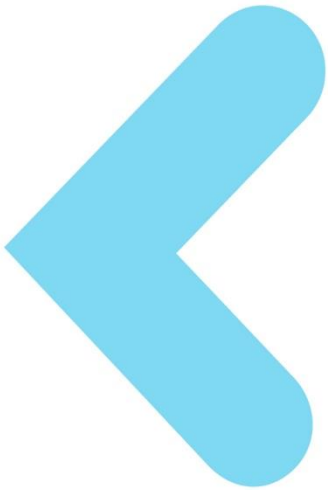
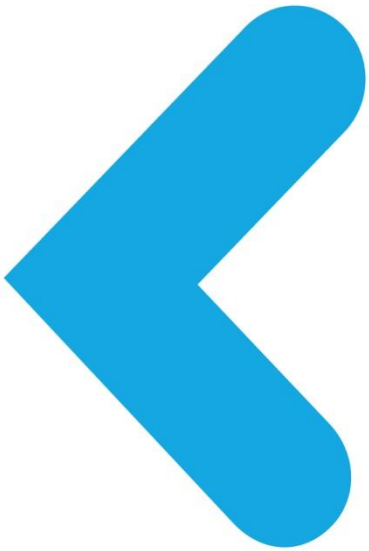
TRUSTONIC

Release Notes



# Kinibi v400A for EXYNOS64

# Release Notes



# PREFACE

This document is the confidential and proprietary information of Trustonic ("Confidential Information"). This document is protected by copyright and the information described therein may be protected by one or more EC patents, foreign patents, or pending applications. No part of the document may be reproduced or divulged in any form by any means without the prior written authorization of Trustonic. Any use of the document and the information described is forbidden (including, but not limited to, implementation, whether partial or total, modification, and any form of testing or derivative work) unless written authorization or appropriate license rights are previously granted by Trustonic.

TRUSTONIC MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF SOFTWARE DEVELOPED FROM THIS DOCUMENT, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. TRUSTONIC SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS DOCUMENT OR ITS DERIVATIVES.

# TABLE OF CONTENTS

# 1 INTRODUCTION

This document is the Release Notes for the Trustonic Kinibi product on Samsung S.LSI ARMv8 platforms with a normal world in 32 bits or 64 bits.

The exact version of the product is:
t-base-EXYNOS64-Android-400A-V012-20180515_143127_49104_78462.

This version is a Commercial Release. It has been fully validated and no issue found.

# 2 What's New in Kinibi

## 2.1 Kinibi v400A

### 2.1.1 New Core Features

400A introduces new features. The TEE API level is changed to **11**.

This Kinibi version is compliant to:

- GlobalPlatform TEE Client API Specification ([Client API]) v1.0, GlobalPlatform TEE Client API Specification v1.0, Errata and Precisions v2.0, GPD_EPR_028.
- GlobalPlatform TEE Internal Core API Specification ([Internal Core API]) v1.1.1.

It passes the FIME GP compliance tool.

#### 2.1.1.1 GP Properties

400A implements all the GP Properties defined by the GP internal API 1.1.1.

Note that `gpd.tee.description` does not anymore contain `<t-base`, but the product build id, e.g. `t-base-Arndale-Android-400A-20160501_011308_9060_36620`.

Note also that `gpd.tee.firmware.implementation.version` and `.binaryversion` are values that need to be provided by the underlying platform for a device to be GP compliant.

See the Kinibi API Documentation for the complete list of properties defined in each version of the product.

#### 2.1.1.2 TEE Capabilities

400A implements new proprietary properties via the GP properties API that give information about the capabilities of the TEE on this specific device. For example:

| | | |
|---|---|---|
| `com.trustonic.tee.isa.arm.neon` | `Boolean` | `True if Kinibi supports NEON and Hardware Floating Point for TA`<br>`Dynamic at build time` |
| `com.trustonic.tee.tui.available` | `Boolean` | `True if TUI is available`<br>`Dynamic, will try to contact TUI driver` |

See the Kinibi API Documentation for the complete list of properties defined in each version of the product.

#### 2.1.1.3 GP Properties Enumeration

400A supports the GlobalPlatform properties enumeration API.

#### 2.1.1.4 GP Time API

400A supports the full GP Time API, including the `TEE_Wait()` function that was previously not supported.

#### 2.1.1.5 GP TA Instance Types

400A supports Single Instance Trusted Applications as well as Multi-Session and Keep-alive TAs. Respective configuration can be set via the new TA manifest.

#### 2.1.1.6 GP Internal Client API

400A supports the GP TA-to-TA communication using the `TEE_OpenTASession()`, `TEE_InvokeTACommand()` and `TEE_CloseTASession()` functions.

Any GlobalPlatform TA can use this API to call another GP TA (any TA can be a *client*).

It depends on the way a TA is installed if the TA can be called in TA-to-TA communication (only some TAs can be a *server*).

For a GlobalPlatform TA to be a server, the TA must be already running or installed into Trusted Storage (TA-to-TA is not loading automatically System TA and SP TAs installed in mcRegistry). To make sure a TA can be called independently of the way the TA is installed, the developer has to use a multi-session TA and first open a session from a Client Application before opening a second session from a Trusted Application.

#### 2.1.1.7 GP Crypto API

The following algorithmic key sizes have been added:

- `AES: 192 bits`
- `DES: 192 bits`

The following AES algorithms have been added:

- `TEE_ALG_AES_CTS`
- `TEE_ALG_AES_XTS`
- `TEE_ALG_AES_CCM`
- `TEE_ALG_AES_GCM`

The following ECDH algorithms have been added:

- `TEE_ALG_ECDH_DERIVE_SHARED_SECRET`

The following ECDSA_SHA algorithms have been added:

- `TEE_ALG_ECDSA_SHA1`
- `TEE_ALG_ECDSA_SHA224`
- `TEE_ALG_ECDSA_SHA256`
- `TEE_ALG_ECDSA_SHA384`
- `TEE_ALG_ECDSA_SHA512`

The following MAC algorithms have been added:

- `TEE_ALG_AES_CBC_MAC_NOPAD`
- `TEE_ALG_AES_CBC_MAC_PKCS5`
- `TEE_ALG_AES_CMAC`
- `TEE_ALG_DES_CBC_MAC_NOPAD`
- `TEE_ALG_DES_CBC_MAC_PKCS5`
- `TEE_ALG_DES3_CBC_MAC_NOPAD`
- `TEE_ALG_DES3_CBC_MAC_PKCS5`

The following missing GP APIs have been added or implemented:

- `TEE_GetOperationInfoMultiple()`
- `TEE_CopyOperation()`

- `TEE_ResetOperation()`
- `TEE_SetOperationKey2()`
- `TEE_AEInit()`
- `TEE_AEUpdateAAD()`
- `TEE_AEUpdate()`
- `TEE_AEEncryptFinal()`
- `TEE_AEDecryptFinal()`

#### 2.1.1.8 Performance optimizations for cryptographic operations

Kinibi-400A leverages ARMv8 AARCH32 crypto acceleration instructions to increase the efficiency of cryptographic operations. Also ARMv7 NEON accelerations are used when available.

This improves the speed of reads and writes of GP SecureStorage API and the speed of GP Crypto API when the following base algorithms are being invoked:

- AES
- SHA1
- SHA256
- SHA512
- RSA key generation

#### 2.1.1.9 Proxy enhancements

The proxy in 400A was enhanced to use zero-copy for shared buffers.

