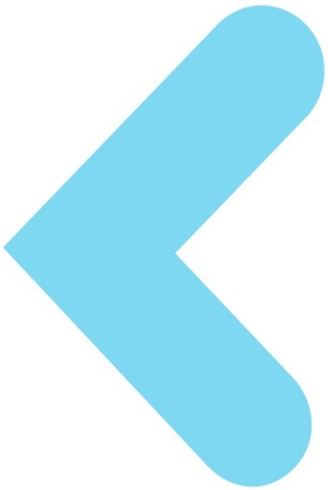
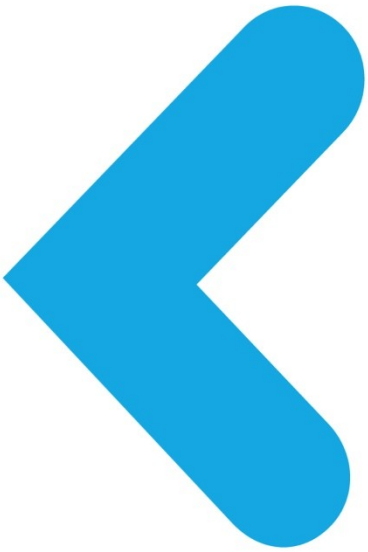
TRUSTONIC

Release Notes



Kinibi v500a Release Notes



PREFACE

TABLE OF CONTENTS

1 INTRODUCTION

This document is the Release Notes for the Trustonic Kinibi TEE product on Samsung S.LSI ARMv8 platforms with a Normal World in Aarch32 or Aarch64.

Secure World TEE is running itself in **Aarch32**.

The exact version of the product is:
t-base-Exynos-Android-500a-V002-20191027_223905_74445_100603.

This version is a Commercial Release. It approved for production.

2 What's New in Kinibi

2.1 Kinibi v500a

Kinibi-500a mainly introduces big system changes with support of SMP (parallel TAs execution and multi-threaded SWd Drivers) and AArch64.

The TEE API level is not changed.

1.1.1 New Core Features

2.1.1.1 AArch64

Kinibi is now built completely for AArch64 (TEE Kernel, RTM and internal crypto library). The SDK builds TAs and Drivers for AArch64 by default. It still supports AArch32 so Kinibi also includes McLib library to run 32-bit TAs.

When porting existing Secure Drivers to Kinibi-500a, integrators have to take precautions for the marshalling structures used for the TA-to-Driver communication and to change the architecture dependent types to the largest type that can store the value.

2.1.1.2 SMP – Symmetric Multi-Processing

Kinibi-500a supports parallel execution of multiple TAs or SWd Driver on a multicore SoC. The TAs have to be single-threaded and by default multi-threaded tasks like Secure Drivers have their threads serialized. However, synchronization issues should be anticipated.

This new TEE version is also offering the early support for distributed SWd Drivers, allowing multi-threading and parallel execution of the threads on several SoC. It has to be specifically enabled into SWd Driver makefile (several restrictions to take care: TEE API not yet thread safe…)

2.1.1.2.1 SMP – Mobiload impact

Due to new TEE SMP architecture, the low-level SMC protocol had to be updated. Impact is transparent into the Rich OS thanks to the TEE Linux driver abstracting the changes. However, for customers who were relying on Kinibi / Mobiload feature, the source code must be updated in order to communicate to Kinibi-500. Contact Trustonic for details and updated reference source code

2.1.1.3 Overall TEE System performances

Above TEE architecture evolution have strong impact on overall System performances. The SMP is improving the TEE reactivity and load distribution, decreasing latency in command execution. In the same way, relying on Aarch64 is bringing 10% to 20% performance increase, especially visible in case of cryptography (using Aarch64 TEE crypto stack).

1.1.2 NWd Components re-organized around Android System/Vendor

With the removal of Trustonic OTAv1 support all user space Kinibi components have been re-organized to reduce complexity and ease integration with clear Android System and Android Vendor split.

