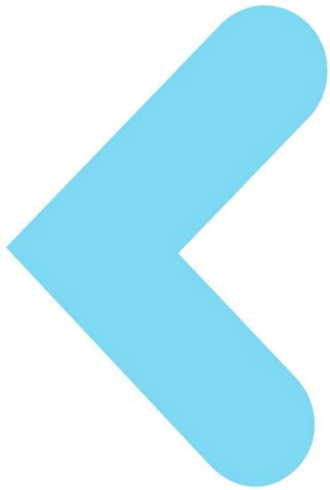
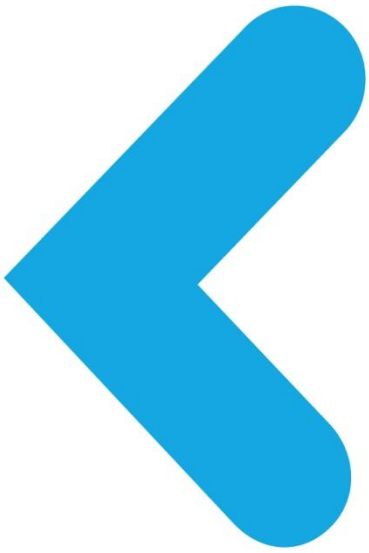
TRUSTONIC

Release Notes



Trustonic TEE v301C-v002 for QC-8909 Release Notes



PREFACE

TABLE OF CONTENTS

1 INTRODUCTION

This document is the Release Notes for the Kinibi product on Qualcomm 8909 platform.

The version of the product is 301C-V001

This version is a Release Candidate.

2 VALIDATION CONDITIONS

The total amount of memory allocated to the Trustonic TEE on any Qualcomm platform is customizable by customers at the integration step. This release has been validated granting **1 MB** of secure memory to the Trustonic TEE.

However, **3.5MB** is the minimal configuration for Trustonic TEE with applications for Trustonic partners. More memory might be needed depending on OEM requirements.

3 What's new in Trustonic TEE

3.1 TRUSTONIC TEE V301C-V002

3.1.1 Fixed Issues

- TQC-48: Potential memory leak issues on NWd Driver.
- TQC-53: Core switch support, FIQ handler is called in monitor mode and not on FIQ mode.
- TINC34: Secure Fix.

3.1.2 Upgrade

This version of Trustonic TEE is fully compatible with the Android M release.

Update OTA package from 1.28 to 1.32 to fix Android M related issues.

3.2 TRUSTONIC TEE V301C-V001

This version of Trustonic TEE is fully compatible with the Android M release.

It is an update of the previous version 301B with fixes in the Normal World Components to allow functionality on recent version of Android M.

A new NWd proxy has been added to be used to circumvent the SE Android restrictions in the Android M release.

Secure world is exactly the same as 301B.

3.3 TRUSTONIC TEE V301B-32MB-QC-8909-ANDROIDL-V003

3.3.1 Fixed Issues

FIQ callback handler

- TQC-33 FIQ callback handler causing a crash, TZBSP stack is destroyed during save and restore.
- TQC-34 FIQ callback handler causing a hang, LR is overridden during restore.

3.3.2 Upgrade

Change in Nwd to be compatible with Android M Preview.

3.4 TRUSTONIC TEE V301B-32MB-QC-8909-ANDROIDL-V002

3.4.1 Fixed Issues

Trusted User Interface

- TBUG-576 MCI buffer wrongly un-mapped at boot time due to MSH and SIQH threads concurrent mapping.

3.5 TRUSTONIC TEE V301B-32MB-QC-8909-ANDROIDL-V001

3.5.1 New Trustonic TEE Core Features

Trustonic TEE V301 introduces some new functionality and APIs.

The new Trustonic TEE API level for Trustonic TEE V301 is 4.

Therefore to use these new features, TBASE_API_LEVEL shall be set to a value equal or greater to '4'.

The following new features and improvements are listed hereafter.

3.5.1.1 Cryptographic Functionality Update

Trustonic TEE now supports:

- RSA key size support increase from 2048 to 4096 bits
- DSA algorithm with support of DSA key sizes from 512 to 3072 bits
- ECDSA with NIST P-192, P-224, P-256, P-384 and P-521 curves

Updated cryptographic parameters in the Trustonic TEE API specifications have been added.