### 2.1.2 New Integration Features

#### 2.1.2.1 TA downgrade protection

Kinibi-400A supports downgrade protection for System TAs that do not use the GlobalPlatform API. This feature is an extension of the RPMB support of 311A and requires that the Kinibi Daemon can access the efs partition. See the Kinibi Integration Guide for more information.

#### 2.1.2.2 TEE Image builder

Kinibi-400A gives the SIP and OEM more flexibility to assemble and configure the TEE image. The new image builder in 400A allows exchanging the RPMB Monotonic Counter TA. The package contains a new folder SecureIntegration/t-base-kit that contains prebuilt TEE components and a python tool to assemble these files. This creates the TEE image. For more information, see the Kinibi Integration Guide.

#### 2.1.2.3 New ATF Input Fastcalls for GlobalPlatform

For a device to be GlobalPlatform-compliant, the TEE must return the exact version of the firmware in the `gpd.tee.firmware.implementation.version` and `.binaryversion` properties. 400A adds a way for platform integrators to define these during the integration. In the case of ATF-based integrations, new IDs for TBASE_SMC_FASTCALL_INPUT have been defined to retrieve such version information.

#### 2.1.2.4 TTS - Trustonic Test Suite

The Kinibi-400A package contains the TTS that SIP and OEMs must use to validate the product on development boards and production devices.

### 2.1.3 New SDK Features

#### 2.1.3.1 TA Manifest file

400A SDK supports a manifest file for GP TAs that allows specification of static properties.

#### 2.1.3.2 TUI double buffering for GP TAs

400A SDK supports the TUI double buffering API for TAs that use the GP API.

#### 2.1.3.3 Downgrade protection flag for legacy System TAs

400A SDK supports the new MobiConvert flag `--downgrade-protected`. TAs that have this flag set will only be loaded on Kinibi versions that have the TA downgrade protection activated.

#### 2.1.3.4 TeeClient

400A SDK contains the TeeClient, an in-APK library for downloadable and native proxy access.

#### 2.1.3.5 Assembler support for TAs

400A SDK supports building and linking assembler files into TAs.

#### 2.1.3.6 New samples

The following samples have been added:

- **CryptoCatalog_GP**: Demonstrate usage of cryptographic APIs using the GlobalPlatform APIs.
- **GP**: Demonstrate TA manifest, TA-to-TA communication, usage of Trusted Storage and GP Properties.
- **PinpadGP**: Implementation of the Pinpad sample using a GP TA and the Trustonic TUI APIs for GlobalPlatform.

### 2.1.4 Fixed Issues

# 3 PAST KINIBI RELEASES

## 3.1 Kinibi v311B

### 3.1.1 New Features

311B is the update of Kinibi for Android Nougat. The TEE API level is unchanged.

This release is not adding any new feature.

#### 3.1.1.1 Performance improvements for GP registered shared memories

The management of the registered shared memories has been reworked in order to improve the performances of TEEC_InvokeCommand().

#### 3.1.1.2 Paths in Android File System

The paths of the Kinibi binaries, libraries and registries had to be changed to follow the new recommendations.

| Component | Name of the binary | Old Containing folder | New Containing folder |
|---|---|---|---|
| Kinibi Daemon *(32 or 64 bit)* | `mcDriverDaemon` | `/system/bin` | **`/vendor/bin`** |
| Kinibi Proxy *(32 or 64 bit)* | `trustonic_tee_proxy` | `/system/bin` | **`/vendor/bin`** |
| Root Provisioning Agent *32 bit* | `RootPA.apk` | `/system/app/` | `/system/app/` |
| Kinibi Client library *(32 or 64 bit)* | `libMcClient.so` | `/system/lib/`<br>`/system/lib64/` | **`/vendor/lib/`**<br>**`/vendor/lib64/`** |
| Kinibi Registry library *(32 or 64 bit)* | `libMcRegistry.so` | `/system/lib/`<br>`/system/lib64/` | **`/vendor/lib/`**<br>**`/vendor/lib64/`** |
| Keymaster1.0 library *32 and 64 bit* | `keystore.$DEVICE.so` | `/system/lib/`<br>`/system/lib64/` | **`/vendor/lib/hw`**<br>**`/vendor/lib64/hw`** |
| Gatekeeper library *32 and 64 bit* | `gatekeeper.$DEVICE.so` | `/system/lib/`<br>`/system/lib64/` | **`/vendor/lib/hw`**<br>**`/vendor/lib64/hw`** |
| Root Provisioning Agent Native | `libcommonpawrapper.so` | `/system/lib/` | `/system/lib/` |

| | | | |
|---|---|---|---|
| library<br>*32 bit* | | | |
| Kinibi Read Only Registry | | `/system/app/mcRegistry/` | **`/vendor/app/mcRegistry/`** |
| Kinibi Read Write Registry | | `/data/app/mcRegistry/` | **`/data/misc/mcRegistry/`** |

### 3.1.2 Fixed Issues

Core

- TBUG-868 System halt related to drApiWaitForIntr().

Keymaster 1.0

- TBUG-880 For RSA PSS signatures, do not hard code the salt length to 20 bytes even if it was compliant with the specifications, it does not work with the most recent versions of the CTS. The salt length is now equal to the digest length, except for MD5, it uses 20 because the digest is too short (16).
- TSEC-261 buffer overrun in TA
- (find_param): Fix preliminary error reporting
- (update): Fix chunking of operations
- (aes_finish): Set output length correctly
- (open) Don't use throwing `new'.
- Remove KM_TAG_ROOT_OF_TRUST from KEY_CREATION_ALLOWED_TAGS

Gatekeeper

- TBUG-757 throttling must be done on SWd and failure records must be stored in the SWd
- TBUG-744 Enroll() and Verify() should be implemented in TA

SDK

- TBUG-869 TEE_GetInstanceData() does not return NULL on first invocation, if TA API_LEVEL >= 8

Trusted User Interface

- TBUG-875 Support for NWd resolution change
- TBUG-863 Ghost TUI activity
- TBUG-837 TUI activity creation failure

ATF

- TBUG-866 memory corruption affecting FiqForward scenario due to misalignment between AFT/SPD and TEE

## 3.2 Kinibi v311A

### 3.2.1 New Features

311A introduces new features. The TEE API level is changed to **10**.

#### 3.2.1.1 Trusted Storage with Rollback Protection

The Trusted Storage which is available using the GlobalPlatform Trusted Storage API has been enhanced to include support for Rollback Protection.

The silicon vendor and the OEM need to implement a RPMB driver and configure Kinibi image to use this feature. See the Kinibi Integration Guide for more information.

#### 3.2.1.2 Trusted Storage Upgrade

Kinibi-311A introduces an automatic conversion tool that allows silicon vendors and device manufacturers to upgrade existing devices from previous versions to 311A.

This conversion tool reformats files stored using the GlobalPlatform Trusted Storage APIs with version 300A to the format of version 310A. The tool is integrated into the Kinibi image and into the Kinibi Daemon and is automatically run on each startup.

#### 3.2.1.3 Post-Mortem Debug and Performance Analysis Tool

Kinibi-311A adds more system debugging support by the addition of tee-ps tool and the TEE DebugSession infrastructure.

The product package contains the **tee-ps** tool in **/t-base-dev-kit/Tools/TlcTeePs** with Src and Bin. See the Kinibi Integration Guide for more information on how to use this tool.

#### 3.2.1.4 GP time API

Kinibi-311A supports more functions of the GP Time API:

The function TEE_GetSystemTime() is now monotonic between two resets.

