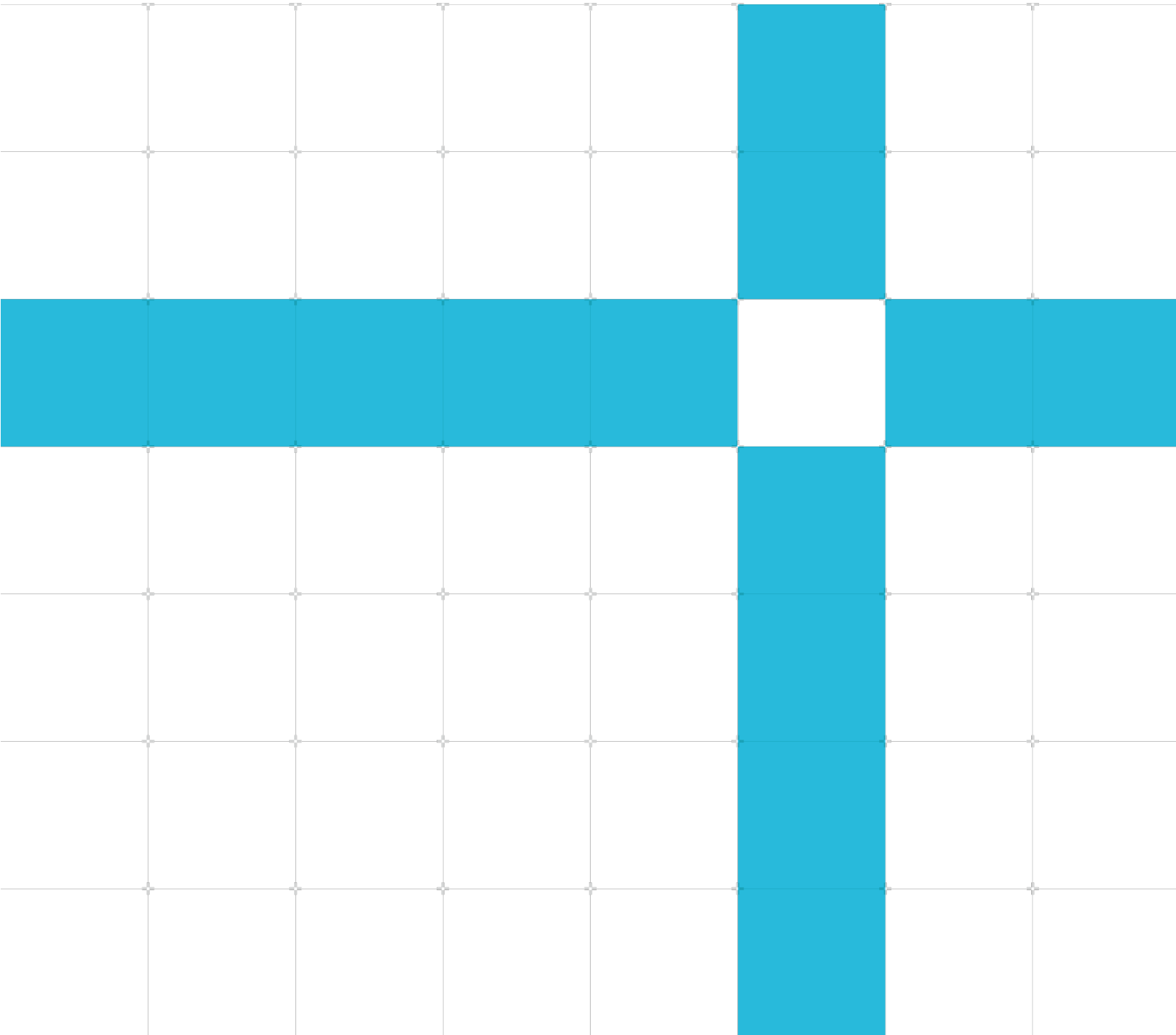
Arm® DRTM Architecture Compliance

Revision: r1p0

Test Scenario

Non-Confidential

Issue 0001-00
ARM040-1254092399-17753



Arm DRTM Test Scenario Document

Release information

Document history

Issue	Date	Confidentiality	Change
01	29 Jun 2024	Non-Confidential	Changes for ALP 1.0

Non-Confidential Proprietary Notice

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is for a final product, that is for a developed product.