- New key pair type TLAPI_DSA added to tlApiKeyPairType_t
- New key pair type TLAPI_ECDSA added to tlApiKeyPairType_t
- New DSA key data structure tlApiDsaKey_t added.
- tlApiDsaKey_t contains DSA parameters p, q, g (generator) as well as public key y and private key x.
- New ECDSA key data structure tlApiEcdsaKey_t added.
- tlApiEcdsaKey_t contains curve type (can be NIST P-192, P-224, P-256, P-384 and P-521), public key data x and y as well as private key
- tlApiKey_t structure has been updated to have pointers to tlApiDsaKey_t and tlApiEcdsaKey_t structures.
- New signature algorithms TLAPI_SIG_DSA_RAW and TLAPI_SIG_ECDSA_RAW added to tlApiSigAlg_t. They provide both DSA and ECDSA signature operations based on raw data (i.e. the crypto driver does NOT calculate digest but instead uses plain raw data)

While using ECDSA, the key size has to be as below:

- P-192: 24 bytes
- P-224: 28 bytes
- P-256: 32 bytes
- P-384: 48 bytes
- P-521: 66 bytes (if using 65 bytes key, it needs to have a leading '0' byte as padding)

Testing of the functionality is performed via the Android 4.4 KitKat keymaster as described below.

3.5.1.2 Android 4.4 Kitkat Keymaster

The Trustonic TEE software package now includes support for Android 4.4 keymaster with its implementation strengthen by ARM TrustZone® through a Trusted Application (TA) in the Secure World and a shared library in the Normal World. It is up to the Trustonic TEE integrator to include this TA and shared library in the device software image.

This keymaster feature is optional to be integrated in a Trustonic TEE integration.

The Trustonic TEE software package for KitKat Keymaster includes:

- a shared library: libMcTeeKeymaster.so
- a Trusted Application: 07060...0.tlbin

The implementation supports the following algorithms:

- RSA with key size up to 4096 bits
- DSA algorithm with support of DSA key sizes from 512 to 3072bits ECDSA with NIST P-192, P-224, P-256, P-384 and P-521 curves.

Further documentation is described in the Trustonic TEE integration guide.

3.5.1.3 Endorsement

When the authentication token is removed by the RootPA, a backup copy is used by the daemon for the endorsement.

3.5.1.4 Trusted User Interface Enhancements

The Trusted User Interface core module supports PNG images with alpha channel.

3.5.2 New Integration Features

3.5.2.1 Support of ARMv8 Application Processors

Trustonic TEE is compatible with the new ARMv8 architecture and now supports Cortex-A53 and Cortex-A57-based application processors. Trustonic TEE running on ARMV8-based chipsets has been designed to execute in conjunction with the ARM Trusted Firmware (https://github.com/ARM-software/arm-trusted-firmware).

The ATF Secure Dispatcher can be found in the product package under the "*ATF*" directory.

Even if the ATF running at EL3 is using the AArch64 instruction set, the Trusted Applications and drivers are executed with the AArch32 instruction set which means backward compatibility is retained for existing binaries.

This product version supports normal world components in 32-bits and 64 bits mode.

NOTE: ARM V8 support does not have any impact on existing Trustonic TEE APIs.

Testing of the functionality is provided by the correct execution of Trustonic TEE test suite on ARMV8-based chipsets.

3.5.2.2 Memory Extension for Secure Drivers

The virtual space accessible to Secure Drivers has been increased to 30MB.

Drivers can map large sections of memory from D9 (0x800000) to D30.

Testing of the functionality is provided by the correct execution of drivers using this functionality.

3.5.2.3 Trusted User Interface Enhancements

- The TUI Normal-World Driver is split into generic and platform dependent parts.
- All physical addresses are always using 64-bit integers to support large physical addresses.

3.5.3 Fixed Issues

Trusted User Interface

- TBUG-125 Possible stack overflow due to recursive function

Power Management

- TBUG-130 fcHandler_SWAP_CPU potential corruption of its stack
- TBUG-158 missing DMB barrier instruction after setting a new value of the currentCore

Crypto

- TD-267 GP Crypto API PSS sign and verify support non-empty salt
- TBUG-172 Potential for function endorse() to overwrite its own memory
- TBUG-229 tlApiMessageDigestInitWithData correctly parses the Data parameter

Daemon

- TBUG-177 RECV_PAYLOAD_FROM_CLIENT() and Connection::readData fixed error handling issue.