1.1.3 New SDK Features

2.1.1.4 Default TA toolchain updated to GCC 8.3.3

2.1.1.5 Extending GP property list for NWd Hypervisor support

In case of NWd environment relying on Hypervisor and different Virtual Machines, adding new GP Property to allow VM identification from TA ("*com.trustonic.client.virtualMachine*").

2.1.1.6 Factory Reset Persistent GP Secure Storage

System TA are now be able to directly select a factory reset persistent GP Secure Storage ID, TEE_TT_STORAGE_PROTECTED, from the GP SFS with the API TEE_CreatePersistentObject(). Data will then be physically stored into an SFS partition written into RPMB (if configured, else into another dedicated Rich OS based file called store1.tf).

TEE Implementation Property gpd.tee.trustedStorage.antiRollback.protectionLevel set to 1000 used to identify final physical storage (RPMB or Rich OS).

2.1.1.7 New Crypto Algorithm

To support Use Cases like TEE HSM and SWd TLS Stack, add support for TEE_TT_ALG_ECDSA_VARIABLE_LENGTH and TEE_TT_ALG_RSASSA_PKCS1_V1_5_VARIABLE_LENGTH

1.1.4 Removed Features

1.1.4.1 Removed support for ARM RVCT compiler 5

The support for ARM RVCT compiler version 5 has been removed.

2.1.1.8 Fastcall Hooks APIs disabled

Feature has always generated complex issue and was more and more seen as security risk due to lack of isolation and running out of any TEE context (all TEE OS security mechanisms like canaries, stack guard, isolations... not existing in Fastcall hook context)

Equivalent feature possible through SWd driver API drApiCallTrustedFirmware().

2.1.1.9 Obsolete APIs removed

- Very old API levels, lower than 5, for TA and SWd Driver removed from TEE.

  Most of these were obsolete APIs from several years and seen for some of them as security risk due to legacy limitations (static memory layout typically, preventing ASLR).
- Legacy TA/SWd Driver API tlApi_callDriver() removed

Obsolete API from several years, replaced by improved version tlApi_callDriverEx()

- Loading of TAs or SWd Drivers with an API level lower than 5 will be now rejected by TEE.
- Legacy macros getstacksize() and GET_STACK_SIZE() from SDK/DDK have been removed (TBUG-1311). Impact on associated local SWd Driver macros fillStack() and clearStack(). For security purpose it is now recommended to use API drApiStackAlloc(), which one ensure dynamic (random) stack allocation, initialization and MMU Page Guards.

2.1.1.10 Trustonic OTAv1 support stopped

With stop of Trustonic OTAv1, very strong cleanup on now several useless NWd components (RootPA, libCurl, mcRegistry removed)

This cleanup has also a big source organization impact and allow a strong TEE Package NWd simplification (less components, re-organized around Android HIDL System/Vendor concepts)

1.1.5 Bug Fixes

TBUG-1435 fix for TEE_AllocateOperation in SWd driver failing with out of memory

TBUG-1451 RTM may not handle MEMREF_WHOLE offsets correctly (last page missing)

TBUG-1475 GP singleInstance multiSession TA had an issue into pSessionContext management

TBUG-1480 Document output length limit for tlApiDeriveKey and return better error code.

TBUG-1495 Allow desCipherDoFinal with no output buffer.

TBUG-1501 Fix state machine to enable TEE_AEInit after TEE_ResetOperation

TBUG-1515 Fix in Aarch64 version of TEE Kernel to prevent NWd NEON registers corruption

TBUG-1519 Timing leak in key RSA generation

TBUG-1532 Reserve enough workspace for all RSA private-key operations on Aarch64 version

TBUG-1550 Adding SPDX Headers into our TEE Linux Driver, mandatory for Linux Kernel above 4.17.

TBUG-1554 TBUG-1555 TBUG-1557 Several fixes to correctly manage in SWd ION Cached and Uncached buffers.

TBUG-1579 / TD-588 1MB size restriction on TA stored into SFS removed.