Progressive terminology commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used terms that can be offensive. Arm strives to lead the industry and create change.

This document includes terms that can be offensive. We will replace these terms in a future issue of this document. If you find offensive terms in this document, please email **terms@arm.com**.

Web Address

www.arm.com

Contents

1 Introduction 5
1.1 Product revision status 5
1.2 Intended audience 5
1.3 Conventions 5
1.3.1 Glossary 5
1.3.2 Typographical Conventions 6
1.4 Useful resources 6
1.5 Feedback 7
1.5.1 Feedback on this product 7
1.5.2 Feedback on content 7

2 Arm DRTM Architecture 8
2.1 DRTM ACS 8
2.2 Interface 9
2.3 Dynamic Launch 11

Appendix A Revisions 14

1 Introduction

1.1 Product revision status

The *rmpn* identifier indicates the revision status of the product described in this book, for example, r*1*p*2*, where:

rm Identifies the major revision of the product, for example, r*1*.

pn Identifies the minor revision or modification status of the product, for example, p*2*.

1.2 Intended audience

This document is for engineers who are verifying an implementation of Arm® Dynamic Root of Trust for Measurement for processors based on the Arm A-profile architecture.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

1.3.1 Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: **https://developer.arm.com/glossary**.

1.3.2 Typographical Conventions

Convention	Use
italic	Introduces citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace` **`bold`**	Denotes language keywords when used outside example code.
`monospace underline`	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Used in body text for a few terms that have specific technical meanings, that are defined in the Arm® Glossary. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

1.4 Useful resources

This document contains information that is specific to this product. See the following resources for other relevant information.

- Arm Non-Confidential documents are available on **developer.arm.com/documentation**. Each document link in the tables below provides direct access to the online version of the document.
- Arm Confidential documents are available to licensees only through the product package.

Arm products	Document ID	Confidentiality
DRTM Architecture For ARM	DEN0113	Non-Confidential
Arm® Architecture Reference Manual Armv8, for Armv8-A Architecture	ARM DDI 0487H.a	Non-Confidential

1.5 Feedback

Arm welcomes feedback on this product and its documentation.

1.5.1 Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

1.5.2 Feedback on content

If you have comments on content, send an email to **support-systemready-acs@arm.com** and give:

- The title Arm Base System Architecture Scenario.
- The number ARM040-1254092399-17753
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.
- Arm also welcomes general suggestions for additions and improvements.

2 Arm DRTM Architecture

The objective of DRTM is to begin a new chain of trust and instantiate a smaller TCB that excludes untrusted and arbitrarily extensible components. DRTM does this by measuring and launching a protected payload.

Establishing an attestable TCB becomes difficult when the number of components in the boot chain grows or when firmware is dynamically extensible, for example by loading drivers from add-in peripherals. The larger and more complex the TCB, the greater the attack surface and the risk of untrusted code executing which can compromise security. For example, UEFI is an extensible boot loader, where multiple EFI programs might run during the Boot Services phase. The EFI programs can include drivers, device option ROMs, and a bootloader. If compromised, these programs might be able to further compromise the target OS by tampering with the OS's code or data.

Dynamic Root of Trust for Measurement (DRTM) begins a new chain of trust by measuring and executing a protected payload. The newly started chain of trust results in a smaller TCB. DRTM is implemented by a trusted agent that ensures the following:

- All cores are placed in a known state
- The target payload is protected against modification
- A single core measures and begins running the payload
- Execution is confined to the payload
- The payload is provided with data that can be used to validate key properties of the system

2.1 DRTM ACS

The tests are divided into interface and Dynamic launch tests. Interface tests checks compliance for DRTM Functions and features supported. Dynamic Launch checks compliance for Successful dynamic launch with an example dlme image.

