Intel® Trusted Execution Technology (Intel® TXT) LAB Handout

September 2010

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPETY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL® PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel may make changes to specifications and product descriptions at any time, without notice.

All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Westmere, Tylersburg, Boxboro, and other code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user

Intel® Trusted Execution Technology does not prevent software attacks. It only provides a means to detect and mitigate software attacks against BIOS and Operating Systems that comply with Intel TXT requirements.

Intel, Xeon, Intel and Intel Sponsors of Tomorrow. and Intel Sponsors of Tomorrow. logo, and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others.

No computer system can provide absolute security under all conditions. Intel® Trusted Execution Technology (Intel® TXT) requires a computer system with Intel® Virtualization Technology, an Intel TXT-enabled processor, chipset, BIOS, Authenticated Code Modules and an Intel TXT-compatible measured launched environment (MLE). Intel TXT also requires the system to contain a TPM v1.s. For more information, visit http://www.intel.com/technology/security

Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, virtual machine monitor (VMM). Functionality, performance or other benefits will vary depending on hardware and software configurations. Software applications may not be compatible with all operating systems. Consult your PC manufacturer. For more information, visit http://www.intel.com/go/virtualization

The original equipment manufacturer must provide TPM functionality, which requires a TPM-supported BIOS. TPM functionality must be initialized and might not be available in all countries.

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the current quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Many factors could affect Intel's actual results, and variances from Intel's current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be the important factors that could cause actual results to differ materially from the corporation's expectations. Demand could be different from Intel's expectations due to factors including changes in business and economic conditions; customer acceptance of Intel's and competitors' products; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Additionally, Intel is in the process of transitioning to its next generation of products on 32nm process technology, and there could be execution issues associated with these changes, including product defects and errata along with lower than anticipated manufacturing yields. Revenue and the gross margin percentage are affected by the timing of new Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel’s response to such actions; defects or disruptions in the supply of materials or resources; and Intel’s ability to respond quickly to technological developments and to incorporate new features into its products. The gross margin percentage could vary significantly from expectations based on changes in revenue levels; product mix and pricing; start-up costs, including costs associated with the new 32nm process technology; variations in inventory valuation, including variations related to the timing of qualifying products for sale; excess or obsolete inventory; manufacturing yields; changes in unit costs; impairments of long-lived assets, including manufacturing, assembly/test and intangible assets; the timing and execution of the manufacturing ramp and associated costs; and capacity utilization. Expenses, particularly certain marketing and compensation expenses, as well as restructuring and asset impairment charges, vary depending on the level of demand for Intel's products and the level of revenue and profits. The majority of our non-marketable equity investment portfolio balance is concentrated in the flash memory market segment, and declines in this market segment or changes in management’s plans with respect to our investment in this market segment could result in significant impairment charges, impacting restructuring charges as well as gains/losses on equity investments and interest and other. Intel's results could be impacted by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Intel’s results could be affected by the timing of closing of acquisitions and divestitures. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust and other issues, such as the litigation and regulatory matters described in Intel's SEC reports. An unfavorable ruling could include monetary damages or an injunction prohibiting us from manufacturing or selling one or more products, precluding particular business practices, impacting our ability to design our products, or requiring other remedies such as compulsory licensing of intellectual property. A detailed discussion of these and other factors that could affect Intel’s results is included in Intel’s SEC filings, including the report on Form 10-Q for the quarter ended June 30, 2010.

Contents

Figures

Tables

1.1. Terminology

Term	Description
AC	*Authenticated Code*: Code that is authenticated and executed in an isolated environment within the processor. Authenticated Code runs internal to the processor and uses no other platform resource.
ACPI	*Advanced Configuration and Power Interface*. See ACPI specification, Revision 4.0, June 16, 2009
ACM	*Authenticated Code Module*: Platform specific code that is authenticated to the chipset and that is executed in an isolated environment within the processor. This term has also been used to denote *Authenticated Code Mode* that is a trusted environment enabled by an AC Module to perform secure tasks.
Authentication	A cryptographic method of verifying both integrity and ownership of a binary module. A module signed with a private / public key pair can be cryptographically authenticated.
Attestation	The ability to attest (prove) to the properties and platform configuration.
BIOS ACM	An Intel provided ACM embedded in BIOS. Executes before BIOS. Contains BIOS policy engine. Can verify BIOS.
Boot	This term is used to identify the process of starting an OS without using the secure launch process (in contrast to Launch)
CS	*Code Segment*
DMAR	*DMA Remapping Table* – see ACPI specification
DPR	*DMA Protected Range* – a region of system memory that is blocked to I/O device's memory accesses to prevent I/O devices from gaining access to Intel TXT restricted memory.
DRTM or D-RTM	*Dynamic Root Of Trust for Measurement* – Starts at any point in a system's software launch process, initiated by a secure event (SENTER).

Term	Description
E820 Memory Map	The system Memory Map as provided by BIOS via INT15 function call with e820 function code.
FIT	*Firmware Interface Table* – A data structure embedded in BIOS so that Intel microcode and ACM can locate BIOS components.
GDT	*Global Descriptor Table*
IDT	*Interrupt Descriptor Table*
Launch	This term is used to identify the process of starting an OS or VMM using the secure launch process (in contrast to Boot). It requires the OS or VMM to be measured and meet the platforms launch policy criteria
Logical Processor	Each physical processor may contain multiple cores and each core can support multiple physical threads (Hyper-Threading). Each of these threads is considered a logical processor.
Measurement	A cryptographic fingerprint of a binary module. Also called Hash or digest. A binary produces a unique hash using a hash algorithm like SHA 256. It is not possible to derive the original binary from the hash bytes.
MLE	*Measured Launch Environment*: The environment measured and launched as a result of the GETSEC [SENTER] instruction. This can be an Operating System, VMM, or any trusted code that supports Intel Trusted Execution Technology.
MSR	*Model Specific Register*
mVMM	*Measured Virtual Machine Monitor* - a particular class of MLE: A VMM/Hypervisor established through a measured launch that enables verification of its identity and protection from corruption. For server platforms the typical MLE is an mVMM.
PCR	*Platform Credential Registers* -- Dedicated registers in the TPM (sometimes referred to as Platform Configuration Registers).
REK	*Root Encryption Key*
SINIT	*Secure Initialization* – a trusted process (i.e., ACM) that measures, validates, and launches an MLE.
SIPI	*Startup Inter-processor Interrupt* (Startup IPI)
SMI	*System Management Interrupt* – an interrupt to invoke SMM

Term	Description
SMM	*System Management Module* – the code the handles RAS, hot plug, and other system
SMRR	*System Memory Range Registers*
SMX	*Safer Mode eXtensions*: the capabilities added to Intel processors which enable Intel Trusted Execution Technology.
SRTM or S-RTM	*Static Root Of Trust for Measurement* – The entity that starts the measurement chain – usually when chain of events begin at machine reset (*BIOS Boot Block*).
Intel TXT	Intel *Trusted Execution Technology*: Formerly codenamed LaGrande Technology (LT) and LaGrande Technology Server Extensions (LT-SX)
Tboot	*Trusted Boot:* is an open source, pre-kernel/VMM module that uses Intel Trusted Execution Technology (Intel TXT) to perform a measured and verified launch of an OS kernel/VMM.
TCB	*Trusted Computing Base* – Includes all the elements that make up the secure environment
TCG	*Trusted Computing Group* – Industry initiative for advancing computer security (http://www.trustedcomputinggroup.org)
TPM	*Trusted Platform Module* – a hardware device defined by the TCG that provides a set of security features used by Intel Trusted Execution Technology.
Verified Launch	This is a measured launch where the MLE is verified against a hash or list of hashes before being launched.
VMM	Virtual Machine Monitor (a.k.a. hypervisor or host OS): An operating system or a process that manages domain (virtual machine) life cycles, assigns resources to domains, and manages communications between domains.
VMX	Virtual Machine Extensions – a set of processor instructions defined by Intel® Virtualization Technology that software uses to provide isolation and protection for virtual environments. A part of VT-x.
VT-d	Intel® Virtualization Technology for Directed I/O – hardware support component of Intel® Virtualization Technology for managing DMA and interrupts generated by I/O devices.