TBUG-1580 Fix in Gicv2/v3 dynamic detection and related vector entries

TBUG-1588 TEEC_MEMREF_PARTIAL_INPUT does not support entire parent shared memory sharing

TBUG-1640 / TBUG-1641 NWd time was not updated correctly and atomically in MCP protocol

TBUG-1651 Ensure SFS compatibility with Kinibi 400x releases by reverting behavior in tlApiDeriveKey() API for drivers.

3 PAST KINIBI RELEASES

3.1 Kinibi v410a

410a introduces new features to improve security, performance, stability and flexibility.

Kinibi-410a supports Android P.

The TEE API level is not changed.

1.1.6 Android P changes

It assures Android P official release compatibility:

- Updating RootPA and TUI components to match Android P/Treble isolation.
- Development of new system services to enable TEE access for OTA and System APKs.
- Update reference SE Linux rules to follow Android evolution.

1.1.7 New SDK features

The SDK is modernized and activates security by default.

1.1.7.1 GCC by default and update to version 7.1.1

The SDK now uses the GNU toolchain by default (setup.sh).

The SDK is now based on the Linaro GCC 7.1.1 version of the compiler for compiling TAs and Drivers. The compiler version GCC 4.8.4 is no longer supported.

The support for ARM RVCT compiler is now deprecated. The SDK still supports RVCT, but to use ASLR, the developer must use GCC.

1.1.7.2 Latest API level by default

By default, the SDK now selects the **latest** `TBASE_API_LEVEL` to make sure that TAs are built with the highest security options. It includes the separate heap, guard pages for stack and PIE.

The TA developer can override the TBASE_API_LEVEL and TOOLCHAIN in the TA's makefile.mk.

1.1.7.3 PIE for Trusted Applications

To use the ASLR for the TA code, the SDK compiles TAs and Drivers as Position Independent Executables using the `-fpie` option. Kinibi-410a understands a new TA format with relocation information and loads the TA at a random address on each load of the TA.

PIE is activated by default in the SDK in setup.sh, using `TA_PIE=PIE`. The SDK sets TA_PIE to `NON_PIE`, when TOOLCHAIN is ARM or when TBASE_API_LEVEL < 5. Note that the SDK comes with two versions of the libraries, one for PIE and one for NON_PIE in respective sub-folders.

1.1.7.4 Stack Smashing Protection

The SDK now enables the Stack Smashing Protection of the compiler by default. The previous configuration options have been removed. The protection is now enabled in Trustonic TAs, drivers and the TEE.

1.1.7.5 Warnings as Errors

The SDK now enables all warnings and elevates warnings to errors. This helps in increasing the code quality and to avoid bugs in the code. GCC is called with `-Wall -Wextra -Werror`.

Also, the SDK detects format problems in `tlApiPrintf` and similar functions.

1.1.7.6 Support for 64-bit printf.

The `tlApiPrintf` and similar functions now support `%llx` to print a 64-bit value.

1.1.7.7 Encrypting System TAs and Drivers

MobiConvert can now encrypt System TAs and Drivers using a 128-bit AES GCM scheme. The TA developer can specify an encryption key in the TA's makefile.mk as

```
ENCRYPT_SERVICE_KEY := Locals/Build/ServiceEncryptionKey128.xml
```

1.1.7.8 MobiConvert TA re-signing

The MobiConvert version V1.6 in Kinibi-410a allows more flexible combinations of existing options and has improved help and feedback on the command-line.

Mobiconvert now supports the --sign mode with an XML keyfile. To activate the former --signraw mode with HSM key retrieval, the -hsm option needs to be added. Sign using key file:

```
java -jar MobiConvert.jar -sign -in <UUID.tabin.raw>
 --output <UUID.tabin> --keyfile Locals/Build/pairVendorTltSig.pem
```

In this same --sign mode, MobiConvert also allows re-signing an already signed tlbin. To select re-sign mode, the extension of input file should be .tlbin, .tabin or .drbin.

The second change is in the main conversion mode, activated by --servicetype. This mode traditionally requires as input an ELF file and many command-line parameters to define the TA identity. Now MobiConvert V1.6 accepts also an already converted tlbin, tabin or drbin as input file, in addition to the set of command-line parameters. This second re-sign mode allows to exchange the TA identity in the header, remove the existing signature and sign with the provided key.