Linux Driver

- TBUG-233 overflow of the counter used for handles was not properly managed

3.6 TRUSTONIC TEE V300

3.6.1 New Core Features

Trustonic TEE V300 introduces the following new features and improvements:

Trusted User Interface (TUI)

The Trustonic TEE Internal API supports a new API for the Trusted User Interface. The TUI API allows Trusted Applications to retrieve securely inputs from the end-user and to securely displays data to the device display.

This API is available to both Legacy and GlobalPlatform Trusted Applications

Trustonic TEE comes with a new unified Secure Driver template for the Trusted User Interface to ease the porting of the TUI on the silicon platform.

<t-play DRM API

The Trustonic TEE Internal API supports a new DRM API for processing DRM content. This API allows Trusted Applications to decrypt and play media content through the secure media components of the platform.

This API is available to both Legacy and GlobalPlatform Trusted Applications

Trustonic TEE comes with a new unified Secure Driver template for DRM to ease the porting on the silicon platform.

GlobalPlatform API

Trustonic TEE V300 supports GlobalPlatform APIs including:

- GlobalPlatform Client API
- GlobalPlatform Cryptographic API
- GlobalPlatform Trusted Storage API
- GlobalPlatform Memory Management API

The list of functions which are supported is indicated in “t-base – API Documentation”.

Memory Management

Trusted Applications can declare and use a heap for dynamic memory management.

Increased number of Trusted Applications and Secure Drivers

Subject to memory availability, up to 19 Trusted Applications and 10 Secure Drivers can be loaded simultaneously.

Endorsement API

Trustonic TEE provides a new API tlApiEndorse() which allows a Trusted Application to sign data and prove that it is generated in a genuine TEE and in the right Trusted Application. This feature is important for service providers such as network operators. This feature is

documented in the Trustonic TEE Developer's Guide, Trustonic TEE API Documentation and Trustonic TEE Integration Guide.

Backward Compatibility

Trustonic TEE V300 provides backward compatibility with the previous APIs and binary compatibility for Trusted Applications and Secure Drivers.

3.6.2 New Integration Features

big.LITTLE MP compliance

Trustonic TEE is compliant with the big.LITTLE MP model. Trustonic TEE executes Trusted Applications and Secure Drivers on one core at a time but can migrate from core to core independently of the clusters.

Large Physical Address Extensions (LPAE) support

Trustonic TEE supports the Large Physical Address Extensions (LPAE) for platforms on which LPAE is supported.

Improvements for Secure Drivers

- The virtual space for drivers has been increased to 24MB.
- Functions have been added to the DrAPI to do the cache maintenance on a specific range of memory.

Fastcall handlers

2 Secure Drivers can now register a fastcall handler. This allows defining a fastcall handler for the silicon provider and a fastcall handler for the OEM.

3.6.3 Fixed Issues

Trustonic TEE V300-V005 fixes the following issues:

mcClientAPI

- FIX TBUG-119: mcCloseDevice()
  The device could be closed when active sessions were pending.

GP Client API

- FIX TBUG-79 TEEC_MEMREF_WHOLE
  An incorrect parameter type was given to the Trusted Application when a whole memory reference is used as parameter.
- FIX TBUG-52 daemon
  The daemon failed to install SP Trusted Applications because it was using the read-only registry (in /system/app/mcRegistry).
- FIX TBUG-50 TEEC_InvokeCommand() did not allow passing NULL as parameter

- FIX TBUG-77 The flags were not checked in TEEC_RegisterSharedMemory() and TEEC_AllocateSharedMemory()
- FIX TBUG-76 When using TEEC_MEMREF_WHOLE, the size must be updated only if the parent shared memory is output or inout.