Term	Description
VT-x	Intel® Virtualization Technology for Execution environment – a set of processor instructions (VMX) and capabilities defined by Intel® Virtualization Technology that software uses to provide isolation and protection for virtual environments.
Xen	An open source technology hypervisor/VMM. XEN v3.4 and later are Intel TXT aware.

1.2. Reference Documents

Document	Document No. / Location
[1] "*RS -Intel® Trusted Execution Technology Server Extensions (LT-SX) BIOS Specification"*	Doc#**27587**
[2] "*Intel Trusted Execution Technology Server Platform Design Guide*"	Doc#432407
[3] "*LT-SX External Architecture Specification*"	Doc#23937
[4] "*Dynamics of a Trusted Platform*" Book	ISBN 978-1-934053-17-1
[5] "Instruction Set Reference, A-M," in the Intel 64 and IA-32 Software Developer Manual, Volume 2A	http://www.intel.com/products/processor/manuals
[6] Intel TXT Toolkit for Servers	CDI#<tbs>
[7] "*Intel Initiatives TPM NV Storage Interface Usage*"	CDI #355867
[8] "Intel® Trusted Execution Technology (Intel TXT) – Trusted Platform Module (TPM) Nonvolatile (NV) Storage Interface Usage - Application Note"	CDI #420735
[9] Intel TXT-enabled Xen	http://xenbits.xensource.com/xenunstable.hg and http://xen.org/download/index.html

Document	Document No. / Location
[10] Intel TXT – Trusted Boot Checkout Kit	TBoot and related tools can be located via the following link. http://www.bughost.org/repos.hg/tboot.hg http://www.bughost.org/repos.hg/tboot.hg/file/e57acd4d1460
[11] EFI shell (DUET – FAT32)	http://developer.intel.com/technology/efi/agreesource.htm
[12] "Intel® Trusted Execution Technology: Software Development Guide: Measured Launched Environment Developer's Guide"	Intel® Trusted Execution Technology Software Development Guide Doc 315168
[13] "*Intel® Trusted Execution Technology – Launch Control Policy (LCP) Tools Reference Kit*"	Doc 362632
[14] Intel Virtualization Technology For Directed I/O (Intel® VT-d) Architecture Specification	http://download.intel.com/technology/computing/vptech/Intel(r)_VT_for_Direct_IO.pdf
[15] *ACPI Specification*	http://www.acpi.info/DOWNLOADS/ACPIspec40.pdf
[16] Intel TXT Website - Build a More Trusted and Secure Computing Environment	http://www.intel.com/technology/advanced_comm/trustedexecution.htm

Chapter 2 Roles & Responsibilities

2.1. Intel TXT Components

Figure 1 shows the major platform components and Intel TXT features.

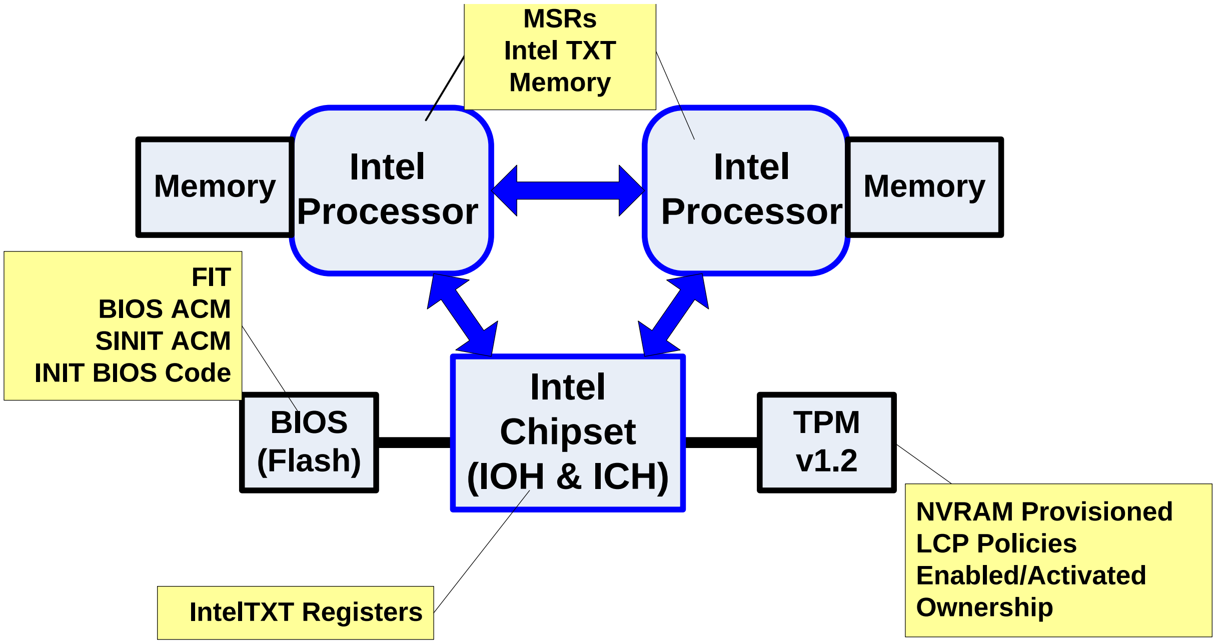

Figure 1 Intel TXT Platform Components

2.2. Test Tool Coverage

	Role	Responsibility	Tool / Test
1	TPM Provisioned with AUX & PD Policy Indices; PD Policy present and write protected; TPM Locked	OEM	TPMPROV **S_TXTINF**
2	All Processors Intel TXT capable	OEM	**S_TXTINF**
3	Chipsets Intel TXT capable	OEM	**S_TXTINF**
4	TPM v1.2 or later	OEM	**S_TXTINF**
5	FIT present and properly formatted	BIOS	**S_TXTINF**
6	Microcode updates	BIOS	**S_TXTINF**
7	BIOS ACM present and valid	BIOS	**S_TXTINF**
8	SINIT ACM present and valid	BIOS/OS	**S_GetSec**
9	ACPI DMAR Tables present and valid	BIOS	**S_TXTINF** VTK

	Role	Responsibility	Tool / Test
10	Reset vector within BIOS Initialization code	BIOS	**S_TXTINF**
11	Measure BIOS ACM – extend PCR0	microcode	LCP tools
12	Measure BIOS Initialization Code – extend PCR0	BIOS ACM	LCP tools
13	Measure SMM and other BIOS trusted code – extend PCR0	BIOS	**S_GetSec**
14	Measure SINIT ACM – extend PCR17	microcode	**S_GetSec**
15	Measure VMM – extend PCR18	SINIT ACM	**S_GetSec**
16	Intel TXT Policy = enabled	BIOS	**S_TXTINF**
17	TPM Policy = enabled	BIOS	**S_TXTINF**
18	TPM Enabled & Activated	Platform Owner	**S_TXTINF**
19	TPM initialized	BIOS	**S_TXTINF**
20	Processors configured properly (SMX, VMX) and locked	BIOS	**S_TXTINF**
21	Intel TXT Memory provisioned properly (MLE, SINIT ACM, Intel TXT Heap)	BIOS	**S_TXTINF**
22	Enable Intel® VT-d and Intel® VT-x	BIOS	**S_TXTINF**
23	Configure Intel VT-d and Intel VT-x	OSV	
24	Lock Memory Configuration	BIOS	**S_TXTINF**
25	Chipsets configured properly and locked	BIOS	**S_TXTINF**
26	Intel TXT Registers configured and locked	BIOS	**S_TXTINF**
27	TPM Ownership	Platform Owner	**S_TXTINF**
28	Intel TXT Status	ACMs	**S_TXTINF**
29	PO Policy	Platform Owner	**S_TXTINF**
30	PO Policy Data Structure	Platform Owner	**S_GetSec**
31	Enforce LCP	SINIT ACM	**S_GetSec**
32	Secure Launch	OSV	**S_GetSec**
33	LCP Maintenance	OSV Install/Update ISV	
34	Secrets Protection	BIOS	**S_GetSec & Secrets**