The functions TEE_GetTAPersistentTime and TEE_SetTAPersistentTime are now supported.

#### 3.2.1.5 Multiple OEM Keys

It is possible to inject up to 32 OEM keys in the Kinibi Core image.

#### 3.2.1.6 GICv3

The Kinibi Core now supports the new generation of ARM Generic Interrupt Controller, GIC-500.

The boot loader must pass the GIC version to the Kinibi Core.

#### 3.2.1.7 GP Client API in Kernel Module

The GP Client API is now available for the Linux kernel.

The API is defined in AndroidIntegration/Src/gud/MobiCoreDriver/public/GP/tee_client_api.h.

The name of the functions and types follow the Linux kernel coding rules.

#### 3.2.1.8 Keymaster M-MR1

The Keymaster for Android M can now support up to 16 cryptographic operations in parallel.

### 3.2.2 Fixed Issues

Core

- TBUG-815 broken FastCall handling due to plat_fc_secondary_core_handler() corrupting regs->r1
- TBUG-820 drApiMapPhysicalBuffer can't map a physical address above 0xFFFFFFFF
- TBUG-802 The TA built with 302C TlSdk is not compatible with Kinibi 310A and later
- TBUG-722 MCP command timeout
- TBUG-777 Boot trace do not show anymore
- TBUG-784 Tee debug image too big
- TBUG-760 Memory leak in MTracker
- TBUG-711 Multiple vulnerabilities in drApiMapTaskBufferImpl
- TBUG-754 MTK does not update timeout for thread in special IPC case
- TBUG-747 SFS assert in L2
- TBUG-718 Integer overflow in RTM's llDecryptAndVerify_SP_TA()
- TBUG-719 RTM does not handle buffer offset correctly in llSafeCopyServiceBlob() and llSafeCopyAdditionalServiceBlob()

Crypto

- TBUG-673 HMAC and digest operations with short tags lead to buffer overruns
- TBUG-689 CR asserts on 0 length hash buffer
- TBUG-743 Insufficient bound checks in static_InjectAttribute
- TBUG-710 Integer overflow in map2Buffers
- TBUG-396 Wrong driver ID usage for the crypto driver

Keymaster

- TBUG-790 Keymaster M: Ignore KM_TAG_CREATION_DATETIME
- TBUG-793 Keymaster M,N: begin() should succeed if KM_TAG_AUTH_TOKEN is not present in the operation parameters and KM_TAG_AUTH_TIMEOUT is not present in the key parameters

Linux driver

- TBUG-804 The time field is not initialized when the MCP buffer is given to the SWd.
- TBUG-792 NWd driver mmap's memory beyond allocated .

Legacy Client API

- TBUG-782 NWd client: add support for MC_DRV_ERR_NO_FREE_INSTANCES

GP Client API

- TBUG-699 NWd client GP: fix memory leak and incorrect origin

Daemon

- TBUG-750 NWd daemon: need to close device on thread exit so driver can cancel pending requests

## 3.3 Kinibi v310C

### 3.3.1 New Features

310C introduces new features. The TEE API level is changed to **9**.

#### 3.3.1.1 Trusted User Interface Service binding

The TUI service is no longer automatically started at boot time.

Client Applications must bind to the TUI Service before starting the communication with the Secure World. This can be transparently achieved by calling the new function TEEC_TT_RegisterPlatformContext() added in this release.

#### 3.3.1.2 Proxy

The Kinibi proxy (trustonic_tee_proxy) is now a standalone process and must be started at boot time.

A new optional Authentication Service (TeeAuthServer) can be integrated in order to filter the access to the Secure World for the Client Applications.

### 3.3.2 Fixed Issues

Trusted User Interface

- TBUG-771 Tui session not stopped if TuiService killed during a TUI session.
- TBUG-767 TuiActivity is not killed if the framebuffer allocation is failing.
- TBUG-761 Memory leak in DrTui.

Crypto

- TBUG-758 During Secure Object creation, source data should not be copied into destination buffer.

Keymaster

- TBUG-706 Keymaster v0.4 testKeyStore_Encrypting_RSA_NONE_NOPADDING failed on Android M
- TBUG-762 Make KitKat Keymaster calls to tlApiWrapObject() immune to TBUG-758

Client Library (NWd)

- TBUG-715 When opening a session, return TEEC_ERROR_OUT_OF_MEMORY if errno is ENOSPC, not TEEC_ERROR_GENERIC.

Linux driver

- TBUG-714 Do not unblock caller for session close until [GP] session is closed in SWd.

Daemon

- TBUG-753 NWd daemon: fix check for user device presence

SDK

- TBUG-755 References to _stack_tlMain_* unprotected by TBASE_API_LEVEL check

## 3.4 Kinibi v310B

### 3.4.1 New Features

310B introduces new features. The TEE API level is changed to **8**.

#### 3.4.1.1 Android SE Proxy

The default Google SEAndroid policy blocks access to Linux kernel devices by Android Java applications. 310B provides a proxy for these situations to allow such applications to talk via Unix sockets to a new proxy component that will then access the Linux kernel device on behalf of the application. The use of the proxy is transparent; it is libMcClient.so that tries to access the kernel device first and falls back to proxy if necessary. The application only uses libMcClient.so as before.

#### 3.4.1.2 Android 6.0 Keymaster M and Gatekeeper

The Kinibi software package now includes support for Android 6.0 keymaster1 and gatekeeper with its implementations strengthened by ARM TrustZone® through Trusted Applications (TA) in the Secure World and shared libraries in the Normal World. It is up to the Kinibi integrator to include these TAs and shared libraries in the device software image.

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

The Kinibi software package for Marshallow Keymaster and Gatekeeper includes:

- shared library: `keystore.$DEVICE.so`
- shared library: `gatekeeper.$DEVICE.so`
- Trusted Application: `07060…04D.tlbin`
- Trusted Application: `070610…0.tlbin`

The implementation does not support optional or deprecated functions like:

- `delete_key()`
- `delete_all_keys()`

The implementation supports only the following import and export formats:

- Symmetric key
  - Import: `KM_KEY_FORMAT_RAW`
  - Export: not supported
- Asymmetric keys
  - Import: `KM_KEY_FORMAT_PKCS8`
  - Public key export: `KM_KEY_FORMAT_X509`
  - Private key export: not supported

Further documentation is described in the Kinibi integration guide.

#### 3.4.1.3 Stack protection for Trusted Applications and Drivers

310B introduces MMU-protected stacks for Trusted Applications and Drivers that use TBASE_API_LEVEL=8. In the SDK, when you select TBASE_API_LEVEL <= 7, the stack is allocated in the BSS of the application binary. When you select level 8, the binary only contains an integer with the minimum stack size. The startup code of the application will allocate a stack with one unmapped MMU page before and after the stack area. That way, stack overflows and underflows will not silently overwrite global variables and heap, but cause a segmentation fault that helps discover stack problems during the development phase.

#### 3.4.1.4 DebugFS interface for MCP timeout and core switching

The kernel module for 310A removed a feature to trigger a core switch on the command line. The new virtual file /trustonic_tee/active_cpu reintroduces this feature.

The kernel module for 310A implements a 50s watchdog for MCP commands in the SWd. With 310B, the /trustonic_tee/mcp_timeout virtual file allows to modify this timeout, e.g. for debugging purpose.

Find more information in chapter 8 "SYSTEM DEBUGGING WITH DEBUGFS" of the integration guide.

