Arm© CryptoCell™-703

Product revision: r0p0-00rel0

OSS RT Release Note

Non-Confidential

PJDOC-
1779577084-32285

Arm© CryptoCell™-703

OSS RT Release Note

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 Final, that is for a developed product.

Web address

www.arm.com

Contents

1 Conventions **5**
1.1 Glossary 5
1.2 Typographical conventions 5

2 Release overview **7**
2.1 Product description 7
2.2 Release status 7
2.3 Standards compliance 7

3 Release contents **8**
3.1 Deliverables 8
3.1.1 Associated products 8
3.2 Differences from previous release 8
3.3 Known limitations 8

4 Get started **9**
4.1 Licensing information 9
4.2 Download the product 9
4.2.1 Unpack the product 9
4.2.2 Compile the product 10
4.2.3 Directory structure 10
4.3 Adapt the product for your system 12
4.4 Examples 12

5 Support **13**
5.1 Tools 13
5.2 OS 13

1 Conventions

The following subsections describe conventions used in Arm documents.

1.1 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**.

1.2 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` <u>`underline`</u>	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.
Caution	This represents a recommendation which, if not followed, might lead to system failure or damage.
Warning	This represents a requirement for the system that, if not followed, might result in system failure or damage.
Danger	This represents a requirement for the system that, if not followed, will result in system failure or damage.

Note	This represents an important piece of information that needs your attention.
Tip	This represents a useful tip that might make it easier, better, or faster to perform a task.
Remember	This is a reminder of something important that relates to the information you are reading.

2 Release overview

2.1 Product description

The Arm© CryptoCell™-703 is an embedded security platform for high-performance systems. It is aimed primarily at client devices, such as smartphones, tablets, smart TVs, and set-top boxes. It offers a rich set of Chinese cryptographic services, targeting multiple threats.

2.2 Release status

This is the REL release of r0p0 Arm© CryptoCell™-703 software.

These deliverables are being released under the terms of the agreement between Arm and each licensee (the "Agreement"). All planned verification and validation is complete.

The release is suitable for volume production under the terms of the Agreement.

2.3 Standards compliance

CryptoCell-703 complies with the following specifications:

Table 2-1 CryptoCell-703 compliance

Document ID	Document name	Compliance
SM2	*Public Key Cryptographic Algorithm Based on Elliptic Curves* (December 2010)	Full
GM/T 0009-2012	*SM2 Cryptography Algorithm Application Specification*	Full
GB/T 0003.1-2012	*Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves - Part 1: General*	Full
GB/T 0003.2-2012	*Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves - Part 2: Digital Signature Algorithm*	Full
GB/T 0003.3-2012	*Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves - Part 3: Key Exchange Protocol*	Full
GB/T 0003.4-2012	*Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves - Part 4: Public Key Encryption Algorithm*	Full
GB/T 0003.5-2012	*Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves - Part 5: Parameter Definition*	Full
GM/T 0010-2012	*SM2 Cryptography Message Syntax Specification*	Full
GM/T 0004-2012	*SM3 Cryptographic Hash Algorithm* (December 2012)	Full
GM/T 0002-2012	*SM4 Block Cipher Algorithm*	Full

3 Release contents

The following subsections describe:

- The component parts that are delivered as part of this release.
- Any changes since the previous release.
- Any known issues and limitations exist at the time of this release.

3.1 Deliverables

Arm® CryptoCell-703 OSS includes the following deliverables:

- CryptoCell-703 runtime software.
- CryptoCell-703 runtime software integration tests.
- Runtime API documentation: Arm® CryptoCell-703 Runtime Software Developers Manual.

Note

Documentation may change between product releases. For the latest documentation bundle, please check the delivery platform.

3.1.1 Associated products

The following parts are available to licensees only:

- Arm® CryptoCell-703 runtime tools
- Arm® CryptoCell-703 Boot Services
- Arm® CryptoCell-703 Hardware

3.2 Differences from previous release

This is the first release of CryptoCell-703 runtime software OSS.

3.3 Known limitations

There are no known limitations at the time of release.

4 Get started

This section describes how to get you started with accessing, setting up, and using Arm© CryptoCell™-703.

4.1 Licensing information

The Arm® CryptoCell-703 runtime library and integration tests are published under two optional licenses, located at the root of the project tree:

- BSD-3 clause - Full license is disclosed in `BSD-3-Clause.txt`.
- Arm non-OSI - Full license is disclosed in `Arm-proprietary-license.txt`.

4.2 Download the product

Arm delivers the files through github.

You can download the product package in one of the following ways:

- Download a `.zip` file directly from **https://github.com/ARM-software/cryptocell-703-runtime**
- Use one of the following git clone commands:

Note

The target directory is only mentioned to align with the compilation commands listed afterwards.

  - `git clone https://github.com/ARM-software/cryptocell-703-runtime.git cryptocell-rt`
  - `git clone git@github.com:ARM-software/cryptocell-703-runtime.git cryptocell-rt`

You can download the product package as a single zip file: `cryptocell-703-runtime-master.zip`.

4.2.1 Unpack the product

If you downloaded a .zip file directly from github, perform the following steps to unpack the product package:

1. Relocate the package file:
   Copy the `.zip` files to the directory where these files are to be installed.
2. Unzip the package.