Chapter 3 Tools

3.1. TPM Provisioning Tools

Intel provides OEMs with a Factory TPM Provisioning Tool (TPMPROV.efi)

- Create AUX index
- Create PD Policy index
- Write PD Policy
- Write protect PD Policy
- Lock NVRAM

For the developer, the ACM release package contains tools to provision the NV_RAM and a special .bat file to create the Auxiliary Index needed for servers (note that a server's AUX size is different than the client AUX size).

For creating policy management tools and software utilities, the tools previously available in the Intel Trusted Execution Technology LCP Tools Reference Kit are now found using the following link: http://www.bughost.org/repos.hg/tboot.hg and browsing to the LCPTools directory. (http://www.bughost.org/repos.hg/tboot.hg/file/e57acd4d1460/lcptools)

This directory contains reference code for a set of tools to:

- Define TPM NV indices
- Create policies
- Write polices to the TPM

3.2. TBoot and OSV Tools for Intel TXT

Intel provides a reference MLE Boot loader (TBoot) for Linux that will perform all of the tasks necessary to perform a secure launch. Specific details can be found in the [10] Intel TXT – Trusted Boot Checkout Kit. TBoot source code and associated tools are available at http://www.bughost.org/repos.hg/tboot.hg and browsing to the appropriate subdirectory. TBoot source and associated tools can be found at: (http://www.bughost.org/repos.hg/tboot.hg/file/e57acd4d1460).

The TBoot kit provides tools for creating and modifying the Platform Owner LCP and LCP Data structure.

For creating policy management tools and software utilities, the tools previously available in the Intel Trusted Execution Technology LCP Tools Reference Kit are now available on that site. The LCPTools directory (http://www.bughost.org/repos.hg/tboot.hg/file/e57acd4d1460/lcptools) contains reference code for a set of tools to:

- Define TPM NV indices
- Create policies
- Write polices to the TPM

3.3. Compliance Test Tools

Intel is continually extending existing tests as well as developing new tests. Please check periodically for availability of new tools and to make sure you have the latest versions.

Table 1 Intel TXT Testing Tools

Tool	Toolkit
S_Txtinfo.efi S_Getsec.efi Secrets.efi	Intel Trusted Execution Technology Server EFI Tool Kit – Rev. 0.7 xxxxxxx
Pcrdump.efi	Intel Trusted Execution Technology EFI Tool Kit – Utility Software - Rev. 2.0.x 390756
read_aux.bat, read_pd.bat, read_po.bat	TPM BDK CDI#356536
Launch Control Policy Tools	Bat files and txt files available as part of ACM package. Source code avalable via http://www.bughost.org/repos.hg/tboot.hg
Intel TXT enabled VMM	Intel TXT-enabled Xen http://xenbits.xensource.com/xenunstable.hg and http://xen.org/download/index.html
EFI Test environment	EFI shell (DUET – FAT32) http://developer.intel.com/technology/efi http://biosguy.mattsinger.com/tech-library/make_duet/

3.3.1. Intel TXT Info Tool

S_TXTINF.efi tool verifies Intel TXT readiness of platform configuration and also displays various Intel TXT related information about processor, chipset, TPM, Intel TXT registers, heap memory, BIOS structure, ACPI DMAR tables and ACMs.

This tool:

- Checks all cores in each processor, their ability to support Intel TXT, and validates that BIOS has enabled Intel TXT and VT-x on all cores.
- Checks all IOHs, their ability to support Intel TXT and validates that BIOS has properly enabled Intel TXT on them.
- Validates that the FIT exists, is valid, contains all of the required records, and that the records are valid. If displays the FIT contents.
- Validates that the BIOS ACM is valid and matches the target platform.
- Validates that BIOS has enabled VT-d and built a DMAR VT-d table (ACPI).
- Verifies that the TPM has been provisioned properly.
- Reads and reports Intel TXT status registers.
- Determines if the platform is properly enabled for Intel TXT and ready for secure launch.

3.3.2. GetSec Tool

S_Getsec.efi tool allows execution of several leaves of SMX GETSEC instruction:

☐ GETSEC [SENTER]

☐ GETSEC [ENTERACCS]

☐ GETSEC [SEXIT]

Execution is performed either in normal or simulation mode

3.3.3. Secrets Tool

Secrets.efi tool issues TXT.SECRETS or TXT.NOSECRETS command to set or clear the TPM Secrets flag.

3.3.4. PCR Dump Tool

PCRDunp.efi displays content of TPM PCRs

Chapter 4 Test Strategy & Tools

4.1. Test Procedures

4.1.1. Intel TXT Tool Kit (TTK)

The S_TXTInf.efi tool performs many of the checks and tests needed to verify Hardware and BIOS compliance with Intel Trusted Execution Technology. This tool is available for both 32-bit EFI and 64-bit EFI environments. The 32-bit version of this tool (and any of the EFI based tools) can be run on a non-EFI32 platform by creating a bootable EFI32 USB flash drive and then enabling the platform to boot from USB.

S_GETSEC.efi performs several Intel TXT functions to validate that the platform is capable of a secure launch.

Secrets.efi is used to test that BIOS is capable of properly handling resets when secrets are in memory.

The ultimate test is launching an Intel TXT enabled VMM such as XEN 3.4 or later.

4.1.2. Testing HW/SW Compliance

Step 1:

Execute S_TXTINF –C:a –a –V:2 > LogFile.txt

Edit LogFile.txt and search for “ERROR”. There should be none

Search for “WARN”. Warnings will indicate problems for production platforms that might be acceptable for development platforms.

The S_TXTINF.efi tool checks many aspects of BIOS implementation and execution.

It is runs from an EFI shell. On non-EFI (legacy) BIOS, a bootable EFI shell is necessary.

The S_TXTINF.efi checks:

- Processor supports Intel TXT including server extensions
- IOH supports Intel TXT
- Processor and IOH have been properly configured for Intel TXT operation
- The TPM is enabled and properly provisioned
- Intel TXT Memory is properly protected
- FIT is properly formed and contains required records

- BIOS ACM is valid and matches the chipset
- Type 7 FIT records meet the required coverage
- BIOS ACM did not report any errors
- SINIT ACM did not report any errors
- BIOS called required Intel TXT initialization functions on all processors
- BIOS has properly programmed Intel TXT registers and Intel TXT Heap
- BIOS has properly configured and programmed Intel TXT Heap
- BIOS has properly locked memory configuration
- ACPI DMAR table exists – run VTK to validate VT-D tables

Step 2:

Run Virtualization Tool Kit (VTK) to validate that BIOS properly builds the ACPI DMAR tables. Note: S_TXTINF.efi checks the BIOS built the DMAR tables, but the VTK provides a more in-depth set of tests. The VTK runs in 32-bit and 64-bit Windows* environment.

4.1.3. Testing Measured Launch

This includes launching a measured environment to validate that Intel TXT components are working together to establish a measured environment. This test assumes that S_TXTINF reports Intel TXT is enabled and there are no Intel TXT errors.

