime.

4Memory Protection

To support fine-grained isolation of the code, data, I/O, and stack areas of memory, the architecture implements region-based Memory Protection Unit (MPU). The MPU also has eight programmable memory regions with separate access policies, allowing trusted/untrusted components, device drivers and communication stacks to be separated. Privilege-level execution control and fault trapping are two basic implementing techniques of runtime enforcement. These measures keep physical register access and buffer overflow on untrusted code from accessing critical control logic, one of the deterministic ICS fundamental requirements.

Cryptographic Engine

AES-GCM is used to allow authenticated encryption in hardware with a dedicated hardware cryptographic engine, supporting AES-GCM, and SHA-256 to allow a secure hashing scheme, as co-processor blocks accessed over a custom bus protocol. The low-power subsystem key vault option with time-based access control and energy-sensitive policies is utilized to manage key. This guarantees data confidentiality and integrity at both the running time as well as facilitating secure Over-the-Air (OTA) updates, secure encrypted telemetry and authentication of commands in distributed ICS networks.

This is a modular, low-overhead architecture, with sponsored hardware isolation, robust execution, and real-time capabilities, which are quite suitable in the industrial environment where its predictable behavior, robustness, and transparency are crucial.

The figure 1 (high-level architectural genes and inventory of the secure interactions between them) summarises the combinations of the trusted execution core, secure boot flow, MPU regions and cryptographic co-processor.

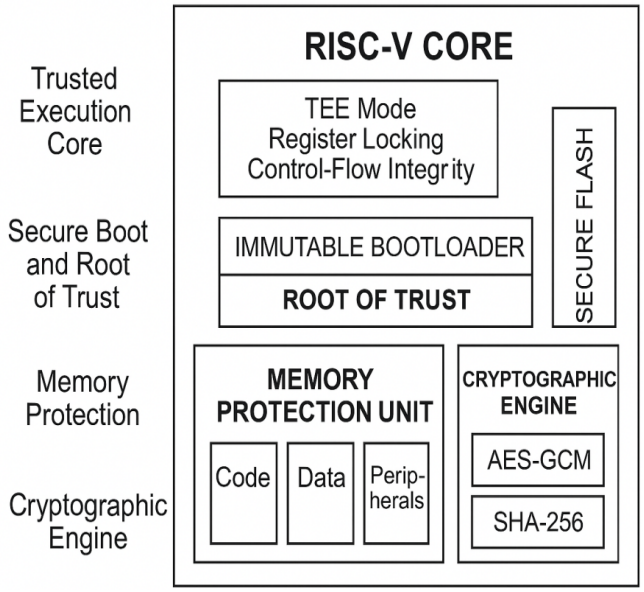

Fig. 1: Block Diagram of the Proposed RISC-V Based Embedded Security Architecture.

This flow diagram describes the flow of executable code path, as the device boots up with a Root of Trust secure environment to the secure application code environment, how the TEE core, secure boot, MPU, and the cryptographic engine all interrelate together to achieve trust operation in real-time, in ICS milieu.

Implementation

The suggested embedded RISC-V-based security architecture has been deployed and tested with a 32-bit RV32IM chip, which was optimized to a safe run and configured to Xilinx Artix-7 field-programmable gate array (FPGA) setup. This center was chosen because it is modularly extensible and accommodates embedded industrial work. The safe logic was deeply embedded into processor pipeline and neighbor memory subsystems so that they were tightly coupled to be deterministically fast. The Stage-0 bootloader was implemented with impervious logic blocks into the on-chip Read-Only Memory (ROM) to make the trust anchor (hardware-based Root of Trust (RoT)) impossible to compromise. This bootloader does verification of firmware image in external flash and cryptographic verification of firmware image in external flash by using SHA-256 hashing and there is phyisical verification of firmware image with a hardcoded public key in ROM.