For more information, see the updated MobiConvert Manual in the Kinibi Developer's Guide.

1.1.8 New Core Features

1.1.8.1 ASLR – Address space layout randomization

Kinibi-410a includes ASLR feature for TA code, data, stack, heap, WSM and McLib. On each open-session of a TA or Driver, the TEE generates a new randomized address space layout.

Note: This feature is currently not available for the Fastcall Hook drivers (in consequence, to be able to install the Hook, it must be disabled in SWd driver makefile).

1.1.8.2 TA downgrade protection

Kinibi-410a extended support of downgrade protection to cover Legacy System TAs along with TAs that use the GlobalPlatform API. See the Kinibi Integration Guide for more information.

1.1.8.3 Cryptography improvements

410a includes various refactoring and performance and concurrency improvements:

- Alignment of crypto code with the FIPS-certified library
- Unification of the crypto code across all the APIs
- Removed support for RSA ISO 9796 padding
- `TEE_MemCompare` is constant-time

Note: the function TEE_MemCompare was previously based on compiler built-in memcmp that is highly optimized. Now the TEE_MemCompare is significantly slower.

1.1.8.4 Stability improvements

410a includes various refactoring and performance and concurrency improvements:

- MTK runs with IRQs disabled.
- TA loading is less affected by crypto driver usage.
- TA closing only interacts with drivers that the TA has been using.
- Secure Filesystem L2 cache optimized.
- Normal world Registry Update has been removed
- Privileges of secure drivers have been reduced
- Automatic core switch balancing following Linux kernel scheduler

3.1.1.1 90 Trusted Applications, 30 Secure Drivers

The total number of applications which can be running simultaneously has been increased from 64 to 128. From this number, 8 slots are assigned to internal use cases. For the rest:

- 60 slots for System Trusted Applications
- 30 slots for Secure Drivers
- 30 slots for Service Provider Trusted Applications

1.1.8.5 World-shared memory (WSM)

410a allows TA Connectors to share WSM with more than 1MB of size. The WSM maximum size is limited by the TA’s virtual address space which is 124 MB.

1.1.8.6 ION buffer support (WSM)

410a includes support for ION buffer file handles in the `TEEC_RegisterSharedMemory` function.

1.1.9 New Integration Features

1.1.9.1 New Document to guide in System Debugging

The new document `Kinibi_System_Debug_Guide.pdf` provides a practical guide on TEE debugging including the various phases: TA development, TEE system integration and post-mortem analysis. The various debug options, tools and expected traces are explained in detail.

1.1.9.2 Extended Debug support

To give a deeper understanding of events that can lead to a crash, the tee-ps tool was extended to include information about the last 10 crashes, including ThreadID, PC and SP, as well as ASLR offset code and ASLR offset McLib.

To overcome debugging challenges introduced by ASLR, Kinibi-410a contains ASLR-specific extensions to the existing debug channels. When a debuggable TA is loaded, Kinibi Release version prints the ASLR offsets to the traces. When a TA crashes, Kinibi outputs the ASLR offsets to the Kinibi crash dump. That way the PC can be traced back to the place in the lst2 file.

To help evaluate Kinibi scheduling and CPU time utilization, the tee-ps tool was extended to show TEE entry and exit time-stamps as well as entry and exit reasons (SMC, IRQ, FIQ) and CPU id.

To help debugging on a multi-core system, in Kinibi-410a, the traces are extended to include the CPU id on which the trace was generated. This id is prepended to each line printed.

1.1.9.3 Experimental AArch64 support

The 410a product package contains a new Kinibi version where the Kinibi kernel uses AArch64 instruction set. All existing TAs and Drivers using ARMv7 or AArch32 instruction-set continue to run on this Kinibi version. However, certain features may not work, like the fastcall hook.

Note: A new version of ATF SPD is required to start the AArch64 Kinibi.