DRTM compliance also require system to be complaint with BSA specification. Please refer https://github.com/ARM-software/bsa-acs/blob/main/docs/Arm_Base_System_Architecture_Scenario_ES.pdf for BSA tests.

The tests are classified as:

- Interface
- Dynamic Launch

2.2 Interface

Test number	Rule ID	Scenario	Algorithm
1	NA	Major Version of the DRTM implementation should be 1.	Check for the version of DRTM implementation using DRTM_VERSION function ID
2	NA	1. Unimplemented functions return NOT_SUPPORTED error code 2. If the value of Reserved bits is not zero then get NOT_SUPPORTED error code	1. Pass Invalid Function ID and check for NOT_SUPPORTED error code 2. Pass Invalid reserved bits and check for NOT_SUPPORTED error code
3	R31000	DRTM instance must implement mandatory functions	Check all the mandatory functions return status as 0
4	NA	TCB hash features Reserved bits must be zero	Send DRTM_FEATURES function with 0x5 as Feature ID and check reserved bits
5	R58000	DMA protection features Reserved bits must be zero	Send DRTM_FEATURES function with 0x3 as Feature ID and check reserved bits
6	NA	TPM features Reserved bits must be zero	Send DRTM_FEATURES function with 0x1 as Feature ID and check reserved bits
7	NA	Minimum memory requirement Reserved bits must be zero	Send DRTM_FEATURES function with 0x2 as Feature ID and check reserved bits
8	R512000	The version of PSCI should be 1.0 or later	Get and check the version of PSCI is greater than 1
9	R512020	The version of SMCCC should be 1.0 or later	Get and check the version of SMCCC is greater than 1
10	R54010	GIC that implements LPIs should support clearing GICR_CTLR.EnableLPIs	Get ITS Address for current ITS. Check GITS_CTLR.Enabled = 0 and GITS_CTLR. Quiescent = 1
11	R54020	If GIC does not support clearing GICR_CTLR.EnableLPIs after it is set, modification of GICR_PENDBASER when GICR_CTLR.EnableLPIs == 1 must not permit	Get RDBase Address for current PE, Check GICR_CTLR.EnableLPIs = 0 and GICR_CTLR.RWP = 0
12	R31010	If the DRTM_SET_TCB_HASH function is implemented, the DRTM_LOCK_TCB_HASHES function must be implemented.	1. Check if DRTM_SET_TCB_HASH function is implemented. 2. If SUCCESS is returned then check that when DRTM_FEATURES function with DRTM_LOCK_TCB_HASHES as Function ID should return SUCCESS

Test number	Rule ID	Scenario	Algorithm
13	R315040	If this maximum number of entries in the hash table is exceeded, the implementation must return the return value OUT_OF_RESOURCE	1. Check if DRTM_SET_TCB_HASH function is implemented. 2. Fill the hash table with hashes greater than the maximum available hash entries from DRTM_FEATURES function. Send the DTRM_SET_TCB_HASH function and check if OUT_OF-RESOURCE is returned
14	R315040	1. Error in the hash table should result in INVALID_PARAMETERS status 2. Invoke DRTM_LOCK_TCB_HASHES and then invocation of DRTM_SET_TCB_HASH function should result in DENIED status	1. Check if DRTM_SET_TCB_HASH function is implemented. 2. Fill the hash table with incorrect revision and set hashes with DRTM_SET_TCB_HASH function. Check if INVALID_PARAMETERS is returned. 3. Lock the hashes by using DRTM_LOCK_TCB_HASHES function. Fill the hash table. Send the function and check if DENIED status is returned