#### 3.4.1.5 Calling the Firmware from a Driver

The new API, `DrApiCallTrustedFirmware()` can be used to call functionality of the Firmware from inside a driver. This is mostly intended for ARMv8 platforms that run the generic ARM Trusted Firmware in EL3 mode.

#### 3.4.1.6 Add support for timeouts in TlApi

It is now possible to use a timeout value in ms for `tlApiWaitNotification`().

Only an immediate or infinite timeout was supported until now.

### 3.4.2 Fixed Issues

Core

- TBUG-135 drApiThreadSleep with a timeout value besides 0 and INFINITE is undefined
- TBUG-178 Separate Code and Data pages in MTK
- TBUG-470 SWd Kernel clips thread priority and this causes ISR not to run in expected order
- TBUG-645 random failures when using DSA key generated with TEE_GenerateKey()

Normal world

- TBUG-600 The Client Library is not able to map the same buffer multiple times
- TBUG-623 TUIActivity creation problem
- TBUG-657 A loop is required for send() or sendmsg() for Unix socket in the Daemon

### 3.4.3 Deprecations

Kinibi 310B deprecates the use of API LEVEL 1-4 for Secure Drivers. Driver developers must use the new APIs for memory extension introduced in Kinibi 302A and the stack protection introduced in Kinibi 310B. The next major Kinibi release will not support API LEVELS 1-4 for Secure Drivers.

## 3.5 Kinibi v310A

**WARNING: the new format of the Trusted Storage introduced in version 310A is not compatible with previous version. This means data stored on a device using the GlobalPlatform Trusted Storage API with version 302A or earlier cannot be read if the device is upgraded to 310A. Therefore Trustonic recommends silicon vendors and device manufacturers to only apply version 310A on new devices and not to upgrade existing devices with 310A. Trustonic will provide a compatibility tool to fix this issue in forthcoming 310B release. Note this issue does not affect data stored with the Secure Object API.**

### 3.5.1 New Features

310A introduces new features. The TEE API level is changed to **7**.

#### 3.5.1.1 Trusted Storage

The Trusted Storage which is available using the GlobalPlatform Trusted Storage API from GP Trusted Applications has been optimized.

The Trusted Storage is using a sophisticated B+ tree which allows an efficient retrieval of the persistent objects with a reduced number of I/O operations.

The persistent object enumeration functions are now fully supported. It is also possible to rename a persistent object.

The Trusted Storage API is now also available for Secure Drivers (DrApi).

The following functions have been added:

- `TEE_AllocatePersistentObjectEnumerator()`
- `TEE_FreePersistentObjectEnumerator()`
- `TEE_GetNextPersistentObject()`
- `TEE_ResetPersistentObjectEnumerator()`
- `TEE_StartPersistentObjectEnumerator()`

The following functions defined in the GP1.1 Internal Core API specification have been added:

- `TEE_GetObjectInfo1()`
- `TEE_RestrictObjectUsage1()`
- `TEE_CopyObjectAttributes1()`
- `TEE_CloseAndDeletePersistentObject1()`

#### 3.5.1.2 Trustonic KPHv2 support

Device manufacturers can use KPHv1 or KPHv2 with this version of the product in order to provision the TEE Device Binding key with the Key Provisioning Host during device manufacturing.

The CMTL and the MobiConfig tool have been updated in order to support the KPHv2.

#### 3.5.1.3 Legacy Crypto API

The following RSA OAEP algorithms have been added:

- `TLAPI_ALG_RSA_SHA1_OAEP`
- `TLAPI_ALG_RSA_SHA224_OAEP`
- `TLAPI_ALG_RSA_SHA256_OAEP`
- `TLAPI_ALG_RSA_SHA384_OAEP`
- `TLAPI_ALG_RSA_SHA512_OAEP`
- `TLAPI_ALG_RSACRT_SHA1_OAEP`

- TLAPI_ALG_RSACRT_SHA224_OAEP
- TLAPI_ALG_RSACRT_SHA256_OAEP
- TLAPI_ALG_RSACRT_SHA384_OAEP
- TLAPI_ALG_RSACRT_SHA512_OAEP

The following RSA PKCS1 algorithms have been added:

- TLAPI_SIG_RSA_SHA1_PKCS1
- TLAPI_SIG_RSA_SHA224_PKCS1
- TLAPI_SIG_RSA_SHA256_PKCS1
- TLAPI_SIG_RSA_SHA384_PKCS1
- TLAPI_SIG_RSA_SHA512_PKCS1
- TLAPI_SIG_RSACRT_SHA224_PKCS1
- TLAPI_SIG_RSACRT_SHA256_PKCS1
- TLAPI_SIG_RSACRT_SHA384_PKCS1
- TLAPI_SIG_RSACRT_SHA512_PKCS1

The following RSA PSS algorithms have been added:

- TLAPI_SIG_RSA_SHA224_PSS
- TLAPI_SIG_RSA_SHA384_PSS
- TLAPI_SIG_RSA_SHA512_PSS
- TLAPI_SIG_RSACRT_SHA224_PSS
- TLAPI_SIG_RSACRT_SHA384_PSS
- TLAPI_SIG_RSACRT_SHA512_PSS

The following HMACs algorithms have been added:

- TLAPI_ALG_HMAC_SHA224
- TLAPI_ALG_HMAC_SHA256
- TLAPI_ALG_HMAC_SHA384
- TLAPI_ALG_HMAC_SHA512
- TLAPI_ALG_HMAC_MD5

The following digest algorithms have been added:

- TLAPI_ALG_MD5
- TLAPI_ALG_SHA224
- TLAPI_ALG_SHA384
- TLAPI_ALG_SHA512

#### 3.5.1.4 GP Crypto API

The following RSA OAEP algorithms have been added:

- TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA1
- TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA224
- TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA256
- TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA384
- TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA512

The following RSA PKCS1 algorithms have been added:

- TEE_ALG_RSASSA_PKCS1_V1_5_MD5
- TEE_ALG_RSASSA_PKCS1_V1_5_SHA224
- TEE_ALG_RSASSA_PKCS1_V1_5_SHA384
- TEE_ALG_RSASSA_PKCS1_V1_5_SHA512

The following RSA PSS algorithms have been added:

- TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA224
- TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA384
- TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA512

The following HMACs algorithms have been added:

- TEE_ALG_HMAC_MD5
- TEE_ALG_HMAC_SHA224
- TEE_ALG_HMAC_SHA384
- TEE_ALG_HMAC_SHA512

The following digest algorithms have been added:

- TEE_ALG_MD5
- TEE_ALG_SHA224
- TEE_ALG_SHA384
- TEE_ALG_SHA512

The following DSA algorithms have been added:

- TEE_ALG_DSA_SHA1
- TEE_ALG_DSA_SHA224
- TEE_ALG_DSA_SHA256

The support for `TEE_DeriveKey()` has been added in this release for the `TEE_ALG_DH_DERIVE_SHARED_SECRET` algorithm.

RSA key size up to 4096 bits is supported.

The TEE Arithmetical API is supported. TEE_BigIntXXX().

#### 3.5.1.5 Normal World refactoring

The Normal World components have been refactored in order to simplify the design and ease the integration.

The UNIX socket between the user space client API (`libMcClient`) and the daemon (`mcDriverDaemon`) has been removed.

The NETLINK socket between the kernel space client API (KernelApi) and the daemon has been removed.

The kernel module `MobicoreKernelApi` was merged into the `MobicoreDriver` module.

