TFNAB01.16
Release Notes

Date of Issue: August 08th, 2012
Project Name: TF Tegra3
Document Type: Technical Specification

Reference & Version: CP-2012-RT-13
Classification: Confidential
Number of pages: 9 (including 1 header pages)

TABLE OF CONTENTS

1 INTRODUCTION

This document contains the *Release Notes* for the Trusted Foundations on Nvidia Tegra3 platform, running Android.
This release is identified by TFNAB01.166.

2 SUPPORTED AND REFERENCE SW COMPONENTS AND TOOLS

This version has been integrated with the Android Ice-cream Sandwich Nvidia BSP (Linux kernel 2.6.39) and tested on a Tegra3 Cardhu board (with “e-fuses” not burnt).

Here are the full details of supported / reference components and tools:

2.1 TF TEGRA3 ANDROID

SW Components & Tools	Purpose	Source	Version
Android BSP and kernel 3.1.10	Android BSP running on Tegra3	Nvidia	`Android ICS for Cardhu. BSP version is 15r7`
ARM toolchain	Build of TF driver	Code Sourcery	Gingerbread/prebuilt/linux-x86/toolchain/arm-eabi-4.4.3/bin
ARM toolchain	Build of Normal World Client applications and libraries.	Code Sourcery	Gingerbread /prebuilt/linux-x86/toolchain/arm-eabi-4.4.3/bin
RVCT / RVDS	Build of Secure Services	ARM Limited	4.0 Build 650 (running on Win32)

2.2 WIN32 SIMULATOR

This version has been tested on a PC running Microsoft Windows XP Service Pack 3.

SW Components & Tools	Purpose	Source	Version
Microsoft Visual Studio®	Use of the Windows Simulator	Microsoft	2008 or 2008 Express (running on Win32)

3 NEW FEATURES AND VERSION HISTORY

3.1 NEW FEATURES IN TFNAB01.16

This version optimizes the LP2 transition. The L2 cache keeps its data across the transition, and only a minimal subset is flushed.

3.2 NEW FEATURES IN TFNAB01.15

This version adds a “tasks profiler” option to the Trusted Foundations (gives a visibility on secure task scheduling) and fixes an issue in GIC during LP transitions (some Normal World pending IRQ were reset by Secure World).

It also fixes “Linux kernel compliancy” warnings into TF Linux driver.

3.3 NEW FEATURES IN TFNAB01.14

This version fixes and improves several points:

- Performance improvement in L2 Cache Controller driver (parameter “ways” added L2CC shutdown function, if set to “03”, only flush workspace memory)
- Memory mapping used for L2 Cache Controller is now “shareable device” (was previously “strongly ordered”, access performance improved twice)
- The instruction CLREX (Clear exclusive) is called during TF initialization process and during each Normal World/Secure World context switch (including context switch done for Fast SMC).
- Regression in support of hardware floating point fixed (Abort in secure when Neon was asked by a Secure Service)

3.4 NEW FEATURES IN TFNAB01.13

Linux kernel entry point is now part of TF dynamic boot args (was previously defined at postlink).

3.5 NEW FEATURES IN TFNAB01.12

Changes done in this version of Trusted Foundations package:

- The package now contains two versions of Linux TF Driver : one for K2.6 and one for K3.0
- The Trusted Foundations binary fixes an invalid memory access in the very early boot process and improves the overall boot performances (caches and branches predictions are now activated immediately after boot).
- The “Dynamic clock gating” and the “Standby mode” of the L2 Cache Controller (PL310, Power Control register) are enabled at L2CC startup.

3.6 NEW FEATURES IN TFNAB01.11

This version of Trusted Foundations package adds the TEE Client API at kernel level and provides an update of TF Memory Profiler (available on Linux OS and includes now Secure Service stack information).

3.7 NEW FEATURES IN TFNAB01.10

This version adds in the Trusted Foundations package a PC tool who allows memory profiling of Secure Services running into the Trusted Foundations.

3.8 NEW FEATURES IN TFNAB01.09

This release fixes a bug that left the Normal World unstable when using the ‘FIQ Debugging’ feature.

3.9 NEW FEATURES IN TFNAB01.08

With this version, traces done by Secure Services are forwarded from the Secure World to the Android Kernel. Secure logs are now part of the kernel logs.

3.10 NEW FEATURES IN TFNAB01.07

This version supports a dynamic configuration of UART Secure Trace Driver: UART identifier to use is now dynamically transmitted to the Trusted Foundations inside the TF Boot args structure.

This version also extends the maximum supported size used by the Trusted Foundations workspace (maximum size statically extended to 15 MBytes).

In addition, the power management framework has been reworked. Cold boot, LP0 and LP2 are now running form firewalled SDRAM and LP1 is the only mode running from TZRAM (previously, it was also the case for LP2)

3.11 NEW FEATURES IN TFNAB01.06

This version introduces the support of boot parameters. Some properties that were previously defined statically at postlink time are now configurable through the boot parameters of the Trusted Foundations (For now, the set of parameters supported is limited to some “hardware” descriptions).