2.3 Dynamic Launch

Test number	Rule ID	Scenario	Algorithm
101	R312000 - R312050 R314000 - R314040	1. The DRTM parameters must start at a 4KB aligned address 2. The DLME region, image and data must start at a 4KB aligned address 3. The DLME image must come before the DLME data and must not overlap with it	Pre-Condition: Send DRTM_FEATURES function with 0x2 as Feature ID to get Minimum size of DLME data and set DRTM parameters. 1. Send DRTM_DYNAMIC_LAUNCH function with unaligned DRTM parameters address and check return status as INVALID_PARAMETERS 2. Send DRTM_DYNAMIC_LAUNCH function with unaligned DLME Region, Image and Data start address in DRTM parameters and check return status as INVALID_PARAMETERS 3. Send DRTM_DYNAMIC_LAUNCH function by Swapping DLME image address and DLME data address, so that DLME Data is before DLME Image in DRTM parameters and check return status as INVALID_PARAMETERS.
102	R42000 R42030 R42060 R43000 R43010 R45380 R45390 R45450	Dynamic Launch Event should success	Pre-Condition: Send DRTM_FEATURES function with 0x2 as Feature ID to get Minimum size of DLME data and set DRTM parameters. Send DRTM_DYNAMIC_LAUNCH function with DRTM parameters. Dynamic Launch should perform. Call DRTM Unprotect Memory After Dynamic Launch, check for x0 and x1 values saved during DLME Image with expected values. Check DRTM_GET_ERROR Success Case in case of no errors in previous Dynamic Launch Call unprotect memory again, it should return DENIED as no memory protected.

Test number	Rule ID	Scenario	Algorithm
103	NA	Successive Dynamic Launch and Dynamic Launch Denied Error Case	Pre-Condition: Send DRTM_FEATURES function with 0x2 as Feature ID to get Minimum size of DLME data and set DRTM parameters. Send DRTM_DYNAMIC_LAUNCH function with DRTM parameters. Dynamic Launch should perform. Do Not Call Unprotect Memory After Dynamic Launch, check for x0 and x1 values saved during DLME Image with expected values. Check DRTM_GET_ERROR Success Case in case of no errors in previous Dynamic Launch Invoke Dynamic Launch Again, it should return fail as unprotect memory not done. Call unprotect memory, it should return Success. Invoke Second Dynamic Launch. After Dynamic Launch, check for x0 and x1 values saved during DLME Image with expected values.
104	NA	Check DRTM Close Locality	Pre-Condition: Send DRTM_FEATURES function with 0x2 as Feature ID to get Minimum size of DLME data and set DRTM parameters. Send DRTM_DYNAMIC_LAUNCH function with DRTM parameters. Dynamic Launch should perform. Check Close Locality for Locality 1, Should Result INVALID Parameters. Check Close Locality for Locality 2, Should Result Success. Check Close Locality for Locality 2 Again, Should Result Already closed, as it is already closed.

Test number	Rule ID	Scenario	Algorithm
105	R312060 R313000 R313010 R313020 R314050 - R314150 R42020 R45160 R45250 R45260 R45270 R45420 R45440	Check DLME Data Rules	Pre-Condition: Send DRTM_FEATURES function with 0x2 as Feature ID to get Minimum size of DLME data and set DRTM parameters. Send DRTM_DYNAMIC_LAUNCH function with DRTM parameters. Dynamic Launch should perform. Call DRTM Unprotect Memory. Read DLME Data from DLME Data Region. Check that DLME Data is properly populated.
106	R316000 R316020 R316030	Check DRTM Event Log	Pre-Condition: Send DRTM_FEATURES function with 0x2 as Feature ID to get Minimum size of DLME data and set DRTM parameters. Send DRTM_DYNAMIC_LAUNCH function with DRTM parameters. Dynamic Launch should perform. Call DRTM Unprotect Memory. Read Event Log from DLME Data Region. Check that Event Log is properly populated.
107	R42010	When DLME image is launched on boot PE then all other PEs should be off	Pre-Condition: Send DRTM_FEATURES function with 0x2 as Feature ID to get Minimum size of DLME data and set DRTM parameters. Switch on the secondary PE. Send DRTM_DYNAMIC_LAUNCH function with DRTM parameters. Check return status as SECONDARY_PE_NOT_OFF

Appendix A Revisions

This appendix describes the technical changes between released issues of this book.

Table A-1 Issue 02

Change	Location
First release	-