A new directory `trustonic_tee` was added to Linux debugfs to help in system debugging.

#### 3.5.1.6 DRM API

Two new functions `tlApiDrmProcessContentEx()` and `TEE_TBase_DRM_ProcessContentEx()` have been added to the DRM API. They allow passing more parameters to the DRM driver.

More links (HDCP 2.1 and 2.2) can be checked with `tlApiDrmCheckLink()` or `TEE_TBase_DRM_CheckLink()`.

#### 3.5.1.7 Power management notifications to drivers

Drivers can register themselves with `drApiEnablePowerEvents()` in order to receive Power management transitions notifications.

For example, this is needed if the hardware must be turned off when the system is going to be suspended.

The driver will receive two new messages, `MSG_SUSPEND` and `MSG_RESUME`.

#### 3.5.1.8 Threading

Two new APIs, `drApiGetCurrentThreadId()` and `drApiGetThreadNo()` can be used to retrieve current the thread identifier and number.

#### 3.5.1.9 FIQ forward

It is now possible to customize which FIQs are forwarded by the TEE (EL1) to the Trusted Firmware (EL3).

The documentation for the FIQ forward mechanism can be found in the Kinibi integration guide.

#### 3.5.1.10 GP Time API

The following functions are supported:

- `TEE_GetSystemTime()`
- `TEE_GetREETime()`

The system time is based on REE-controlled timers.

#### 3.5.1.11 GP Login API

It is now possible to specify a `connectionMethod` for `TEEC_OpenSession()` different than `TEEC_LOGIN_PUBLIC`.
The following logins are supported:

- `TEEC_LOGIN_PUBLIC`
- `TEEC_LOGIN_USER`
- `TEEC_LOGIN_GROUP`
- `TEEC_LOGIN_APPLICATION`
- `TEEC_LOGIN_USER_APPLICATION`
- `TEEC_LOGIN_GROUP_APPLICATION`

In the TA, the function TEE_GetPropertyAsIdentity("gpd.client.identity") can be used to retrieve the identity of the client and perform access control.

#### 3.5.1.12 debugfs

In order to ease the debug of issues which are difficult to reproduce (random, long run, freeze…) the Linux driver now creates new entries under the `trustonic_tee` directory of the debugfs.

It is possible to retrieve the state of the TEE and a recent history. For example, the last SMCs, the last MCP commands or the sessions can be listed.

Please have a look at chapter 8 “SYSTEM DEBUGGING WITH DEBUGFS” of the integration guide.

# 3.6 Kinibi v303A

## 3.6.1 New Core Features

Kinibi v303A introduces new features. The TEE API level is changed to **6**.

### 3.6.1.1 Trusted User Interface

Double buffering:

The Trusted User Interface (TUI) driver (DrTui) is now using a double buffering with a back framebuffer and a front framebuffer. The benefit is to refresh the display without tearing.

The direct consequence for the OEM is that three secure buffers (two framebuffers, one working buffer) must be allocated (or reserved) for the DrTui instead of two (one framebuffer, one working buffer).

Additionally the number of threads in TUI secure driver has been increased from 4 to 5. It must be updated in the parameters given to mobiconvert.

New APIs have been added to TUI in order to take advantage of the double buffering. Trusted Applications will be able to draw and compose their images in the back framebuffer (`tlApiTuiDrawBuffer, tlApiTuiDrawImage and tlApiTuiFillRectangle`) and then flip the front and back buffers when ready. (`tlApiTuiFlipFrameBuffers`).

Existing Trusted Applications calling `tlApiTuiSetImage` are not affected by this change.

Raw buffer drawing:

A new API `tlApiTuiDrawBuffer` has been added in order to display a raw buffer. The previous APIs only allowed drawing PNG images.

Touch event queue:

The DrTui is now able to queue multiple touch events and an error will be returned to the Trusted Application in case of overflow.

## 3.6.2 Fixed Issues

Core

- TBUG-548 Fix in Trusted Application Container Parent Id (SpId) Management
- TBUG-576 Potential race condition on RTM startup between SIQH and MSH
- TBUG-590 MTK_FEATURE_FASTCALL_IDENTITY_MAPPING creates Secure mappings in 302B
- TBUG-592 Unfinished closeSession leading to a SWd hang
- TBUG-614 tlApiRandomGenerateDate(PLATFORM_RANDOM) has been fixed to always return an error.

Normal-World

- TBUG-567 – Fixed race condition in kernel driver. The fix applied to 302B has been reworked because it could lead to a dead-lock.
- TBUG-573 Correctly define MIN_NQ_LEN and MAX_NQ_LEN
- TBUG-581 fix for memory leak in mc_check_owner_fd()
- TBUG-610 invalid error handling of system call fault
- TBUG-611 Increase Notification Queue size from 16 to 64 elements

- TBUG-613 NWd daemon is linked against libMcRegistry

Trusted User Interface

- TBUG-393 Remove touch coordinates orientation conversion from core TUI.
- TBUG-467 Reject PNG containing an invalid DEFLATE stream
- TBUG-562 Service failing when started again
- TBUG-617 TUI IOCTL doesn't work when it's built as 64-bit

## 3.7 Kinibi v302B

### 3.7.1 New Core Features

Kinibi v302B is a maintenance release, the Kinibi API level is unchanged.

#### 3.7.1.1 Porting kit

The release contains an optional folder SecureIntegration/t-base-kit that allows modifying the ARMv7 power management integration for the given chip. The porting kit contains a build environment, prebuilt object files, header files and the source code of the porting layer. The porting layer implements the platform API that links the platform code to the core binary.

#### 3.7.1.2 Android lib curl update

Kinibi v302B now contains lib curl version 7.41.

#### 3.7.1.3 Alignment with Trustonic SDK-r7

The SDK supports the Kinibi API Level 5.

New features

- **Eclipse plugin:**
  A plugin for Eclipse IDE that intends to help Trusted and Client Applications development.
- **2 new samples:**
  Float: this sample shows how to use hardware or software floating points in a Trusted Application
  CryptoCatalog: this sample shows how to use the new cryptographic algorithms introduced with Kinibi v301 (DSA, ECDSA and RSA4096).

### 3.7.2 Fixed Issues

Kinibi Core

- TBUG-480 – Fixed the heap corruption in big TA.
- TBUG-486 – Fixed in heap extension mechanism.
- TBUG-497 – Fixed RTM crash in case of closing a SWd driver with pending TA connections.
- TBUG-503 – Fixed RTM crash in case of GP TA with more than 1MB stack.

Trusted User Interface

- TBUG-489 – Fixed crash of DrTui when HAL is built with GCC.

Normal-World

- TBUG-200 – Fixed hang of mc_open_device() in t-base-tui when Daemon is not up yet.
- TBUG-459 – Fixed hang in secure driver timeout.
- TBUG-482 – Fixed infinite loop in case of mcCloseSession error.
- TBUG-484 – Fixed race condition when resolving WSM at mcDaemon start.
- TBUG-499 – Fixed TUI starting hang with uninterruptable wait Zendesk #885
- TBUG-500 – Fixed wrong start order between TuiServer and Netlink Server.

- TBUG-502 – Fixed synchronization issue between TuiServer and t-base-tui when PM events fail the semaphore call
- TBUG-531 – Fixed deadlock in TuiServer when two intents arrive at the same time
- TBUG-543 – Fixed kernel crash when two clients use KernelApi
- TBUG-567 – Fixed race condition in kernel driver
- TBUG-545 - Kernel API crashes when a client attempts to open the device and another one closes a connection at the same time

## 3.8 Kinibi v302A

### 3.8.1 New TEE Core Features

Kinibi v302A introduces new features. The Kinibi API level is changed to **5**.