Fixes for LPAE

- FIX TBUG-96 Linux driver
  phys_to_page() was called with the MMU descriptor instead of the physical address.
- FIX TBUG-82 Linux driver
  MCI mapping to daemon downcasts a paddr -> pfn to 32 bits before calling remap_pfn_range
- FIX TBUG-81 mcClientLib
  mcMallocWsm() could fail if the allocated physical buffer's address was outside 32 bit range
- FIX TBUG-80 Core
  MCP sched flags information from RTM to kernel truncated physical addresses to 32bits

Crypto

- FIX TBUG-86 the CR exception handler thread had the same priority as the CR worker thread and prevented to resume after an exception.

Core

- FIX TBUG-38 DrSdk had a wrong value (outdated) of FASTCALL_OWNER_SIP
- FIX TBUG-40 MobiConvert crashed when it could not find required files
- FIX TBUG-11 GIC save and restore was broken when NUM_HW_INTR is bigger than GIC_DIST_NUM_INTR: 256
- FIX TBUG-88 MTK crashed when stopping intr 32769

Trusted User Interface

- FIX TBUG-55 Infinite loop in inflate library state machine
- FIX TBUG-56 Critical buffer overflow in PNG library
- FIX TBUG-57 SegFault in PNG Library
- FIX TBUG-103 Secure memory dumped on LCD screen (Arndale)

Linux Driver

- FIX TBUG-123 Kernel displayed RCU warnings when daemon initialized

Trustonic TEE V300-V004 fixes the following issues:

- It is no longer necessary to set the endorsement key when MobiConfig is used to configure the Trustonic TEE image – CR-248
- MobiConfig's default KID is now 1 - TD-86
- Fixed a race condition in the startup of the daemon - MCTWO-2513
- Fixed two minor Coverity defects in the daemon - CID 10677 10678.

- Minor fixes for GP storage - MCTWO-2505 MCTWO-2506 MCTWO-2519 MCTWO-2521 MCTWO-2523
- Minor fixes for GP client API - TD-114 TD-117
- tlApiEndorse() now accepts a NULL message - TSEC-157
- Platforms with LPAE can have the same number of TAs and drivers - MCTWO-2340
- Add support for strongly ordered mapping - MCTWO-2525
- phys_addr_t size was not correct in McLib - MCTWO-2526
- rfu parameter in processDrmContent() is now used
- More sanity checks adde for RSA parameters
- race conditions on Session Identifiers in Crypto Driver - TSEC-93
- When using LPAE, drivers can only map after D8 blocks which are 2MB aligned - MCTWO-2533
- warmBoot resume code uses main stack - MCTWO-2542
- Crypto Driver uses client buffer for internal endorsement operations – TBUG-15
- Potential memory leak reported by code analysis tool, in kernelApi - TBUG-8
- Malicious TA can disable System logging - reset the line size as well when the log is 0 - TBUG-43
- TUI components have been updated to include all the fixes and enhancements since Trustonic TEE V300-V001

Trustonic TEE V300-V003 fixes the following issues:

- Removed optimization options for GCC -fdata-sections & -ffunction-sections which created alignment issues - TD-104
- Fixed the alignment of the GP properties to 4-byte - MCTWO-2470
- Security fix for LPAE: the Normal World could set the NS bit to 0 when registering the shared buffers - MCTWO-2478
- Crypto driver cleans the stack after each command - TSEC-155
- Fixed 1MB section mapping with MAP_UNCACHED attribute - MCTWO-2484
- Fixed AES CTR mode: if data smaller than block size (16 byte) is provided, cipher update is expected to encrypt/decrypt the amount of data and return length field accordingly - TFAE-118

Trustonic TEE V300-V002 fixes the following issues:

- Device key generated on MTK platform was invalid - TSEC-171
- Issue in doTzbspSymCipherInit when we free context and fix code to detect when the old method to use hardware key is not available (Qualcomm Platform) - MCTWO-2033
- The Normal World was notified too often - MCTWO-2452
- PSS signature/verification could crash the GP TA
- The kernel could crash because buffer overflow when LPAE is activated
- GCC build generates extra section (option -fdata-sections). The extra .bss.* sections are moved to .bss section - TD-104

Trustonic TEE V300-V001 fixes the following issues:

- Error when loading Trusted Application binaries bigger than 16KB - MCTWO-2141

- Potential error when loading Trusted Applications under low-memory conditions resulting in the unavailability of the Secure-World - MCTWO-2327
- Potential stack overflow in the Secure Cryptographic Driver for RSA operations resulting in the unavailability of the Secure-World - MCTWO-2112
- Potential crash in the Normal-World Daemon under low-memory conditions - MCTWO-2334
- Potential crash in the Normal-World Kernel API under low-memory conditions - MCTWO-2300
- Potential deadlock in Secure Drivers when calling drApiWaitForIntr() with ANYINTR - MCTWO-2252
- Function drApiGetClientRootAndSpId() does not return the correct return code - MCTWO-2119
- Potential memory leak in the Normal-World Daemon if the TLC client crashes - TFAE-40

3.7 TRUSTONIC TEE-202

3.7.1 New Core Features

Client Kernel API

Trustlet Connector API can be used from a Linux driver through the Kernel API module.

3.7.2 New Integration Features

Secure Driver access permission

Drivers may require to have a way to restrict access to certain trustlets. This CR adds the drApiGetClientRootAndSpId function to the Driver API so that a driver can know which trustlet is calling it.

Chip Unique Value for Trustlets

The tlApiDeriveKey function has been added to the Trustlet API. This allows trustlets to securely derive a chip unique value for their internal use.

Introduce Firmware Driver and FastCalls in Trustonic TEE

This adds the possibility for the silicon provider to implement a specific driver, called the Firmware Driver, which can intercept so-called FastCalls from the normal world, i.e. calls to the secure world which don't trigger a full context switch.

3.7.3 Fixed Issues

MCTWO-313 – Wrong error code when TL/DRV binary cannot be found

- mcOpenSession now actually returns MC_DRV_ERR_TRUSTLET_NOT_FOUND when the trustlet or the driver cannot be found.

MCTWO-1726 – TLCs can map WSMs twice thus causing some maps not to be freed

- The Linux driver now prevents a WSM to be mapped twice. This could lead to memory leaks in the kernel and even normal world security issues.

MCTWO-1786 – cacheInstInvalidateAll() must call set_CP15_ICIALLUIS() as we are running in a multiprocessing system

- All cores in a multiprocessing environment must be notified that Trustonic TEE wishes to invalidate cache lines. This fix makes sure that an inner shared cache invalidation instruction is issued.

MCTWO-2097 – DSB and BPIALLIS are missing in space_l1_kmap()

- According to the ARM Architecture Reference Manual, an Inner Shared Branch predictor Invalidation and a Data Synchronization Barrier are required when updating MMU entries.

MCTWO-2098 – flushBTAC() must call set_CP15_BPIALLIS() as we are running in a multiprocessing system

- An Inner Shared operation must be issued to make sure that the Branch Target Address Cache is actually flushed on all cores in a multiprocessing environment.

TSEC-1 – Uninitialized members in the Linux driver

- While this is not a security issue per se, it is good practice to ensure that structure members such as pointers are properly initialized.

TSEC-2 – Fix memory leaks in Linux driver and daemon

- A memory leak was found and fixed in the cleanup code for session buffers in the daemon.

TSEC-76 – mcRegistryCleanupTrustlet() segfaults if passed NULL for UUID

- Proper parameter validation has been added to the client library. MC_DRV_ERR_INVALID_PARAMETER is now returned in case a NULL is UUID is provided.

3.8 TRUSTONIC TEE-201

3.8.1 New Core Features

The maximum number of Trusted Application and Secure Driver sessions have been increased to 10 Trusted Applications and 5 Secure Drivers.

3.8.2 Fixed Issues

MCTWO-1648 – Remove limit on L2 tables

MCTWO-1664 – Drivers to unmap trustlet memory on trustlet close

MCTWO-1705 – Session limit counter counted incorrectly

MCTWO-1739 – Fix assertion failures

MCTWO-1741 – Limit L2 table creation to match MM requirements

MCTWO-1779 – Fix performance issues

MCTWO-1864 – Trustonic TEE fails with high session limits

MCTWO-1896 – Remove unintended heap use from Trustonic TEE 200

MCTWO-1877 – Crashes in randomized testing

MCTWO-1887 - _mutex_* functions needed for ARM library

MCTWO-1997 – Define IO mappings as not executable in MMU tables