RISC-V Security Model (non-normative)

RISC-V Security Model Task Group

Version 0.1, 10/2023: This document is in development. Assume everything can change. See http://riscv.org/spec-state for details.

Table of Contents

Preamble

> *This document is in the Development state*
>
> Assume everything can change. This draft specification will change before being accepted as informative, so implementations made to this draft specification will likely not follow the future informative specification.

Copyright and license information

Contributors

This RISC-V specification has been contributed to directly or indirectly by (in alphabetical order): Ali Zhang, Andy Dellow, Carl Shaw, Colin O’Flynn, Dean Liberty, Dong Du, Deepak Gupta, Colin O’Flynn, Guerney Hunt, Luis Fiolhais, Manuel Offenberg, Markku Juhani Saarinen, Munir Geden, Mark Hill, Nicholas Wood, Paul Elliott, Ravi Sahita, Samuel Ortiz, Steve Wallach, Suresh Sugumar, Terry Wang, Victor Lu, Ved Shanbhogue, Yann Loisel

Chapter 1. Introduction

This specifications provides guidelines for how Risc-V systems can use Risc-V security building blocks to build secure systems for different use cases.

It does this through a few example uses cases based on commonly used profiles. This is not intended to be exhaustive, it is expected that the principles described in the chosen example are general enough to be applicable to other use cases as well. But the examples may be extended over time as required.

1.1. Requirements and tracking

Where this specification makes formal recommendations, those are captured as trackable requirements using the following format:

ID#	Requirement
CAT_NNN	The `CAT` is a category prefix that logically groups the requirements and is followed by 3 digits - `NNN` - assigning a numeric ID to the requirement. The requirements use the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" that are to be interpreted as described in RFC 2119 when, and only when, they appear in all capitals, as shown here. When these words are not capitalized, they have their normal English meanings.

A requirement or a group of requirements may be followed by non-normative text providing context or justification for the requirement. The non-normative text may also be used to reference sources that are the origin of the requirement.

Trackable requirements are intended for ease of reference across dependant specifications.

1.2. Relationship to external profiles

For the purpose of this specification, external profiles apply to existing ecosystems or segments, but do not generally mandate implementations or architectures. This specification does not aim to establish new profiles. Its main purpose is to provide guidelines for how Risc-V security building blocks can be used to build Risc-V products which can comply with existing profiles.

Some profiles cover some or all of:

- Security reference architectures and taxonomy
- Hardware and software security requirements
- Interfaces and programming models
- Protection profiles and certification programmes
- Reference firmware/software

Other profiles are focussed on processes and methodology.

Examples of external profiles include:

Profile	Description
Global Platforms (GP)	Trusted execution environments(TEE) and trusted firmware for mobile, connected clients, and IoT Secure element (SE) for tamper resistant storage of and operations on cryptographic secrets globalplatform.org/
Trusted computing group (TCG)	Trusted platform module (TPM) and Device identifier composition engine (DICE) for trusted platforms trustedcomputinggroup.org/
Confidential computing consortium	Common principles and protocols for protecting data in use (confidential compute) confidentialcomputing.io/
NIST	Widely used US standards for security processes, protocols and algorithms. Examples for the purposes of this specification: NISTIR 8259 - IoT device cybersecurity capability SP800-207 - Zero Trust Architecture www.nist.gov/

This is not an exhaustive list. It is provided by way of example.

Chapter 2. Risc-V security model overview

2.1. Refence model and taxonomy

2.2. Adversarial model

2.3. Security objectives

Chapter 3. Risc-V security building blocks

3.1. Isolation

3.2. Runtime integrity

3.3. Control flow integrity

3.4. Memory safety

3.5. Cryptography

3.6. Roadmap

3.6.1. Capability based architecture

Cheri/Capstone

3.6.2. Memory safety

3.6.3. Lightweight isolation

3.6.4. System level isolation

WorldGuard

3.6.5. Cryptography enhancements

Chapter 4. Use case examples

4.1. Basic non-virtualized system

4.1.1. Overview

4.1.2. Isolation model

4.1.3. Device access control

4.1.4. Sealing

4.1.5. Attestation

4.2. Basic virtualized system

4.2.1. Overview

4.2.2. Isolation model

4.2.3. Device access control

4.2.4. Sealing

4.2.5. Attestation

4.3. Global Platforms TEE

4.3.1. Overview

4.3.2. Isolation model

4.3.3. Device access control

4.3.4. Sealing

4.3.5. Attestation

4.4. Confidential compute system (Cove)

4.4.1. Overview

4.4.2. Isolation model

4.4.3. Device access control

4.4.4. Trusted device assignment

4.4.5. Sealing

4.4.6. Attestation

4.5. Additional examples

(Variations on the above)

Android pKVM

Chapter 5. Cryptography

<use models enabled by crypto ISA extensions>

<crypto guidelines>

Appendix A: References

1. https://www.intel.com/content/www/us/en/newsroom/opinion/zero-trust-approach-architecting-silicon.html
2. https://www.forrester.com/blogs/tag/zero-trust/
3. https://docs.microsoft.com/en-us/security/zero-trust/
4. https://github.com/riscv/riscv-crypto/releases
5. https://github.com/riscv/riscv-platform-specs/blob/main/riscv-platform-spec.adoc
6. https://www.commoncriteriaportal.org/files/ppfiles/pp0084b_pdf.pdf
7. https://docs.opentitan.org/doc/security/specs/device_life_cycle/
8. https://nvlpubs.nist.gov/nistpubs/ir/2021/NIST.IR.8320-draft.pdf
9. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-193.pdf
10. https://www.rambus.com/security/root-of-trust/rt-630/
11. https://docs.opentitan.org/doc/security/specs/
12. https://trustedcomputinggroup.org/work-groups/dice-architectures/
13. https://ieeexplore.ieee.org/iel7/8168766/8203442/08203496.pdf
14. https://dl.acm.org/doi/10.1145/168619.168635
15. https://dl.acm.org/doi/abs/10.1145/3342195.3387532
16. https://github.com/riscv/riscv-debug-spec/blob/master/riscv-debug-stable.pdf
17. https://csrc.nist.gov/csrc/media/events/non-invasive-attack-testing-workshop/documents/08_goodwill.pdf
18. https://www.iso.org/standard/60612.html
19. https://ieeexplore.ieee.org/document/6176671
20. https://tches.iacr.org/index.php/TCHES/article/view/8988
21. https://ieeexplore.ieee.org/abstract/document/1401864
22. https://www.electronicspecifier.com/products/design-automation/increasingly-connected-world-needs-greater-security
23. https://www.samsungknox.com/es-419/blog/knox-e-fota-and-sequential-updates
24. https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/supply-chain-malware
25. https://dl.acm.org/doi/10.1145/3466752.3480068
26. https://arxiv.org/abs/2111.01421
27. https://www.nap.edu/catalog/24676/foundational-cybersecurity-research-improving-science-engineering-and-institutions
28. https://trustedcomputinggroup.org/work-groups/dice-architectures/

Bibliography

- [1] The RISC-V Instruction Set Manual Volume II: Privileged Architecture Document Version 20211203 (link)