#### 3.8.1.1 Memory Extension

Kinibi v302A introduces an extended memory layout for Trusted Applications and Trusted Secure Drivers. With the extended memory layout, The Trusted Applications and Trusted Secure Drivers can use more than 1MB of code and data and their optional heap can also be extended. The maximum value is currently 120MB.

By default, this extended memory layout is not activated and must be explicitly requested in the makefile.

When using the extended memory layout, Secure Drivers must map memory using the following dedicated functions: drApiMapTaskBuffer(), drApiMapPhysicalBuffer(), drApiUnmapBuffer() and drApiUnmapTaskBuffers().

The compatibility between Trusted Applications and Secure Drivers using different memory layouts is as follow:

- A Trusted Application using the legacy memory layout can call a Secure Driver using the legacy memory layout.
- A Trusted Application using the extended memory layout cannot call a Secure Driver using the legacy memory layout.
- A Trusted Application using the extended memory layout can call a Secure Driver using the extended memory layout.
- A Trusted Application using the legacy memory layout can call a Secure Driver using the extended memory layout.

It is highly recommended to use the extended memory layout when writing new Secure Drivers.

The new memory layout is detailed in the Kinibi Developer's Guide and the Kinibi Driver Developer's Guide.

#### 3.8.1.2 42 Trusted Applications, 17 Secure Drivers

The total number of Trusted Applications which can be running simultaneously has been increased to 42.

The total number of Secure Drivers which can be running simultaneously has been increased to 17.

#### 3.8.1.3 Neon, Hardware Floating-Point Unit

Trusted Applications and Secure Drivers can now use the FPU registers and use the Neon or FPU hardware engines in the Secure World.

This is available for both ARMv7 and ARMv8 platforms but may not be supported by all the Systems-on-Chip.

Note that for ARMv8, Kinibi in the Secure World is running in Aarch32 mode and not all the FPU registers are available for the Trusted Applications and Secure Drivers.

This feature is documented in the Kinibi Developer's Guide and the Kinibi Driver Developer's Guide.

#### 3.8.1.4 Init entry point for Trusted Applications

It is possible to define an optional _init()entry point which is called before the tlMain() or TA_CreateEntryPoint() entry point. This is described in the Kinibi Developer's Guide.

#### 3.8.1.5 Design Enhancements

The Trusted Storage Driver is now built into Kinibi. Its privileges have been decreased so that it cannot map physical memory.

Kinibi can load Trustonic modules from the Trusted Storage. The modules are verified with Trustonic public key included in Kinibi.

#### 3.8.1.6 ARMv8 FIQ dump

In case of a crash of the Linux kernel, an FIQ (Secure Interrupt) can be used to analyze postmortem the state of the NWd OS.

When this interrupt is triggered (typically by a watchdog), a special callback `plat_tbase_dump`() is called in the ARM Trusted Firmware (ATF).

#### 3.8.1.7 Add support for timeouts in TlApi and DrAPI

For some platforms, it is now possible to use a timeout value in ms for `tlApiWaitNotification`(), `drApiThreadSleep`() or `drApiWaitForIntr`().

Only an immediate or infinite timeout was supported until now.

#### 3.8.1.8 Debug improvements

If a Trusted Application is built with the debugable flag, it is now possible to retrieve more information in case of a crash.

The UUID, the exception type or the value of general purpose registers can be retrieved in the Linux kernel logs (dmesg). See the developer guide for more information.

In case of Kinibi halt, the daemon is printing useful information, such as the cause of the fault or the UUID of the driver which provoked the halt.

### 3.8.2 Fixed Issues

Kinibi Core

- TBUG-369 – Fixed FIQ priority in GIC.
- TSEC-203 - use time invariant memory compare for signature verification.
- TBUG-417 – Check that Secure Drivers blob sent by Mobiload are aligned before loading them in Kinibi.
- TBUG-426 - TEE_GenerateKey might panic the Trusted Application when used with RSA in some conditions.
- TBUG-424 - tlApiWrapObjectExt was failing if the parameter consumer was different than NULL.

Trusted User Interface

- TBUG-248 – Support cancelation of Trusted User Interface session opening.

Normal-World

- TBUG-357 - mcWaitNotification returns an error status when a signal occurs.

- TBUG -374 - Kinibi Normal-World Driver might not start Kinibi when used with Large Physical Addresses.
- TBUG-405 – Fix potential memory corruption in GP Client API for TEEC_OpenSession and TEEC_InvokeCommand.
- TD-471 - mmap() called from the Normal-World Daemon with a negative offset is failing.
- TBUG-433 – Normal-World releases WSM pages too early when closing a GP-TA session.
- TBUG-436 - Kinibi Normal-World Driver 64 bit Normal-World registers corrupted after SMC call.

Keymaster

- TBUG-368 – Add support for 64 bytes input in Keymaster ECDSA/DSA sign/verify functions.

## 3.9 Kinibi v301B

### 3.9.1 New Core Features

Kinibi v301B is a maintenance release, the Kinibi API level is unchanged.

#### 3.9.1.1 Android 64

This release adds the support for Android user space applications using the AArch64 instruction set (Android64).

The 32-bit and 64-bit versions of the user space libraries are provided and Kinibi can support both types of Client Applications.

#### 3.9.1.2 Android 4.4 Kitkat Keymaster

A new function TEE_GetKeyInfo() is exported by the Normal-World library for Keymaster.

This function can be used to retrieve the key type, length and meta data from a key blob.

#### 3.9.1.3 Endorsement

The endorsement feature has been updated in order to prevent an explicit reboot of the device in some situation.

The endorsement feature could not be used if the Kinibi daemon was started without the authentication token and root container.

#### 3.9.1.4 DrApi

The DrApi allows the mapping of non-secure buffers.

A new attribute MAP_NOT_SECURE can be used when calling drApiMapPhys64() in order to map a buffer with the MMU NS bit set to 1.

A new function tlApi_callDriverEx() has been added for the Trusted Application to Driver communication. All the Kinibi drivers have been updated accordingly. Drivers can use drApiExtractMsgLen() and drApiExtractMsgCmd() helper functions to retrieve the length of the payload and the command sent by the Trusted Application.

### 3.9.2 Fixed Issues

Core

- TBUG-280 The FIQ interrupting the Secure World were not propagated properly
- TBUG-281 An invalid configuration of the GIC prevented to handle secure interrupts on ARMv8 platforms.
- TBUG-303 Fixed a potential crash of a task due to an incomplete context switch. (only when LPAE is used in the Secure World).
- TBUG-315 SSIQ boot parameter was not propagated in Kinibi (ARMv8)
- TBUG-330 Aligned all the stacks to 8-byte. This could potentially cause a segmentation fault during the startup of Kinibi.
- TBUG-32 drApiGetPhysMemType() returned an incorrect memory type. It is now returning either DRAPI_PHYS_MEM_TYPE_NON_SECURE or DRAPI_PHYS_MEM_TYPE_SECURE. DRAPI_PHYS_MEM_TYPE_HIGH_SECURE is no longer returned.
- Zendesk ticket #258 In some circumstances, large physical addresses could be truncated because of an incorrect macro.

- TBUG-321 The Core migration was not properly saving the TTBR0 in the case of LPAE.
- TBUG-352 - LPAE alignment check on TTBR1 L1 table is incorrect.
- TBUG-345 FIQ did not work with other CPUs than CPU0.