Step 1:

For 32-bit EFI: Run S_GETSEC SEN <SINIT_filename>

For 64-bit EFI: Run S_GETSEC –L SEN –a <SINIT_filename>

This test executes SENTER and tests the ability to launch a secure environment. If the test completes without an Intel TXT reset, proceed with step 2.

Step 2:

For 32-bit EFI: Run S_ GETSEC SEXIT

For 64-bit EFI: Run S_GETSEC –L SEXIT

Step 3:

Launch an MLE such as the Intel TXT enabled Xen open source project.

Validation can be done using any MLE that has been configured to be launched using SINIT ACM. This will test the launch and the Launch Control Policy.

To Test:

[1] Install and launch an MLE.

Note: it is important to use an Intel TXT enabled MLE. Just adding TBoot to an OS loader can cause problems because when the OS shuts down, it will not properly clear secrets and Intel TXT will detect that as a threat.

- □ Confirm SINIT successfully launched the MLE and forwarded control to the MLE.
  For the Xen TBoot case, Xen will successfully launch
  - o TXT_STATUS (offset 0x0 in Intel TXT address space):
    - ■ [bit 0: TXT_SENTER_DONE_STS] = 1;
    - ■ [bit 1: TXT_SEXIT_DONE_STS] = 0
  - o TPM PCR 17/18 has updated measurements
  - o Intel's Ltview/Lttest display values from key Intel TXT registers
- □ TXT.CRASH register can be used to diagnose failure condition
  - o S_TXTINF –c –a will display error information

[2] Launch Control Policy

Set platform supplier policy to "launch any" as described in section 3.2 of the "Intel Trusted Execution Technology: Launch Control Policy Architecture Specification" This is the recommended Intel TXT launch configuration.

MLE Options:

- o S_getsec.efi SENTER <SINIT ACM> Launches a simple shell MLE
- o S_getsec.efi SEXIT Brings down the MLE
- o EFI-shell> mm to look at Intel TXT environment
- o SV2.9, Xen-Intel TXT patch

4.1.4. Testing Teardown

In normal operation when SEXIT is used, confirm that the MLE take down does not behave differently from a non-measured environment take down.

Special testing consideration occurs when an MLE is brought down improperly (e.g. sudden power loss) and the SECRETS flag has been set by the MLE. The tests below must cause BIOS to scrub memory after the power cycle.

Test Methods:

Note: System restore BIOS image is advised.

Method 1:

1. Run S_GETSEC SENTER as described above.
2. Set the secret bit -- use Secrets.efi /s
3. Power cycle

4. Platform should boot normally – verify by running S_TXTINF.efi

Method 2:

1. Bring up MLE as above.
2. Power down by pulling power plug
3. Power up
4. Platform should boot normally – verify by running S_TXTINF.efi

Chapter 5 Intel TXT Reference Material

5.1. Chipset Registers

Table 2 Chipset Registers for Intel TXT

TXT.DPR
• Protects memory immediately below TSEG (or UMA or TOLUD) from DMA • BIOS must program the size to 3 MB to protect Intel TXT Device memory
TXT.SINIT.MEMORY.BASE and TXT.SINIT.MEMORY.SIZE
• Storage space for SINIT module prior to SENTER, protected by DPR • BIOS must reserve 128K for this region
TXT.HEAP.BASE and TXT.HEAP.SIZE
• Data area used by Intel TXT software, protected by DPR • BIOS must reserve 896K for this region
TXT.CRASH Register (aka TXT_ERROR Register)
• Contents preserved across reset • Contains error codes generated by ucode, SINIT ACM, and the MLE • Encoding defined in release notes for the AC modules
TXT.AcmCrashCode Register
• Contents not preserved across reset • Contains error codes generated by BIOS ACM • Encoding defined in release notes for the BIOS ACM
TXT.Status Registers
• Status and configuration registers

5.2. Safer Mode Extensions (SMX)

Table 3 GETSEC Leaves

■ **GETSEC [CAPABILITIES]** – returns a bit vector of supported GETSEC functions.
■ **GETSEC [PARAMETERS]** – returns specific parameter information supported by the processor.
■ **GETSEC [SENTER]** – invokes the process to measure and start the MLE. It stops all threads but the calling thread, which must be Boot Strap Processor (BSP).
■ **GETSEC [ENTERACCS]** - invokes the process to measure and start the AC code. This leaf can only be invoked on the BSP.
■ **GETSEC [ACEXIT]** – Performs a jump to the MLE entry point after MLE validation, changes the paging mode, sets CS:EIP & SS:ESP, and returns the processor cache to normal operation. This instruction has also been named EXITAC
■ **GETSEC [SEXIT]** – terminates the MLE so that another MLE can be loaded without a reboot
■ **GETSEC [SMCTRL]** – Perform SMX specific mode control operations (such as re-enable SMI events after SENTER).
■ **GETSEC [WAKEUP]** – Sends signal to all processors currently in the SENTER sleep state to wake up (used by MLE).