This package also includes a new version of the UART Secure Trace Driver (check UART availability).

3.12 NEW FEATURES IN TFNAB01.05

This version adds:

- An improvement on the Secure Trace Driver (based on UART, robustness improved during LP1),
- The support of Linux OS for the tf_resc and tf_postlinker tools,
- A new version of the tf_postlinker tool: It now support multi-stage postlink (i.e. Postlink step can be repeated several time to aggregate incrementally secure services).

3.13 NEW FEATURES IN TFNAB01.04

This version adds Tag and Latency properties from L2 Cache Controller into the secure world configuration file.

3.14 NEW FEATURES IN TFNAB01.03

This version adds to the Trusted Foundations on Nvidia Tegra3:

- the support of hardware acceleration for floating point number (based on Neon),
- optimizations of code in charge of power management transitions (flush of cache L2),
- the secure driver to enable traces from a secure services (based on the UART),
- the secure driver to communicate with Nvidia OTF hardware component.

3.15 NEW FEATURES IN TFNAB01.02

This version adds to the Trusted Foundations on Nvidia Tegra3 the support of the dynamic provisioning process.

It also fixes some issues in the ARM Errata configuration. With this version, the Trusted Foundations is configured to work on an CortexA9, r2p9.

3.16 NEW FEATURES IN TFNAB01.01

This is the first Trusted Foundations release on Nvidia Tegra3 with Android.

This release supports the following features:

- a secure integration of the Trusted Foundations to the Tegra3 device leveraging the TrustZone technology to isolate a Trusted Execution Environment and Secure Services from the main operating system,
- all the Trusted Foundations Developer API v3.0,
- a reference integration of the Trusted Foundations for the reference boot loader in order to protect and start the Trusted Foundations along with the Secure Boot process of the Tegra3 platform.

4 RELEASE RESTRICTIONS

4.1 TF TEGRA3 ANDROID

4.1.1 CRYPTOGRAPHIC API

In this release, the maximum size for PKCS11 Object Data (CKO_DATA) that can be stored and retrieved is limited to 4096 bytes.

4.1.2 MEMORY

The total size of the secure services plus TF Core binary is limited to 1020kB, i.e. the complete TF binary with the postlinked Secure Services.

4.1.3 TEE CLIENT API IN KERNEL

- Sources of TEE Client APi don't respect the "linux kernel coding rules".
- Exchanging shared buffers with the Secure World is not a fully supported feature in current implementation. In case of uses of TEE Client API in kernel, only buffers allocated by "kmalloc" kernel functions can be shared (i.e. sharing a buffer allocated on stack is not supported).

4.2 WIN32 SIMULATOR

4.2.1 CRYPTOGRAPHIC API

In this release, the maximum size for PKCS11 Object Data (CKO_DATA) that can be stored and retrieved is limited to 4096 bytes.

4.2.2 CUSTOM DRIVER

The simulators do not support the Custom Driver features defined in the Trusted Foundations Product Reference Manual.

5 KNOWN ISSUES

MINOR ISSUES:

[2618] POSTLINKER TOOL DOES NOT OBEY OBJECT SECTION ALIGNMENT

The postlinker tool, `tf_postlinker.exe`, does not currently link Native services in accordance with the alignment requirements specified in the ELF object file. The current alignment support in the postlinker is fixed at 8-byte alignment, which supports standard C code compiled into ARM and Thumb. Assembler source code which additionally specifies section alignment requirements higher than 8 bytes, such as the following code sample, may not function as expected: `; ALIGN=X requests 2x byte alignment, in this case 32 byte alignment AREA example_alignment_area, CODE, READONLY, ALIGN=5 CODE32 example_function ; Dummy function that does nothing BX lr END` For data the work around for this restriction is to manually copy incorrectly aligned data structures in the code in to dynamic heap memory at the correct alignment, and using an explicit copy. For code, there is no current workaround, as the software marks dynamic heap memory as „execute never" (XN) in the page tables.

[3982] POSTLINKER TOOL DOES NOT ERROR IF STACK SIZE VIOLATES HEAP SIZE

The postlinker tool, `tf_postlinker.exe`, does not currently produce an error if the stack size set for a Secure service or Secure driver would exceed the available heap memory for that component. The Secure World binary will incorrectly postlink; the postlinker produces a binary that can be integrated into a platform. Any use of the component with inadequate heap memory will fail at run-time when it used; in the case of drivers this typically means the system will not boot. Developers must ensure that their component's heap configuration provides adequate space for its memory requirements.

[4984] MAXIMUM SIZE OF CKO_DATA IS LIMITED TO 4KB

CKO_DATA objects up to 4KB can be manipulated with the External Cryptographic API. If you need to store larger objects, you are advised to use the External Secure Storage API instead of the CKO_DATA objects.

[5862] BUFFER SIZE FOR SINGLE OPERATION IS LIMITED TO 1MB

In the external and internal Cryptographic API, the buffer size for single operation is limited to 1MB. The work-around is to use multi-stage (update) operations.