Crypto

- More checks on the RSA key used for the signature or verification have been added.
- TBUG-300 More checks on the input parameters have been added to the digest functions.

Daemon

- TBUG-257 - Memory leak in case of error during the initialization of the daemon.
- TBUG-260, TBUG-265 Fixed several minor defects reported by the Coverity tool.
- TBUG-320 The daemon was not closing pending sessions when restarting.
- TBUG-301 Memory leak if mmap() failed.
- TBUG-325 getdtablesize() function used by mcDriverDaemon is no more present in Android L. daemon() is used instead.

Kernel API

- TBUG-258 Memory leak in case of error during the initialization of the Kernel API
- mc_driver_ctrl() and mc_manage() have been removed from the API since they were unused.
- Fixed several minor defects reported by the Coverity tool.

GP

- TBUG-282 the "/data/app/mcRegistry/TbStorage" folder was created with an incorrect attribute (600 instead of 700).
- TBUG-214 TEE_GenerateKey() was making the wrong assumption on where to find the TEE_ATTR_RSA_PUBLIC_EXPONENT.

Client Library

- TBUG-261 Fixed several minor defects reported by the Coverity tool.

KeyMaster

- The declaration of TEE_ECDSAVerify() was missing in the previous release and has been added.
- TBUG-279 tci.h has the proper copyright header.
- TD-389 Several issues for ECDSA have been fixed in TEE_GetPubKey().
- Unused data in teeRsaPrivKeyMeta_t and teeRsaCrtPrivKeyMeta_t structures has been removed.

Kernel Driver

- TBUG-302 fcheck_files() can return a NULL pointer.
- Fixed several minor defects reported by the Coverity tool.
- TBUG-333 - Driver: IRQ should be freed on init error even if PM is disabled.
- TBUG-327 changed version.h from GPL to BSD

Trusted User Interface

- TBUG-285 the image could not be displayed correctly..

- TBUG-295 Handle the case when the activity is already closed (activity destroyed), for instance when the home/back key has been pressed and we try to close the TUI session.
- TBUG-247 added a sanity check to prevent the use of PA bigger than 32 bits.
- Fixed several minor defects in DrTui reported by the Coverity tool.
- TBUG-346 PNG transparency rendering issue.
- TBUG-358 Implemented a fast bitblt supporting alpha blending.

TlSdk

- All the Client Applications samples have been updated to be position independent executables in order to be compatible with Android 4.4.

  APP_PIE := true has been added to Application.mk.

- TBUG-335 SDK allowed GPTAs to not specify a UUID key
- TBUG-359 MobiConvert was not correctly counting the total code size in some situations when the TA had several code sections with different alignments.

DrSdk

- TBUG-136 In Driver/TA communication, retVal could overlap with the payload.

RootPA and CMTL

- Fixed 4 Flawfinder lvl4 issues fixed

# 3.10 Kinibi v301

## 3.10.1 New Core Features

Kinibi v301 introduces some new functionality and APIs.

The new Kinibi API level for Kinibi 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.

### 3.10.1.1 Cryptographic Functionality Update

**Kinibi 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 Kinibi 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.

### 3.10.1.2 Android 4.4 Kitkat Keymaster

The Kinibi 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 Kinibi integrator to include this TA and shared library in the device software image.

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

The Kinibi 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 Kinibi integration guide.

#### 3.10.1.3 Endorsement

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

#### 3.10.1.4 Trusted User Interface Enhancements

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

### 3.10.2 New Integration Features

#### 3.10.2.1 Support of ARMv8 Application Processors

Kinibi is compatible with the new ARMv8 architecture and now supports Cortex-A53 and Cortex-A57-based application processors. Kinibi 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 only normal world components in 32-bit mode.

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

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

#### 3.10.2.2 Memory Extension for Secure Drivers

The virtual space accessible to Secure Drivers has been increased to 60MB.

Drivers can map large sections of memory from D9 (0x800000) to D60 (0x3B00000).

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

#### 3.10.2.3 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.

### 3.10.3 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

## 3.11 Kinibi v300

### 3.11.1 New Core Features

Kinibi v300 introduces the following new features and improvements:

#### Trusted User Interface (TUI)

The Kinibi 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

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

#### DRM API

The Kinibi 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

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

#### GlobalPlatform API

Kinibi 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 "Kinibi – 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

Kinibi 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 Kinibi Developer's Guide, Kinibi API Documentation and Kinibi Integration Guide.

#### Backward Compatibility

Kinibi v300 provides backward compatibility with the previous APIs and binary compatibility for Trusted Applications and Secure Drivers.

### 3.11.2 New Integration Features

#### big.LITTLE MP compliance

Kinibi is compliant with the big.LITTLE MP model. Kinibi 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

Kinibi 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.

### 3.11.3 Fixed Issues

Kinibi 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

Kinibi v300-V004 fixes the following issues:

- It is no longer necessary to set the endorsement key when MobiConfig is used to configure the Kinibi 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 Kinibi v300-V001

Kinibi 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

Kinibi 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

Kinibi 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

## 3.12 Kinibi v202

### 3.12.1 New Core Features

#### Client Kernel API

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

### 3.12.2 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 Kinibi

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.

### 3.12.3 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 Kinibi 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.

## 3.13 Kinibi v201

### 3.13.1 New Core Features

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

### 3.13.2 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 – Kinibi fails with high session limits

MCTWO-1896 – Remove unintended heap use from Kinibi 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

# 4 What's new in Kinibi for EXYNOS64

The following lists the deltas of all the Early Access, Feature Complete and Commercial Releases performed to Samsung LSI.

## 4.1 Trustonic Kinibi-400A-EXYNOS64-v012 (build 78462)

This version of Trustonic TEE is a Commercial Release update of Kinibi-400.

SWd Changes:

- TBUG-1351, fixing Kinibi issue in blacklisting of 64bit address.

DrTUI:

- TBUG-1263 fix for early platform hang linked to DrTui and MSG_CLOSE_TRUSTLET (already delivered by email on 2017-11-20)

## 4.2 Trustonic Kinibi-400A-EXYNOS64-v012

**Release status: Commercial Release**

This version of Trustonic TEE is a Commercial Release update of Kinibi-400.

SWd Changes:

- TBUG-1158 Miscalculation of parameter in DES key generation
- blacklist feature manageable by SMC from all cores
- Support for new MPIDR formats.

It also contains DrTui fixes:

- TBUG-1138 NULL pointer dereference in the drTuiCore driver
- TBUG-1138 Fix touch thread queue initialization

It also integrates a list of NWd Security fixes:

- TD-1079 NWd driver: only log CPU status when actually switching
- TBUG-1137 NWd driver: protect driver against TA size overflows
- TBUG-1143 NWd driver: lock around file->private_data
- TBUG-1159 NWd driver: do not blindly reset client on close in clientlib

It assures Android O preview 2 compatibility:

- Support for all Linux kernels up to v4.9
- NWd client: use ANDROID rather than LOG_ANDROID in log.h, to solve Android O preview 2 issue

This version of the product has been fully validated and no issue found. As a commercial Release quality, it is approved for production.

## 4.3 Trustonic Kinibi-400A-EXYNOS64-v011

**Release status: Commercial Release**

This version of Trustonic TEE is a Commercial Release update of Kinibi-400.

SWd Changes:

- Integer overflow check added in RTM at TA loading time (FIX TBUG-1078).
- Page pool maximum increased to go from 32MB to 128MB (TD-1011).