5.3. BIOS Components

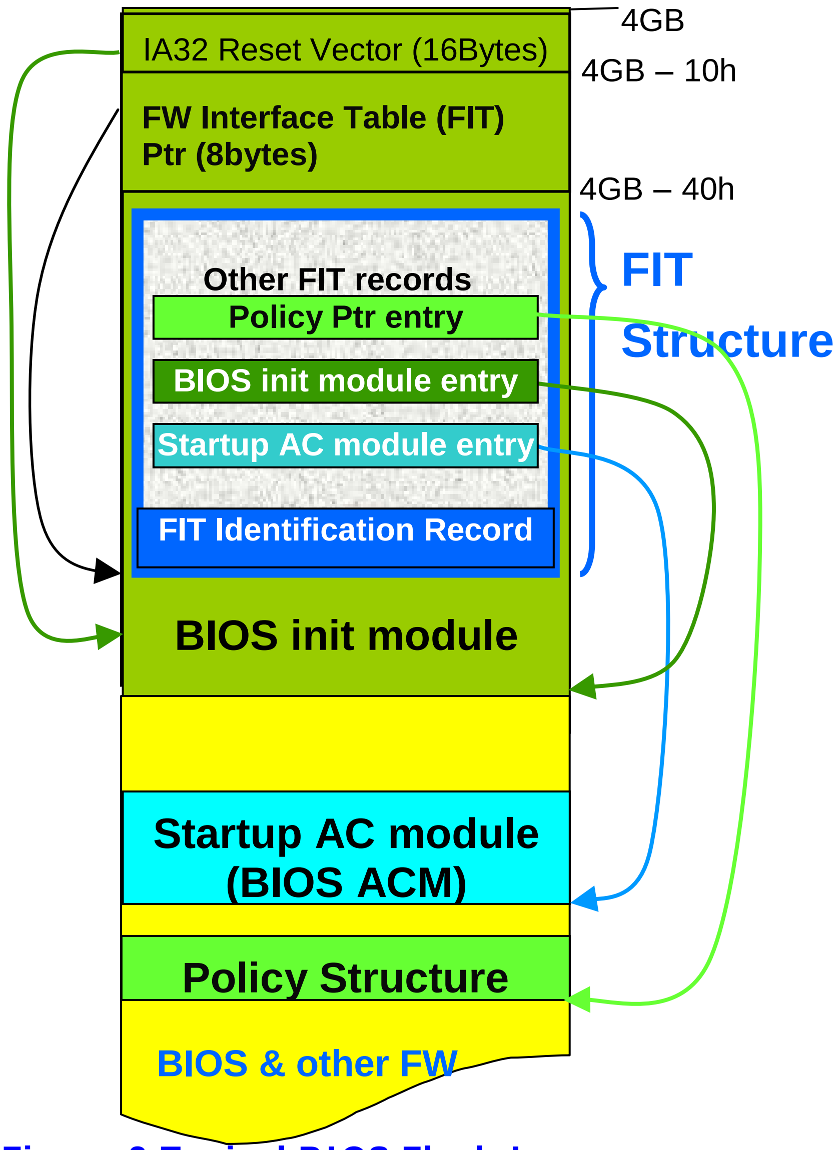

Figure 2 Typical BIOS Flash Image

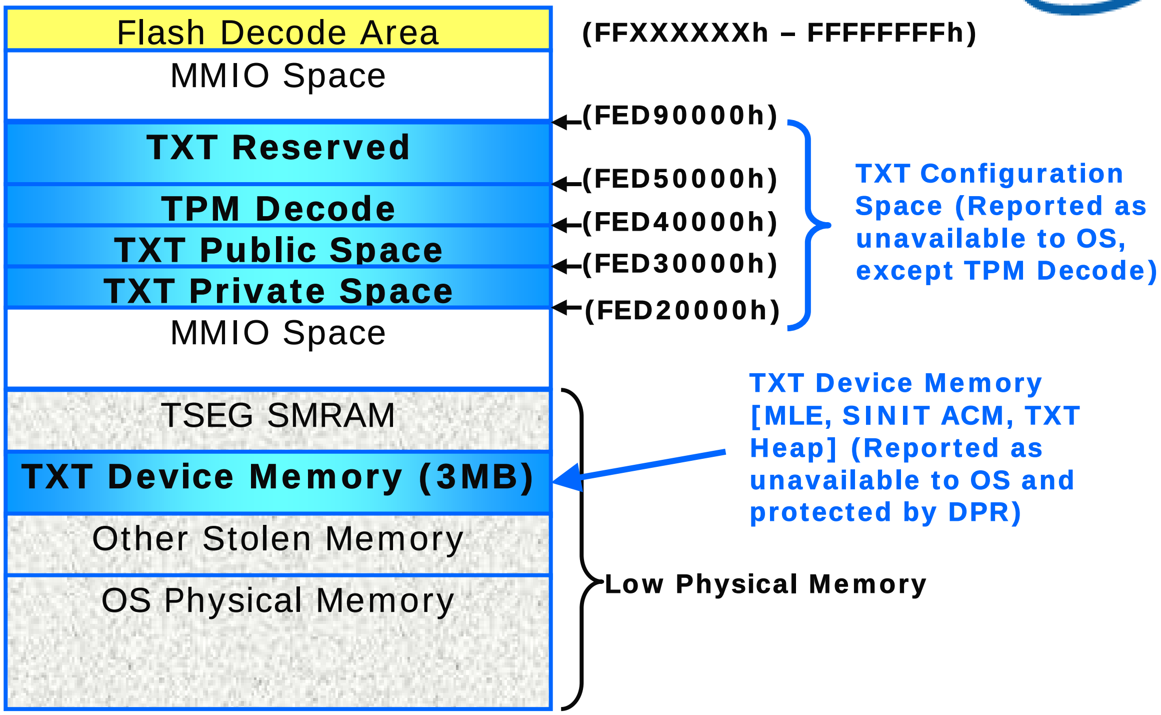

Figure 3 Intel TXT Memory Map

5.3.1. Firmware Interface Table (FIT)

Record Type	Number of Records	Definition
0	1	FIT header
1	0 or more	Microcode Update
2	1	BIOS AC Module (base & size)
3-6	0	reserved
7	1 or more	BIOS Init Module (base & size)
8	0 or 1	TPM Policy Record
9	0 or 1	BIOS Policy Data Record
10	0 or 1	TXT Configuration Policy

Records appear in order of Record Type

5.4. ACM Error Code Locations

5.4.1. Description:

Intel TXT error code (crash code) in CRASH register (FED20030h) is NOT cleared by system reset.

BIOS ACM error/progress codes in the BIOSACMCode register (FED20328h) is non-sticky (i.e., not sustained across a hardware reset).

5.4.2. Intel TXT CRASH Codes

The CRASH register is cleared only by a power good reset.

Table 4 LT.Crash Code Format

Bit	Name	Description
31	Valid	Valid error when set to '1'. The rest of the register contents should be ignored if '0'.
30	External	External Internally vs. externally induced. '0' if induced from the processor, '1' if induced from external software
29:16	reserved	Reserved
15:0	Type	This is implementation and source specific. It provides more details on what step was being performed at the time a failure condition was detected.

Table 5 Processor Initiated TXT-Shutdown Codes

Value	Mnemonic	Type Error condition
0	#LegacyShutdown	Legacy (non-SMX specific) shutdown
1 - 4		Reserved
5	#BadACMMType	Authenticated RAM load memory type error
6	#UnsupportedACM	Unrecognized AC module format
7	#AuthenticateFail	Failure to authenticate
8	#BadACMFormat	Invalid AC module format
9	#UnexpectedHITM	Unexpected snoop hit detected
10	#IllegalEvent	Illegal event

Table 5 Processor Initiated TXT-Shutdown Codes

11	#BadJOINFormat	Invalid JOIN format
12	#UnrecovMCError	Unrecoverable machine check condition
13	#VMXAbort	VMX abort
14	#ACMCorrupt	AC memory corruption
15	#IllegalVIDBRatio	Illegal voltage/bus ratio
16 – 65535		Reserved

5.4.2.1. BIOS ACM Progress / Error Codes

The following tables provide definition of the BIOSACMCode register (FED20328h):

Table 6 BIOS ACM Progress/Error Code Format

Bit	Name	Description
31	Valid	Valid error when set to ‘1’. The rest of the register contents should be ignored if ‘0’
30	External	Internally vs. externally induced. ‘0’ if induced from the processor, ‘1’ if induced from external software.
29:25	Unused	
24:0	Type	This is implementation and source specific. It provides more details on what the step was being performed at the time a failure condition was detected

The following table provides type field definition for AC modules:

Table 7 BIOS ACM Type Field Format

Bit	Description
24:16	TPM command return code, valid only if the [9:4]=0x0d

Table 7 BIOS ACM Type Field Format

15	‘0’ if crash generated by AC code, ‘1’ if not. If ‘0’, definition of bits 24:16 and 14:0 apply.
14:10	AC module error codes.
9:4	AC module progress codes.
0:3	AC Module Type 0000 BIOS AC 0001 SINIT 0010 - 1111 Reserved for future use

The following table provides progress and Error Codes for BIOS ACM:

Table 8 BIOS ACM Progress/Error Codes

Progress Code	Error Code	Description
01h	0000	BIOS ACM Entry Point
	0001	HEAP Uninitialized
	0010	DPR Uninitialized
02h	0000	Initial Checks
	0001	Non-supported Device.ID
	0010	Non-supported Extended.ID
	0011	Current BIOS AC module not registered in the TPM NVRAM
	0100	Memory is not locked
	0101	SENTER used to launch the AC module
03h	0000	Start MTRR Check
	0001	MTRR Rule 1 Error
	0010	MTRR Rule 2 Error
	0011	MTRR Rule 3 Error
	0100	MTRR Rule 4 Error

Table 8 BIOS ACM Progress/Error Codes

Progress Code	Error Code	Description
	0101	MTRR Rule 5 Error
	0110	MTRR Rule 6 Error
	0111	Invalid MTRR mask value
	1000	Invalid MTRR mapping
04h	0001	Memory Unlock and Scrub
	0010	Memory unlock and scrub failure.
	0011	Memory Reference Code failure.
	0100	Memory Locked and Scrub
0dh	0000	TPM_Extend attempt
	0001	TPM access register contents invalid
	0010	Unable to get access to the locality
	0011	TPM status register contents invalid
	0100	TPM not ready to accept a command
	0101	Command failed
	0110	Output buffer for the TPM response to short
	0111	Input parameter for the function invalid
	1000	Invalid response from the TPM
	1001	Time out for TPM response
	1010	TPM returned an error
1011		TPM NV RAM not locked
	1100	TPM is disabled
	1101	TPM is deactivated
	1110	TPM NV indices incorrectly defined