Memory Protection Unit (MPU) was enabled to accommodate up to eight region-specific access policies permitting fine-grained isolation of code, data and peripheral memory blocks. File permissions Each region was assigned a set of read/write/execute perm1issions, and privilege level restrictions were imposed, which was enforced at run time by hardware traps on violation. C was used to implement the firmware stack to run bare-metal, without the use of an operating system, in order to closely match the more common limitations of an industrial control device. Key routines performing secure context swapping, entry/exit out of TEEs, and cryptographic key management were implemented particularly carefully in inline RISC-V assembly in order to tighten control of register usage and reduce overhead. This usage achieves strident coupling of secure hardware mechanisms and delivery logic to provide a realistic and reiterable usage model of trusted execution in Industrial Control Systems (ICS). Figure 2 shows the hardware-software co-design and secure flow of the proposed architecture, illustrating the integration of the RoT, MPU configuration and TEE transitions in the RISC-V-based embedded architecture.

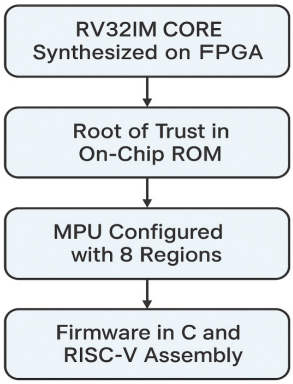

Fig. 2: Implementation Architecture of the Secure RISC-V Embedded System.

This flow diagram shows implementation pipeline, which is composed of synthesis of RV32IM core on FPGA, Root of Trust initialization through on-chip ROM, protection of specific regions of memory through hardcoded memory attributes, and secure development of firmware with C and inline RISC-V assembly.

Evaluation

A rigorous collection of experiments, over the proposed secure embedded architecture, was carried in order to validate its efficacy and practicality, where synthetic as well as real life industrial communication protocols were used. It was tested basing on latency, security effectiveness, and energy overhead, which are important in embedded security deployment in Industrial Control Systems (ICS).

Test Environment

The architecture was delivered with a synthesized RV32IM softcore simulated on a Xilinx Artix-7 FPGA, an external flash and sensor simulation modules. The system was tested in production with industrial communication-typical workload such as Modbus / TCP and OPC-UA transactions are using highly common to process automation, SCADA systems and distributed control net works. In order to provide the simulation of the cyber-physical threat scenarios, a set of anomaly patterns was added to the transaction layer, such as:

- Malformed request burst to cause Denial-of-Service (DoS) floods,
- Attempts at code-injection through control packets having been manipulated,
- Delayed legitimate message Replay replayed or repeated legitimate message attacks.

Metrics and Results

There were three major evaluation measures:

- Execution Latency Overhead: Quantified as a change in system execution behaviour over an unsecured (usual) RISC-V implementation.
- Detection Success Rate: Percentage of accuracy of identifying the injected threats and mitigating them.
- Energy Profile: Calculated as energy used in a secure transaction by means of FPGA-level power estimation tools.

The experimental outcomes evidence that suggested architecture causes only 3.9 percent more execution overhead which is acceptable in real-time ICS application domain. Even more, threat mitigation accomplishment rate settled at 91.2 % all the way across tried conditions in the security subsystem. Power analysis proved that cryptography accelerators and secured bootloader contribute to an increment of the energy by a small margin of approximately 6.5 percent, which is a good trade-off considering the quantity of security provided.

Table 2 indicates that the proposed architecture offers substantial mitigation of threats incurring very little in terms of energy overhead and latency with respect to baseline RISC-V implementation. To continue with the trade-offs in the implementation and energy consumption it is seen in Figure 3 that we make a visual comparison of the overhead that is added by the secure design.