Overall TEE package has also been updated with minor fixes in SDK samples:

- A new SamplePinPad, showing TUI usage in GP Trusted App.
- ECDSA sample disabled for old TEE version.

This version of the product has been fully validated and no issue found. As a commercial Release quality, it is approved for production.

## 4.4 Trustonic Kinibi-400A-EXYNOS64-v010

**Release status: Commercial Release**

This version of Trustonic TEE is a Commercial Release update of Kinibi-400.

Changes of this delivery affect only the Secure World image:

- Reject TA if it uses the first page of the virtual address space (TBUG-1085).
- Forbid loading a non-downgrade protected system TA with version >=0.1 when the UUID.version file exist (REV-1792).
- Increasing number of SWd L2 descriptors from 65 to 110 (TBUG_1087).

This version of the product has been fully validated and no issue found. As a commercial Release quality, it is approved for production.

## 4.5 Trustonic Kinibi-400A-EXYNOS64-v009

**Release status: Commercial Release**

This version of Trustonic TEE is a Commercial Release update of Kinibi-400.

Changes in Secure World image and in Normal World:

- Removed the automatic conversion tool that allowed silicon vendors and device manufacturers to upgrade existing devices from Kinibi-302A (TD-989, SWd image and mcDriverDaemon).
- Coverity issues fixed (TBUG-1071, NWd components)

This version of the product has been fully validated and no issue found. As a commercial Release quality, it is approved for production.

## 4.6 Trustonic Kinibi-400A-EXYNOS64-v008

**Release status: Commercial Release**

This version of Trustonic TEE is a Commercial Release update of Kinibi-400.

Changes in Secure World image and in Normal World:

- Restoring "Contiguous world shared memory buffer" feature that was removed in V006.

This version of the product has been fully validated and no issue found. As a commercial Release quality, it is approved for production.

## 4.7 Trustonic Kinibi-400A-EXYNOS64-v007

**Release status: Commercial Release**

This version of Trustonic TEE is a Commercial Release update of Kinibi-400.

Changes in Secure World image:

- Possible integer underflow in IWS buffer parsing in RTM (TBUG-1052).

This version of the product has been fully validated and no issue found. As a commercial Release quality, it is approved for production.

## 4.8 Trustonic Kinibi-400A-EXYNOS64-v006

**Release status: Commercial Release**

This version of Trustonic TEE is the Commercial Release of Kinibi-400.

This version is introducing a new feature for the System TA Antirollback protection, the TA version check can be disabled through efuse burning (EL3 call/ PLAT_TBASE_INPUT_SYS_TA_RP_ENABLED)

It's also fixing few issues:

Changes on the Normal World components:

- Mobicore driver: dead code removed and ARMv8/GP issue fix for input parameters in shared buffed (file mmu.c, TBUG-1045)
- mcDaemon: Fix start issue in case of option `-r` with persistent SWd driver (TBUG-1039)
- t-base-dev-kit/Sample GP: fixing issues to have a correct GP example.

Changes in Secure World image:

- CBC IV generation is not uniformly random in SFS implementation (TBUG-984).
- RTM crash via un-sanitized signature data from the NWd (TBUG-906).
- RTM memory allocator does not unmap memory in case of insufficient RTM heap (TBUG-1040)

This version of the product has been fully validated and no issue found. As a commercial Release quality, it is approved for production.

## 4.9 Trustonic Kinibi-400A-EXYNOS64-v005

**Release status: Commercial Release**

This version of Trustonic TEE is the Commercial Release of Kinibi-400. From the previous delivery, no new feature introduced. It's mainly fixing issues to improve overall TEE stability and Global Platform compliance:

Changes on the Normal World components:

- Mobicore driver: Update of low level SWd communication protocol to improve GP Time management (TBUG-970).
- Client Lib: Several minor updates to fix remaining issues in GP Compliance.
- Client Lib / Proxy: Shared Memory management refactored, fix for memory leak in proxy client and improve support of different SDK versions.
- tlcTui: Report code update to dynamically provide screen resolution to the NWd (Not applicable on Exynos platform).

Changes in Secure World image:

- Update of low level SWd communication protocol to improve GP Time management (TBUG-970).
- Fix for RTM threads synchronization issue on L2 table allocation (TBUG-957).

- Fix for RTM invalid check and free of L2 tables in case of overlapping buffer (TBUG-1031).
- Traces Cleanup and add trace for RPMB upgrade (TBUG-1016).
- Memory leak fix in some TA loading error cases (TBUG-967).

This version of the product has been fully validated and no issue found. As a commercial Release quality, it is approved for production.

## 4.10 Trustonic Kinibi-400A-EXYNOS64-v004

**Release status: Feature Complete**

This version of Trustonic TEE is a Feature Complete release of Kinibi-400. This version is fixing the following issue:

- Invalid parameters returned by API drApiFastCall (SWd binary).

## 4.11 Trustonic Kinibi-400A-EXYNOS64-v003

**Release status: Feature Complete**

This version of Trustonic TEE is the 2nd Feature Complete release of Kinibi-400. This version is mainly fixing a lot of issues and improving the overall TEE stability:

- SWd API DrAPI_GetSecureTimeStamp was returning a random value and, by consequence, was also breaking all the SWd timeout mechanism (all the timeouts were mostly considered as "infinite")
- Several memory leaks fixed in the SWd, leading to OUT_OF_MEMORY TEE error in case of intensive usage of GP APIs.
- Many changes in TEE Linux driver to fix/adapt behaviour to pass the GP Compliance.
- TTS component update to support kinibi-400A.

## 4.12 Trustonic Kinibi-400A-EXYNOS64-v002

**Release status: Feature Complete**

This version of Trustonic TEE is the Feature Complete release of Kinibi-400. It contains all the features and fixes described in generic Kinibi 400A version.

- Documentation has been updated, but not yet finalized.
- Product has been fully validated, no major regression found however several tests are still not passing.
- This version is also not yet passing the GP Compliance testing suite.

## 4.13 Trustonic Kinibi-400A-EXYNOS64-v001

**Release status: Early Access**

This version of Trustonic TEE is the first Early access version of Kinibgi-400.

It introduces the support Rollback Protection for System TA.

Main features not included:

- Documentation is not up to date and still aligned on Kinibi-311B.
- TA to TA communication is not present.
- GP properties are not yet supported.

It has not been fully validated and is not ready for production

‹

# 5 HARDWARE AND SOFTWARE TESTED

The hardware platform tested is:

- Joshua (Exynos 7870).
- Lauterbach t32 is optional

The software components used are listed in the table below:

| SW Components & Tools | Source | Version |
|---|---|---|
| Operating System for development PC | | Windows XP SP3 32bits, Ubuntu 12.04 64bit |
| Android NDK | Google | r7b |
| ARM RVCT | ARM | 4.1 |
| Joshua BSP | S.LSI BSP shared in SecuTrans | 20151001_JoshuaSMDK |
| EL3/Trusted Firmware | S.LSI BSP shared in Secu Trans with Trustonic TEE SPD | Delivered in Trustonic TEE package |

# 6 KNOWN ISSUES AND LIMITATIONS

Due to Wi-Fi availability issue on Exynos SMDK devices, few OTA test cases have not been passed.

This is mandatory that the OEM passes the complete Trustonic TEE test suite including the OTA tests.

Trusted UI tests are also impossible to pass as feature is today not implemented/available development platform.

# 7 TEST RESULTS