Table 8 BIOS ACM Progress/Error Codes

Progress Code	Error Code	Description
11h	0000	Miscellaneous
	0001	Interrupt Occurred
	0100	Pre-Production code not allowed (check LCP Control Policy; PO overrides PD)
13h	0001	One of the FIT table checks failed.
	0010	Verify BIOS failed.
	0011	BIOS policy failure.
14h	0001	Memory locked when ClearSecrets is called.
	0010	BIOS un-trusted when ClearSecrets is called.
15h	0001	Memory locked when LockConfig is called.
16h	0001	TPM policy data bad
	0010	FIT table end not below 4GB
	0011	Didn't find BIOS startup record that includes reset vector
	0100	Didn't find BIOS startup record that includes FIT pointer
	0101	Found overlap between BIOS startup regions.
	0110	Found overlap between a BIOS startup region and the startup ACM.
00h	0000	BIOS ACM Exit Point

Chapter 6 Example S_TXTInf Log

```
SERVER TXTINFO. NUMBER OF TBG CHIPSETS FOUND=1
SERVER TXTINFO. GET_FIT_TABLE() RETURNS # OF FIT ENTRIES=18
DEBUG====Found AC Module in FIT Table at index=10
TXTINFO ==== START LOG BIOS AC MODULE HEADER. SEE MLE DEV GUIDE FOR MORE INFORMATION ====
TXTINFO ==== MODULETYPE VAL=65538
TXTINFO ===  HEADER LENGTH = 161
TXTINFO ===  HEADER VERSION = 0x0
TXTINFO ===  CHIPSET ID= 0x3400
TXTINFO ===  MODULE VENDOR 0x8086
TXTINFO ===  MODULE DATE 0x20100114
TXTINFO ===  MODULE SIZE (32-bit quants) 6832
TXTINFO ===  MODULE FLAGS 0x4000
TXTINFO ==== END LOG BIOS AC MODULE HEADER. SEE MLE DEV GUIDE FOR MORE INFORMATION ====
CPU count = 12

CPU 0 info:
  ID (EAX(CPUID(1))              = 0x206C2
  MCU Rev. (MSR(0x8B)[63:32])    = 0x5
  SMX support (ECX(CPUID(1))[6]) = 1
  VMX support (ECX(CPUID(1))[5]) = 1
  SMRR support (MSR(0xFE)[11])   = 1
IA32_FEATURE_CONTROL MSR(0x3A):
  LOCK [0])                      = 1
  VMXON in SMX enable [1])       = 1
  VMXON outside SMX enable [2])  = 1
  SMRR enable [3])               = 0
  SENTER control [14:8])         = 0x7F
  SENTER enable [15])            = 1
GETSEC[CAPABILITIES]: 0x1fd
GETSEC[PARAMETERS] total parameters supported=4
  TXT chipset present [0])       = 1
  ENTERACCS available [2])       = 1
  EXITAC available [3])          = 1
  SENTER available [4])          = 1
  SEXIT available [5])           = 1
  PARAMETERS available [6])      = 1
  SMCTRL available [7])          = 1
  WAKEUP available [8])          = 1

CPU GETSEC MAXIMUM PARAMETER SUPPORTED=4
CPU : AC MODULE EXECUTION REGION SIZE= 65536(bytes)
 LT-SX Support flag = 1
GETSEC[PARAMETERS]: 0x65
  Server Extensions Support EAX[4:0]       = 5
  Feature Flags EAX[31:5]:
       Machine Check Handling [6]         = 1
       Processor based CRTM support [5]   = 1
```

{The previous information is repeated for each core and that informat has been removed to abbreviate the example}

```
Chipset info:
  MCH/IMC DID (PCI 0:0:0:2 [15:0])                 = 0x3406
  MCH/IMC RID (PCI 0:0:0:8 [7:0])                  = 0x22
  ICH/PCH DID (PCI 0:1F:0:2 [15:0])                = 0x3A16
  ICH/PCH RID (PCI 0:1F:0:8 [7:0])                 = 0x0
  LT Debug disable (PCI 0:14h:2:D4h [31])        = 1
  TXT Server disable (PCI 0:14h:2:D4h [30])        = 0
  TXT disable (PCI 0:14h:2:D4h [29])               = 0

TXT Registers info (Offsets from public space at 0xFED30000):
  STS: Locality 2 open (0 [16])    = 0
  STS: Locality 1 open (0 [15])    = 0
  STS: Locality 3 open (0 [14])    = 1
  STS: SMM open (0 [13])           = 0
  STS: PMRC lock (0 [12])          = 0
  STS: Mem CFG OK (0 [11])         = 0
  STS: NTP enable (0 [10])         = 0
  STS: Private open (0 [7])        = 0
  STS: Mem CFG lock (0 [6])        = 0
  STS: Mem unlock (0 [4])          = 0
  MCH in debug mode (100 [31])     = 1 {Pre-production platform – must be 0 for Production CPUs}
  ESTS: Wake error (8 [6])         = 0
  ESTS: Rogue status (8 [1])       = 0
  ESTS: TXT Reset (8 [0])          = 0 {if not 0, there is a problem}
  Crash (30 [31:0])                = 0x0
    Progress code                  = 0x0
    Error code                     = 0x0
    Module type                    = 0x0
  VID (110 [15:0])                 = 0x8086
  DID (110 [31:16])                = 0xC000
  RID (110 [47:32])                = 0x3F
  DPR Capable (200 [26])           = 1
  PMRC Capable (200 [19])          = 1
  SINIT base (270 [64:0])          = 0x1F700000
  SINIT size (278 [64:0])          = 0x20000
  HEAP base (300 [64:0])           = 0x1F720000
  HEAP size (308 [64:0])           = 0xE0000
  MSEG base (310 [64:0])           = 0x0
  MSEG size (318 [64:0])           = 0x0
  Top of DPR (330 [31:20][19:0=0]) = 0x1F800000
  DPR size (330 [11:4])            = 0x3
  DPR lock (330 [0])               = 1
  FIT Fallback (340 [3])           = 0
  FIT Measured (340 [2])           = 0
  SACM Failed  (340 [1])           = 0
  FIT Failed   (340 [0])           = 0
  Public Key Hash (400 [191:0])    = 08 77 7B 21 EC 4D 7F CE
                                     F7 68 2A 26 96 BC 5F 42
                                     A9 96 45 A4 21 81 10 7F
                                     87 70 C2 24 37 FD E0 2C

Heap memory info (Offsets from 0x1F720000):
BiosOSDataSize (0)        = 0x2C
BiosOSData region (8):
  Version                 = 3
    BIOS SINIT Size         = 0x0
  LCP PD Base             = 0x0
  LCP PD Size             = 0x0
  Number of logical CPUs  = 0xC
  Flags                   = 0x0
OsMLEDataSize (0x2C)      = 0x0
OsMLEData region is not allocated.

TPM info (Offsets from 0xFED40000):
  ACCESS: Establishment (0 [0])    = 1
  ACCESS: Active locality (0 [5])  = 1
  ACCESS: Valid (0 [7])            = 1
  VID: (0xF00)                     = 0x15D1
  DID: (0xF02)                     = 0xB
  RID: (0xF04)                     = 0x10
  NV index 0x50000001 (LCP Supplier) value:
  02 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  00 00 02 00 00 00 00 00 00 00 00 00 00 00
  NV index 0x40000001 (LCP Owner) value:  {not required} Note that Owner LCP is usually not
  02 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  present until end-user takes
  00 00 02 00 00 00 00 00 00 00 00 00 00 00                          possession of the platform
  NV index 0x50000002 (Auxiliary) value:
  00 C4 00 00 00 1A 00 00 00 00 00 00 FF FF FF FF FF FF FF FF
  FF FF FF FF FF FF FF FF FF FF FF FF 19 40 62 1C 3F C7 93 EF
  56 DA 9A A2 60 42 FE 0C 39 63 29 7F FF FF FF FF FF FF FF FF
  FF FF FF FF
TPM is enabled.
Warning: TPM NV Storage is unlocked. {OK for development systems – must be locked for Production}
  TPM Permanent Flags value:
    disable                        = 0
    ownership                      = 1
    deactivated                    = 0
    readPubek                      = 0
    disableOwnerClear              = 0
    allowMaintenance               = 0
    physicalPresenceLifetimeLock   = 0
    physicalPresenceHWEnable       = 0
    physicalPresenceCMDEnable      = 1
    SECPUsed                       = 0
    TPMPost                        = 0
    TPMPostLock                    = 0
    FIPS                           = 0
    enableRevokeEK                 = 0
    nvLocked                       = 0
    readSPKPub                     = 0
    tpmEstablished                 = 0
TPM is activated.
  TPM StClear Flags value:
    deactivated                    = 0
    disableForceClear              = 1
    physicalPresence               = 0
    physicalPresenceLock           = 0
    bGlobalLock                    = 0

DMAR table found at 0x1F4CD000:
  Signature                        = DMAR
  Length                           = 0x1D0
  Revision                         = 0x1
  Checksum                         = 0x78
  OEMID                            = 0x49 0x4E 0x54 0x45 0x4C 0x20
  OEMTableID                       = 0x54 0x48 0x52 0x4C 0x59 0x20 0x20 0x20
  OEMRevision                      = 0x1
  CreatorID                        = 0x4D 0x53 0x46 0x54
  CreatorRevision                  = 0x100000D
  HostAddressWidth                 = 0x27
  Flags                            = 0x1
```