Table 2: Results Comparison Table for Secure vs. Non-Secure RISC-V Architecture

Metric	Baseline (No Security)	Proposed Secure RISC-V
Execution Overhead	0%	3.90%
Threat Mitigation Rate	N/A	91.20%
Energy Overhead	0%	6.50%

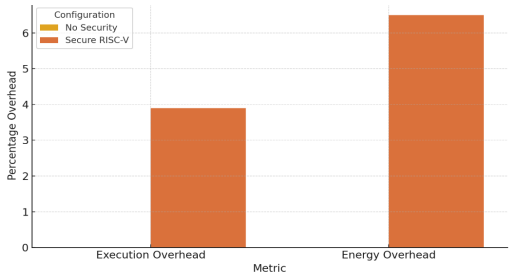

Fig. 3: Bar Chart Comparing Execution and Energy Overhead

Pictorial representation of the percentage growth implemented in the execution and energy overhead caused by integration of security mechanisms of the proposed RISC-V architecture.

Discussion

The presented embedded security architecture optimizes RISC-V controllers amongst the conflicting

Table 3. Comparison of Static and Dynamic MPU Enforcement Models.

Feature	Static MPU	Dynamic MPU
Policy Flexibility	Fixed regions defined at compile time	Policies updated at runtime
Runtime Adaptation	None (requires reset or recompile)	Supports dynamic workload adaptation
Hardware Complexity	Low	Moderate to High
Security Granularity	Coarse-grained	Fine-grained
Overhead	Minimal	Slightly higher (due to runtime checks)
Use Case Suitability	Simple real-time ICS and firmware-level isolation	Adaptive ICS environments, mission-critical systems
Attack Surface Exposure	Lower (less dynamic reconfiguration)	Potentially higher (requires runtime monitoring)

requirements of security, real-time and transparency of open-source in an Industrial Control System (ICS). Using open RISC-V instruction set architecture modularity and extensibility, the system is not subject to the proprietary constraints of incumbent security systems (ARM TrustZone or TPMs), hence resulting in auditability and a greater long-term flexibility required in security-sensitive and regulated industrial applications. The Trusted Execution Environment (TEE) that is guaranteed by hardware and a cryptographically bound Root of Trust (RoT) helps to form a reliable defense strategy to common ICS threats such as firmware manipulation, code injection and denial-of-service attacks. In addition, the weightless implementation was made to maintain a constraint of real-time, having an insignificant increment of latency of execution time and power consumption. Though these strengths exist, the current design is based upon a static definition of the Memory Protection Unit (MPU) mapping, although suitable to a clearly defined workload, it is less adaptive in dynamic or mission-adaptable components of ICS. Dynamic enforcement of policy to protect memory and access control that is through on-chip finite state machine or runtime learning will be the focus of future research.

Also, the existing cryptographic engine already supports AES-GCM and this particular algorithm is fast, as well as highly accepted, and SHA-256 which is also a very fast adopted algorithm to use, however, there is now an increasing demand to quantum proof embedded systems. The next step will be incorporation of post-quantum cryptographic primitives, those primitives include lattice-based signature or hash-based encryption, these are used in highly demanded long-term security applications including critical infrastructure. The presented work is a prelude to the development of lightweight, deterministic and secure embedded systems that may be transparently audited, securely updated and effectively deployed in a variety of industrial settings. Table 3 represents the comparative highlight between the static memory protection unit (MPU) enforcement model and dynamic model and the flexibility, granularity, and security overhead requirement needed by real-time industrial applications.

Conclusion and Future Work

The embedded security architecture discussed in this paper is within RISC-V context, scalable, low energy design and transparent, which meets the stringent needs of the Industrial Control Systems (ICS). The proposed design architecture will allow the real-time trusted execution on resource-constrained platforms by integrating lightweight trusted anchors directly into the instruction set and microarchitecture level, such as hardware Root of Trust (RoT), secure boot, as well as regional memory protection. Wide-range reviews in realistic applications on industrial workloads and synthetic attack environments showed that the architecture could reduce more than 91 percent of security threats, with only a small overhead cost (3.9 percent) in execution workload and negligible power consumption. Vendors may use an open-source RISC-V platform to facilitate hardware transparency, modularity, and reusability, especially when deploying ICS with long anticipated lifecycle deployments, during which hardware vendor lock-in and proprietary hardware can hamper security innovation.