1.1.9.4 Mitigation for Variant 1, 2 and 3 – Spectre and Meltdown

Kinibi 410a has been checked with the Coverity tool for Variant 1 issues and respective findings have been reworked.

Kinibi 410a includes a new vector table that includes the BP invalidate instruction before any branch, as mitigation against Variant 2.

Kinibi 410a includes a new kernel memory layout to implement KPTI – kernel page table isolation, as mitigation against Variant 3.

The mitigations for Variant 2 and Variant 3 are activated by default.

Note: a new version of ATF SPD is required for Kinibi 410a, because of the Variant 2 fix.

1.1.9.5 Memory Blacklist

Kinibi 410a has a new security feature that allows Hypervisor in EL2 and Monitor in EL3 to configure blacklisted memory areas that are inaccessible by Secure Drivers in Kinibi.

1.1.9.6 Embedding TAs

Kinibi 410a allows embedding TAs and Drivers into the Kinibi image using the image builder python script. Such TAs can be started at boot, right after Kinibi boot initialization. See the Kinibi integration guide for the details.

1.1.9.7 Injecting and loading decryption key

Kinibi 410a supports System TA encryption. The integrator must do three things: 1) encrypt its TAs and Drivers, 2) in the factory, inject the decryption key into the device using the TAKeyInjectionTool, and 3) on startup of the mcDriverDaemon, pass it the wrapped decryption key for loading into the TEE. The Kinibi integration guide contains the details.

1.1.9.8 Loading Secure Drivers

410a extended the -r option of the Daemon to allow loading secure drivers by passing only their UUID if they’re present in the registry or embedded in the TEE image.

1.1.9.9 ARMv7: Porting kit switches to GCC 7.1.1

The 410a Kinibi porting kit now works only with GCC 7.1.1. Support for ARM RVCT has been dropped. Also the whole of Kinibi is compiled with GCC 7.1.1.

1.1.10 Removed Features

3.1.1.2 Removed support for Fastcall Identity Mappings

A sub-feature of the fastcall driver support, the identity mapping has been removed from Kinibi-410a. The `.prepareIdenticalMapping` function of the `fcContext_t` structure is set to NULL and calling it will create an instruction fetch abort.

3.1.1.3 Removed support for IMEM

Kinibi-410 does not support IMEM or internal RAM. Respective memory regions passed as parameters in boot arguments are ignored. Trying to load a TA from IMEM will fail.

3.1.1.4 Deprecation of DrApi for legacy memory layout

In Kinibi-410a the memory mapping API for drivers using the legacy memory layout is marked as deprecated using the compiler attribute. With warnings-as-errors this will stop the compilation and the developer must convert the driver or add a line to the Driver makefile to ignore this warning. The legacy memory layout for Drivers will be removed from the next Kinibi product.

1.1.10.1 Android Keymaster and Gatekeeper as stand-alone package

The 410a Kinibi software package no longer includes the Android Keymaster application.

The Android Keymaster 1 and Android Keymaster 3 applications now support multiple Kinibi versions and are delivered in stand-alone packages.

1.1.11 Fixed Issues

Core

- TBUG-886 TEE 32-bit OS should use SMC32 convention instead of SMC64

Note: The SMC id definitions that exist in MobiLoad/SPD(ATF) are also updated to SMC32 convention, so it is mandatory to update the SMC id in MobiLoad/SPD code during the integration.

- TBUG-1161 SWd should reject unknown NWd page table entries instead of blindly accepting them
- TBUG-1293 TA exception when stack size is not defined
- TBUG-1336 TEEC_MEMREF_WHOLE memory type loses READ attribute when parent shared mem defines direction flag as input
- TBUG-1346 mcDaemon -r option allowing Linux kernel crash
- TBUG-1351 Kinibi-310/400 not supporting blacklisting of 64bit address
- TBUG-1353 PLAT_TRACE_EVENT broken : crash in Kinibi Debug
- TBUG-1364 Missing libpng symbols in drTuiCore from k410A release package
- TBUG-1374 Lock missing in client_gp_{register,release}_shared_mem
- TBUG-1387 Xen bus sending multiple time same state change notifications
- TBUG-1392 MobiConvert from 410A not retro compatible (only allow 1 bss/data section)

3.2 Kinibi v400C

1.1.12 New Integration Features

400C is the update of Kinibi for Android Pie. The TEE API level is unchanged.