4.2.2 Compile the product

The following steps describe how to compile this product.

1. Set the environment variables.

   A typical setup is as follows:

```
export ARCH=arm64
export CROSS_COMPILE=<compiler-prefix>
export COMPILER_TYPE=gcc
export PATH=$PATH:/path/to/compiler/executable/dir/bin
export TEE_OS=linux64
export KERNEL_DIR=/path/to/kernel/
export TEST_BOARD=<your-board_name> (either juno or zynq)
export TEST_AL_CONFIG_NUM=< Test configuration index >
```

   Where the test configuration index depends on your test board, use the appropriate configuration number, as specified in Table 4-1:

Table 4-1 Test board

Test board (TEST_BOARD)	Configuration number (TEST_AL_CONFIG_NUM)
juno	10
zynq	3

2. It is assumed that the current location includes the extracted directory Cryptocell-703-TEE-Lib-master:

```
% cd Cryptocell-703-TEE-Lib-master
% make -C host/src ARCH=arm64 COMPILER_TYPE=gcc TEE_OS=linux64
```

   The Non-Confidential TEE library is located in `host/lib`.

3. Execute the following commands to build the corresponding Non-Confidential TEE runtime SW Integration tests:

```
% make -C host/src/tests/test_engine ARCH=arm64 COMPILER_TYPE=gcc
TEE_OS=linux64 TEST_BOARD=juno TEST_AL_CONFIG_NUM=10
% make -C host/src/tests/integration* ARCH=arm64 COMPILER_TYPE=gcc
TEE_OS=linux64 TEST_BOARD=juno TEST_AL_CONFIG_NUM=10
```

   The integration test executables can be found in `host/bin`, which need to be tested in the hardware under test.

4.2.3 Directory structure

Figure 4-1 shows the principal directory structure of this release created after unpacking the bundle:

Figure 4-1 CryptoCell-703 Non-Confidential TEE Runtime SW and Integration Tests directory structure

```
Cryptocell-703-TEE-Lib-master
```

```
|-- codesafe
|   `-- src
|       `-- crypto_api
|           |-- cc7x_sym
|           |   |-- adaptor
|           |   |-- api
|           |   `-- driver
|           |-- chinese_cert
|           |-- common
|           |-- ec_wrst
|           |   `-- ecc_domains
|           |-- pki
|           |   |-- common
|           |   `-- ec_wrst
|           |-- rnd_dma
|           |   `-- local
|           `-- sm2
|               |-- auxiliary
|               |-- dsa
|               |-- internal
|               `-- ke
|-- docs
|-- doxygen
|   `-- additional_doc_files_cc703
|-- doxygen-it
|   `-- additional_doc_files_cc703_it
|-- host
|   `-- src
|       |-- cc7x_teelib
|       |   `-- slim
|       |-- hal
|       |   `-- cc7x_tee
|       |-- pal
|       |   |-- cc_linux
|       |   |-- linux64
|       |   |   `-- driver
|       |   `-- no_os
|       `-- tests
|           |-- TestAL
|           |   |-- TestAL-Lite
|           |   |   `-- pal
|           |   |       `-- linux
|           |   |-- configs
|           |   |-- hal
|           |   |   |-- Juno
|           |   |   |-- Zynq
|           |   |   `-- include
|           |   |-- pal
|           |   |   |-- include
|           |   |   |-- linux
|           |   |   `-- no_os
|           |   |-- scripts
|           |   `-- tests
|           |       |-- includes
|           |       `-- src
|           |-- integration_cc7x3
|           |   |-- cc71x_tee
|           |   `-- cc71x_tee_ree
|           |-- proj
|           |   `-- cc7x
|           |-- test_engine
|           `-- tests_helper
|               `-- menu_engine
`-- shared
```

```
    |-- hw
    |   `-- tee_include
    `-- include
        |-- cc_util
        |-- crypto_api
        |   `-- cc7x_tee
        |-- pal
        |   |-- cc_linux
        |   |-- linux64
        |   `-- no_os
        `-- proj
            `-- cc7x_tee
        `-- ccree
```

4.3 Adapt the product for your system

For more information, see the *Arm© CryptoCell™-703 Software Integrators Manual*.

Note

The *Arm© CryptoCell™-703 Software* Integrators *Manual* is a confidential book that is only available to licensees.

4.4 Examples

The TEE Runtime SW Integration Test Package is reference code that you can you use to test the integration of your own specific platform.

5 Support

If you have any issues with the installation, content or use of this release, please create a ticket on **https://support.developer.arm.com**. Arm will respond as soon as possible.

5.1 Tools

This release has been developed with the following tools:

Table 5-1: Tools used in developing this release

Tool type	Tool name	Version
OS	Ubuntu	16.04.2 LTS: Linux 4.13.0-32-generic x86-64
Board	Arm Juno	R2
Compiler	GCC	7.3.0
File system	Buildroot	2016.05rc2

5.2 OS

This release has been developed with the following operating system:

Table 5-2: Operating system used in developing this release

Operating System	Version
Linux Kernel Vanilla	4.19.46