{removed – DMAR tables will be platform specific – Virtualization Tool Kit should be used to check VT-d compliance}

{NOTE: FIT Table size and content will vary for every implementation}

```
SERVER TXTINFO : START LOG: FIT TABLE :
 # of ENTRIES IN PLATFORM FIT TABLE=18
Entry: 0
Address:        2020205F5449465Fh    _FIT_
Size:             12h
Version:        0100h
Type:           00h           Must be exactly one Type 0 “FIT HEADER” record
C_V:            01h
Checksum:       6Dh

Entry: 1
Address:        00000000FFF40060h
```

```
Size:             180h
Version:        0100h
Type:           01h            May be zero or more Type 1 “Microcode Patch” records
C_V:            00h
Checksum:       00h

    {removed – Entries 2 – 9 are additional microcode patches

Entry: 10                 Must be exactly one Type 2 “BIOS ACM” record
FIT TABLE ENTRY : BIOS AC MODULE : Address:        00000000FFD30000h
Size:             1000h
SERVER TXTINFO. ERROR TYPE2 ENTRY SIZE IS NOT ZERO. INVALID SIZE FIELD FOR TYPE2
FIT TABLE ENTRY : BIOS AC MODULE : Version:        0100h
FIT TABLE ENTRY : BIOS AC MODULE : Type:           02h
FIT TABLE ENTRY : BIOS AC MODULE : C_V:            00h
FIT TABLE ENTRY : BIOS AC MODULE : Checksum:       00h

Entry: 11
Address:        00000000FFF90000h
Size:             1000h
Version:        0100h
Type:           07h       Must be at least one or more Type 7 “BIOS Startup code” records
C_V:            00h
Checksum:       00h

Entry: 12
Address:        00000000FFFA0000h
Size:             1000h
Version:        0100h
Type:           07h
C_V:            00h
Checksum:       00h

Entry: 13
Address:        00000000FFFB0000h
Size:             5000h
Version:        0100h
Type:           07h
C_V:            00h
Checksum:       00h

Entry: 14
Index Register IO Address:   0070h
Data Register IO Address:    0071h
Access Width in Bytes:       01h
Bit Position:                05h
Index:                       002Ah
Size:             00h
```

```
Version:                     0000h
Type:                        08h   {optional – ignored when there is a Type 0Ah record}
C_V:                         00h
Checksum:                    00h

Entry: 15
Address:        00000000FFFFE743h
Size:             06h
Version:        0100h
Type:           09h   {optional pointer to Platform supplier LCP data structure, if one}
C_V:            00h
Checksum:       00h

Entry: 16
Address:        002A040100710070h
Size:             00h
Version:        0000h
Type:           0Ah   {optional points to the flag indicating if TXT is enabled on this platform}
C_V:            00h
Checksum:       00h

Entry: 17
Address:        002A060100710070h
Size:             00h
Version:        0000h
Type:           7Fh
C_V:            00h
Checksum:       00h

SERVER TXTINFO : END LOG: FIT TABLE :
```

{The following is the summary information (short form of the information previously displayed) and is always output regardless of the Verbose setting}

```
Configuration summary:
CPU summary:      Count            = C
  CPU 0:          ID               = 0x206C2
                  MCU Rev.         = 0x5
  Capability:     SMX/VMX/SMRR     = 1/1/1
  IA32_FEATURE_CONTROL MSR enables:
  Senter/SenterCtrl/SMRR/VMX/SMX/Lock = 1/0x7F/0/1/1/1
```