1.1.12.1 Android P changes

It assures Android P official release compatibility:

- Updating RootPA and TUI components to match Android P/Treble isolation.
- Development of new system services to enable TEE access for OTA and System APKs.
- Update reference SE Linux rules to follow Android evolution.

3.3 Kinibi 401a

3.3.1 New Features

This release is an evolution to the 400A release with changes to Normal world only.

3.3.1.1 XEN hypervisor support

401a adds support for Xen-based virtualization. The TEE works with client applications in Dom0 and in DomU.

3.3.1.2 GNU/Linux support

In addition to supporting Android, 401a adds support for classic Linux distributions in a 32-bit or 64-bit OS. Currently the Yocto BSP is explicitly supported.

3.3.1.3 4096-bit support for OEM keys

For signing drivers and system TAs, 401a support for 3072-bit and 4096-bit RSA keys.

3.4 Kinibi v400b

1.1.13 New Integration Features

400b is the update of Kinibi for Android Oreo. The TEE API level is unchanged.

1.1.13.1 Android O changes

It assures Android O official release compatibility:

- Updating RootPA and TUI components to match Android O/Treble isolation.
- TBUG-1190, libMcClient, Android protobuf version conflicts for Android O (removing native TEE proxy, as feature not used and no more needed).
- 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.
- Update reference SE Linux rules to follow Android evolution.

1.1.13.2 Stack Smashing Protection

t-base-dev-kit which is comprised of SDK and DDK was updated to include the Stack Smashing Protection feature for TAs and DRVs. It is an optional feature that could be enabled by using the macro DECLARE_STACK_PROTECTOR and the compile time flag ENABLE_STACK_PROTECTION set to true as explained in the Kinibi_Developers_Guide.pdf (*§3.1.3 Trusted Application Address Space*).

Samples such as TASampleRot13 and DrSampleRot13 contain a demonstration of the Stack Smashing Protection activation.

1.1.13.3 RPMB Integration

The RPMB subsystem was refactored along with the Secure File System to store the whole TA versioning partition in the RPMB partition.

The Secure RPMB driver is now embedded in Secure System image.

1.1.13.4 Secure Drivers

Secure Drivers have the possibility to call GP APIs.

In addition, the API tlApi_callDriverEx was made available to driver for inter drivers communications.

1.1.14 Fixed issues

Secure World:

- TBUG-1114 update legacy server state machine to avoid MSH thread being blocked.
- TBUG-1158 Miscalculation of parameter in DES key generation.
- TBUG-1165: Kinibi is now no more impacted by heavy NWd IRQ flow (previously unable to execute Trusted App in these specific cases).
- TBUG-1194: Infinite loadh loop in case of TlEntry failure for GP TAs.
- TBUG-1207 Calling a shared-memory API with zero length buffers leads to wrong physical mapping in the Kinibi.

DrTui:

- TBUG-1138 NULL pointer dereference in the drTuiCore driver.

- TBUG-1138 Fix touch thread queue initialization.
- TBUG-1157 [drTui] Memory leak in drTui.
- TBUG-1202 DrTui panic if it gets spurious notifications from the NWd.

Keymaster:

- TBUGAPP-3 TLC, TEE_Begin() failed when trying to map a ReadOnly buffer from dev_tee_keymaster.
- TBUGAPP-4 TLC and TA, KM_TAG_BLOB_USAGE_REQUIREMENTS may be set by caller.