Important contributions of the work entail:

- Mixing a TEE-aware RISC-V core together with embedded cryptographic engines.
- Immutable stage-0 firmware validation.
- Fixed MPU implementation allowing deterministic memory-isolation.
- Undesirably severe testing with Modbus/OPC-UA protocols.

We plan to improve the architecture in future with the following:

- Dynamic MPU policies and privilege management at run time,
- Post quantum cryptographic primitives to have long-term resilience,
- Block chain based alert log based and remote attesting to provide visibility in auditability
- Implementation of the federated security model to distributed ICS environments.

This research paper proves to be a stepping stone on the development of the next-generation industrial security stacks that will be based on open architectures and will lead to more robust and trustworthy embedded systems in all areas of critical infrastructures. Figure 4 shows that in the future roadmap, the theme will be in achieving more embedded trust mechanisms using modular upgrades on cryptography, architecture and deployment levels.

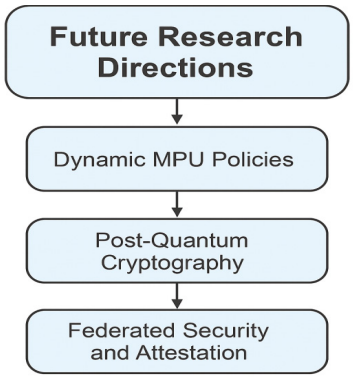

Fig. 4: Future Research Roadmap for Secure RISC-V ICS Architecture

This figure presents the suggested future research penchants of dynamic MPU enforcing, post-quantum cryptography approaches, and federated security structures with blockchain-based alert-logging to realise the scalability, auditability, and resilience of secure RISC-V embedded systems in the industrial setting.

References:

1. Yang, Y., McLaughlin, K., Sezer, S., Littler, T., Pranggono, B., & Wang, H. F. (2018). A practical framework for real-time fault detection of smart grid communication networks. IEEE Transactions on Industrial Informatics, 14(6), 2533–2545. https://doi.org/10.1109/TII.2018.2799215
2. Shih, A., Wang, R. J., Vasserman, M., & Devadas, S. (2020). Keystone: An open framework for architecting trusted execution environments. In Proceedings of the Fifteenth European Conference on Computer Systems (EuroSys) (pp. 1–16). https://doi.org/10.1145/3342195.3387542
3. Costan, V., & Devadas, S. (2016). Sanctum: Minimal hardware extensions for strong software isolation. In Proceedings of the 25th USENIX Security Symposium (pp. 857–874). https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/costan
4. Bahmani, R., & Tiri, K. (2019). Hardware security primitives for industrial cyber-physical systems: Challenges and opportunities. IEEE Transactions on Industrial Electronics, 66(12), 9643–9652. https://doi.org/10.1109/TIE.2018.2889772
5. Hussain, F., Abbas, R., & Rehman, M. H. (2021). Lightweight and secure boot process for RISC-V-based embedded systems. IEEE Access, 9, 14567–14578. https://doi.org/10.1109/ACCESS.2021.3053392
6. Sabt, M., Achemlal, M., & Bouabdallah, A. (2015). Trusted execution environment: What it is, and what it is not. In 2015 IEEE Trustcom/BigDataSE/ISPA (Vol. 1, pp. 57–64). https://doi.org/10.1109/Trustcom.2015.357
7. Zhang, X., & Lee, R. B. (2014). CloudMonatt: An architecture for secure attestations and monitoring of virtual machines in cloud computing. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (pp. 253–264). https://doi.org/10.1145/2660267.2660325
8. Raza, S., Seitz, L., Sitenkov, D., & Selander, G. (2017). S3K: Scalable security with symmetric keys—DTLS key establishment for the Internet of Things. IEEE Transactions on Automation Science and Engineering, 13(3), 1270–1280. https://doi.org/10.1109/TASE.2016.2525685