{The previous information is repeated for each core and has been removed - it is also a short form of the information previously displayed. The following information is also a summary of information already displayed}

```
Chipset summary:  MCH/IMC DID      = 0x3406
                  ICH/PCH DID      = 0x3A16
```

```
  MCH/IMC/IIO/IOH is running in TXT mode.
TXT Registers summary: DID         = 0xC000
                  SINIT Base/Size  = 0x1F700000/0x20000
                  Heap Base/Size   = 0x1F720000/0xE0000
                  MSEG Base/Size   = 0x0/0x0
                  DPR Size/Lock    = 0x3/1
Heap summary, BiosOSData region:
                  Size             = 0x2C
                  Version          = 3

TPM summary:      DID              = 0xB
  NV index 0x50000001 (LCP Supplier) is allocated.
  NV index 0x40000001 (LCP Owner)   is allocated.
  NV index 0x50000002 (Auxiliary)   is allocated.
TPM is enabled.
TPM is activated.

VT-d summary:
DMAR table found at 0x1F4CD000:
    VT-d Engine #1 Base Address = 0xFE710000
  RMRR Tables:
    RMRR 0 at 0x1F4CD050:
    RMRR 1 at 0x1F4CD0A8:
    RMRR 2 at 0x1F4CD0C8:
    RMRR 3 at 0x1F4CD0E8:
    RMRR 4 at 0x1F4CD108:
    RMRR 5 at 0x1F4CD128:
    RMRR 6 at 0x1F4CD148:
    RMRR 7 at 0x1F4CD168:
    RMRR 8 at 0x1F4CD188:
  ATSR Tables:
    ATSR 0 at 0x1F4CD1A8:

Warning: TPM NV Storage is unlocked.

All checks are passed. Platform appears to be correctly
configured for establishing of TXT environment.
BIOS ACM found in flash part at address 0xFFD30000

ACM file info (Offsets from 0xFFD30000):
  Module type                      = 0x10002
  Header length                    = 0xA1
  Header version                   = 0x0
  Chipset ID                       = 0x3400
  Flags                            = 0x4000
    Module is production chipset key signed.
    Pre-production module.
  Module vendor                    = 0x8086
  Module date                      = 0x20100114
  Module size                      = 0x1AB0
  Code control                     = 0x0
  Error entry point                = 0x0
  GDT size                         = 0x20
  GDT base pointer offset          = 0x700
  Segment selector                 = 0x8
  Module entry point               = 0x54FB
  Key size                         = 0x40
  Scratch field size               = 0x8F
  Module public key:
  41 F6 9C B0 C2 BF A1 B3 F9 4E 6A F0 28 E9 60 EC 8D 5F F0 95
  B5 51 E2 FB 6E E1 F8 B4 A1 3A E2 1D 31 D3 B1 5E 79 6C 22 0F
  2D 48 3B 97 B0 D8 36 17 9D 7D 96 02 C3 C9 EF F5 28 CA 4D 7C
  C2 94 A7 2B BE 80 80 81 8F B7 89 6C 84 77 2B 5F F1 61 EC C6
  1C 1C 9B 83 7D B8 83 62 35 13 18 87 FC 79 29 1A 57 51 FA 1B
  25 35 E9 0F 96 FC 07 13 4E 53 95 D7 AD EE 7A 6A AF 93 51 5A
  DB 45 80 9D 11 AB EB 53 44 AA 9D 18 97 E1 BE 32 B0 7A FA 4C
  A6 B8 1F 3A DD 63 C4 DD 77 3C 94 E1 6B F7 DF 15 DB DF AE C5
  C4 05 33 94 70 57 D9 39 40 9D 02 25 4F 8C FF E1 A7 02 39 01
  44 85 FE CD B5 22 CC CD 4A 78 70 D0 D2 AB 8E 98 89 CC A1 C0
  17 A2 34 78 43 7E 18 22 31 AC B5 5C 7F AE 19 7A AD 9E E7 56
  3F 31 02 9E 7A 3F 67 81 4F 39 26 79 25 0A BB F8 42 8A B7 C3
  69 FA A5 72 22 F0 70 06 D5 17 5D 05 15 30 F0 D8
  Module public key exponent       = 0x11
  Module signature:
  B0 54 FC 28 82 91 39 5B 88 D1 D1 FC BE 17 C7 B8 53 69 AD FD
  8B 2D 3F 1A 67 C7 14 4E 38 FD 59 6F A5 FD 38 CC 44 B7 8F 41
  68 28 56 0B CB FB 49 FF F9 24 C4 FA 7E E4 99 D4 F2 B3 58 D9
  28 28 C4 E4 1A 47 3B E4 B8 AA 41 00 5B 91 06 A1 D1 6F 48 3E
  47 32 54 45 17 FE 82 BD A2 9F BA DB D2 16 CB 8E 55 C0 E2 94
  9A 72 C0 C8 4D 73 C0 10 EC CC EC 82 79 6F 85 A6 46 D6 7E 72
  DC AF 98 64 55 AB 42 D4 9A B6 49 5D 1D 50 83 91 A5 62 66 8A
  1D 86 15 9F 02 E5 8F B6 6F 9B 61 6B 8B 02 3B 13 4F 44 DF 73
  39 49 C1 87 1E F2 BE 1E 6F CF 15 3A 29 44 1A AE 04 44 67 61
  05 E5 D8 6C 6F 2B 59 C5 AE 84 35 CB 58 6E 9A 29 3B 0B 84 62
  25 2D BA 03 25 DB 9E 1D EA 7F 5D 67 80 83 ED 6B 11 0D C4 30
  00 90 03 35 03 26 0B 1C 00 D0 55 31 57 AA 08 CE C3 47 49 23
  C1 F8 B6 84 4D 3F 73 A2 ED 61 F6 C0 48 30 3F 7E
  Module hash                      = 68 8B E2 AB E1 BC 16 3F 75 A5
                                     BA C6 B2 10 C5 FA A3 78 F6 03
                                     68 8F 53 A4 A1 91 69 20 0B A9
                                     AB 54
Chipset ACM Information Table (0x4C0):
  UUID = 0x7FC03AAA 0x18DB46A7 0x8F69AC2E 0x5A7F418D
  Chipset ACM Type                 = 0x0
  Table Version                    = 0x3
  Table Length                     = 0x28
  Chipset ID list table offset     = 0x4E8
  Maximum OsSinitTable version     = 0x4
  Minimum MLE Header version       = 0x20000
  Capabilities                     = 0x2
    Module supports MONITOR address RLP wakeup-method.
  ACM Version                      = 0x12
Chipset ID list table (0x4E8):
  Count                            = 0x1
  Chipset ID entry 0x0:
    Flags                          = 0x1
    Vendor ID                      = 0x8086
    Device ID                      = 0xC000
    Revision ID                    = 0x7
  Public key hash                  = 08 77 7B 21 EC 4D 7F CE F7 68
                                     2A 26 96 BC 5F 42 A9 96 45 A4
                                     21 81 10 7F 87 70 C2 24 37 FD
                                     E0 2C
ACM summary:
  Module type                      = 0x10002
  Chipset ID                       = 0x3400
  Flags                            = 0x4000
  Module vendor                    = 0x8086
  Module date                      = 0x20100114
  Chipset ACM Type                 = 0x0
  ACM digital signature is valid.
  ACM and Chipset Public key hashes are equal.
  ACM Chipset ID matches TXT Chipset ID.

Examined ACM can be used on current platform.

SERVER TXTINFO. START VERIFYING TXT MEMORY MAP. PLEASE REFER TO LT-SX BWG FOR MORE DETAILS
S_TXTINF. MEMORY MAP QUERY. ERROR. INTEL(R) TXT PUBLIC SPACE IS NOT MARKED RESERVED
S_TXTINF. MEMORY MAP QUERY. ERROR. INTEL(R) TXT PRIVATE SPACE IS NOT MARKED RESERVED
SERVER TXTINFO. END VERIFYING TXT MEMORY MAP

 Reading TXT PolicyCMOSEntry at index io port=0x70 data io port=0x71 access width=1 bitpos=4
index=0x2a
SERVER TXTINFO. TXT POLICY VALUE=0x57
SERVER TXTINFO. TYPE10 RECORD IS PRESENT AND TXT POLICY IS SET TO ENABLED


SERVER TXTINFO. VERIFYING THAT IF CPU SUPPORTS LT-SX AND TXT IS ENABLED IN MSR, LT.SPAD INDICATES
SUCCESSFUL PROCESSING BIOS ACM
SERVER TXTINFO. BIOS ACM PROCESSING COMPLETED SUCCESSFULLY. LT.SPAD(Offset 0xA0 in TXT public
space) UPPER DW[0x80000000] LOWER DW[0x1]
                                                LT.CRASH REGISTER-OFSET 0x30[0x0]
                                                        LT.ACMCODE UPPER DW OFFSET-0x32c[0x0]
LT.ACMCODE LOWER DW OFFSET-0x328[0x0]
SERVER TXTINFO. DONE VERIFYING BIOS ACM PROCESSING STATUS USING LT.SPAD

SERVER TXTINFO. START: VERIFYING THAT CONFIGURATION HAS BEEN LOCKED BY BIOS USING
GETSEC[ENTERACCS][LOCKCONFIG]
```

```
SERVER TXTINFO. TBG-BXB chipset present at PCI BUS[0] LTLOCK[0] is set
SERVER TXTINFO. ALL TBG CHIPSETS HAVE BEEN MEMORY LOCKED. GETSEC[ENTERACCS:LOCKCONFIG] HAS BEEN
EXECUTED SUCCESSFULLY
SERVER TXTINFO. END: VERIFYING THAT CONFIGURATION HAS BEEN LOCKED USING
GETSEC[ENTERACCS][LOCKCONFIG]
```