Normal World:

- NWd daemon: fix crash on signal received.
- TBUG-1127 NWd driver: fix KAPI mc_open_trustlet to use in-kernel buffers.
- TBUG-1137 NWd driver: protect driver against TA size overflows.
- TBUG-1143 NWd driver: lock around file->private_data.
- TBUG-1156 NWd daemon: trigger an alarm before stopping services to make sure we eventually die.
- TBUG-1159 NWd driver: do not blindly reset client on close in clientlib.
- TBUG-1173 NWd driver: lock list in put functions when de-listing is done in release, to avoid unlikely race condition.
- TBUG-1180 NWd driver: prevent daemon reconnection after TEE crash.
- TBUG-1206 SWd activity prevents task freezing.
- TBUG-1212: TEE Linux driver, handle ERESTARTSYS error in wait notification from kernel API.
- TBUG-1213 The daemon is ready to receive some comands from the KernelDriver before being blocked in the SecureWorld ioctl.

3.5 Kinibi v400A

1.1.15 New Core Features

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

This Kinibi version is compliant to the GlobalPlatform API version 1.1.1 and passes the FIME GP compliance tool.

1.1.15.1 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.

1.1.15.2 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 Dynamic at build time
com.trustonic.tee.tui.available	Boolean	True if TUI is available 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.

1.1.15.3 GP Properties Enumeration

400A supports the GlobalPlatform properties enumeration API.

1.1.15.4 GP Time API

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

1.1.15.5 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.

1.1.15.6 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.

1.1.15.7 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()

1.1.15.8 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

1.1.15.9 Proxy enhancements

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

1.1.16 New Integration Features

1.1.16.1 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.

1.1.16.2 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.

1.1.16.3 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.

1.1.16.4 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.

1.1.16.5 Removed Trusted Storage Upgrade

Kinibi-400A removes the automatic conversion tool that allowed silicon vendors and device manufacturers to upgrade existing devices from Kinibi-302A.

1.1.17 New SDK Features

1.1.17.1 TA Manifest file

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

1.1.17.2 TUI double buffering for GP TAs

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

1.1.17.3 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.

1.1.17.4 TeeClient

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

1.1.17.5 Assembler support for TAs

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

1.1.17.6 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.

1.1.18 Fixed Issues

4 What’s new in Kinibi for Exynos

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

4.1 Trustonic Kinibi-500a-Exynos_v002

Release status: Commercial Release

This version of Trustonic TEE is a Commercial Release of Kinibi-500a (Aarch32 TEE version. It’s adding on top of previous version:

Changes in SWd TEE binary:

- TEE Internal DrRPMB SWd driver involved in GP Secure Storage on Exynos platform was still calling tlApi_callDriver() [TBUG-1668]
- Minor cosmetic fix in TEE Debug logs on Serial port, missing '\r' in EOL [TBUG-1654]

Changes in MobiConfig:

- Backward compatibility with Kinibi-400 on RPMB options restored [TBUG-1660]

This version is a Commercial Release. It has successfully passed all our internal testing and no issue reported. It is approved for production.

4.2 Trustonic Kinibi-500a-Exynos_v001 (build 73723)

Release status: Commercial Release

Post release update (build 73723):

- Critical SWd TEE init fix for Aarch32 only (ldr/str exclusive used before Cache enabled) [TBUG-1656]

This version is a Commercial Release. It has successfully passed all our internal testing and no issue reported. It is approved for production.

4.3 Trustonic Kinibi-500a-Exynos_v001

Release status: Commercial Release

This version of Trustonic TEE is the 1st Commercial Release of Kinibi-500a (Aarch32 TEE version). It contains all the features and fixes described in generic Kinibi 500a version.

Note on this version:

- The SWd / TEE image is specifically re-enabling the backward compatibility with Fastcall hook feature.

This version is a Commercial Release. It has successfully passed all our internal testing and no issue reported. It is approved for production.

5 HARDWARE AND SOFTWARE TESTED

The hardware platform tested is:

- Joshua (Exynos 7870), Ramen (Exynos 9610).
- Lauterbach t32 is optional

The software components used are listed in the table below:

Version
Windows XP SP3 32bits, Ubuntu 12.

04 64bit r76 8.3 20151001 JoshuaSMDK 20190418 Android

PE96100-tk.500.V002 - Delivered in Trustonic TEE package


e

6 Known Issues and Limitations

7 Test Results