APPLICATION NOTE

System Design Manufacturing Recommendations for Atmel TPM Devices

AT97SC3204 and AT97SC3205

Introduction

This application note covers system design considerations for designs utilizing the Atmel Trusted Platform Module (TPM) as they relate to system manufacturing. Included are explanations of the TPM operational state and default permanent flag settings as shipped by Atmel.

A suggested system manufacturing TPM setup sequence is provided with an explanation as to why each particular setup command is recommended. These TPM setup explanations are provided to aid the system designer/manufacturer in customizing the TPM setup sequence for the target application and usage/security model.

Additional considerations will need to be addressed when configuring FIPS mode devices. Some of those considerations are addressed in this document. Please consult the Atmel Configuring FIPS/Flexible Application Note for more comprehensive information.

Ultimately, it is up to the system designer/manufacturer developing and manufacturing a product containing the TPM to determine which TPM configuration settings during manufacturing are best for the target application and usage/security model. Please accept these recommendations as suggestions for a standard method of handling manufacturing configuration for the Atmel TPM, and adjust the custom configuration as needed.

1. Atmel TPM Shipping State

When a TPM is shipped by Atmel, they may be shipped in what is considered “Compliance” mode or “Real” mode.

Note: Due to the setup of most TPM-Aware BIOS systems, the execution of physical presence commands may not be available and will be controlled by the system (typically the BIOS).

1.1 Compliance Mode Operational State

Devices shipped in the Compliance mode operational state allows the manufacturer the option of performing known answer testing on the TPM if desired. While in Compliance mode, the TPM will respond with known answers to certain functions as defined in the Trusted Computing Group (TCG) TPM Compliance specification.

Example: While in Compliance mode (before TPM_ForceClear), if the TPM_CreateEndorsementKeyPair command is sent to the TPM, the TPM will ***not*** generate an Endorsement Key, and instead remain in Compliance mode responding with the following public Compliance EK:

KEY A 256 Byte Public Modulus is returned as the EK in Compliance Mode

```
AB 56 7C 0E 60 8C 5C 18 9E 90 2C 37 32 CF E3 FE 4F A7 B5 0C 78 A1 5D A7 39 EB C0
06 87 05 DB 1F E4 AB 2A 9A 68 E3 5B B6 FB 27 69 5A 4B E2 90 65 04 B2 78 CF 44 02
7C 16 4C FB F5 F0 F6 25 7D 31 F1 2E D8 67 93 5A 48 B2 C1 4C 16 FD 97 E5 86 65 4A
2E 07 4B 14 78 F7 66 83 66 05 B0 EA EC 1E 16 CF F9 F9 C5 5C BC 7B 42 24 A1 A7 1B
55 D7 4B B1 62 7F 90 88 EE FB FB 26 B1 4F 56 97 8C D0 12 05 A6 EF 09 C9 08 10 F2
1B 65 9C F2 05 7B CC 4E 6A 65 0C 1C E1 B5 3E 86 7D F8 0B 8B 6F E3 72 2B CB C9 3D
F8 61 F4 83 74 B1 38 A6 CE DE 18 7F 8D C4 8F A1 8E A6 AC 71 A4 89 60 D3 3E 5F 3D
18 5C 32 6C 96 1D 84 8B 50 C3 5B 68 5C 16 2D 9C BB F1 79 60 6E C9 25 AA EC 26 9E
9E D4 D6 89 F3 FF 23 AA 75 46 3B 4A EA 1D E5 03 B9 AC 6D F8 2D 88 FF 84 12 B8 47
CF 3A 32 C9 66 C6 E3 2C 1F 7D 30 D8 99
```

This KEY A 256 Byte Public Modulus is defined in Section 2.21.2 of the TCG TPM Compliance Specification.

Compliance mode is exited and Real Mode is entered the first time the TPM is cleared. Compliance mode is cleared by the manufacturer during system manufacturing, by executing a Clear Operation (Force Clear or an Owner Clear) on the TPM. Clearing Compliance mode is a one-time operation, once Compliance mode is exited, the TPM is permanently in Real Mode. After Compliance mode has been cleared and Real Mode has been entered, the manufacturer may then generate the permanent Endorsement Key Pair. Most manufacturers choose to generate the Endorsement Key Pair during system manufacturing. Regardless of when the Endorsement Key Pair is finally generated, it is recommended the system manufacturer should always clear Compliance mode before shipping their product containing the TPM.

The majority of system manufacturers do not take advantage of Compliance mode known answer testing, and choose to clear the compliance data from the TPM as the first step in their manufacturing process. In the event a system manufacturer would like to perform Compliance Mode system level functional testing, Atmel will supply Compliance mode test vectors upon request.

Atmel strongly recommends system manufacturers clear Compliance mode during the manufacturing process; *Not* doing so leaves the TPM vulnerable to various security risks. While in Compliance mode, all TPMs have the same endorsement key, which is a Compliance mode endorsement key available for known answer testing while the TPM is in Compliance mode. The *real* endorsement key is not created until after the TPM is cleared, either by the Force Clear or Owner Clear commands, followed later by the TPM_CreateEndorsementKeyPair command.

See Section 2., “Sample Manufacturing Sequence” for a suggested TPM command sequence which will clear Compliance mode from the TPM.

1.2 Permanent Flags Default State

Below lists the TPM Permanent Flags default state as the Atmel TPM is shipped in Compliance mode.

TPM Permanent Flags Default State

```
disable                                 : 00 [FALSE]
ownership                               : 01 [TRUE]
deactivated                             : 00 [FALSE]
readPubek                               : 01 [TRUE]
disableOwnerClear                       : 00 [FALSE]
allowMaintenance                        : 01 [TRUE]
physicalPresenceLifetimeLock            : 00 [FALSE]
physicalPresenceHWEnable                : 00 [FALSE]
physicalPresenceCMDEnable               : 01 [TRUE]
CEKPUsed                                : 00 [FALSE]
TPMpost                                 : 00 [FALSE]
TPMpostLock                             : 00 [FALSE]
FIPS                                    : 00 [FALSE]
operator                                : 00 [FALSE]
enableRevokeEK                          : 00 [FALSE]
nvLocked                                : 00 [FALSE]
readSRKPub                              : 00 [FALSE]
tpmEstablished                          : 00 [FALSE]
maintenanceDone                         : 00 [FALSE]
```

Five of these TPM Permanent Flags are related to the management of Physical Presence and the disabled/deactivated state. Since Atmel ships the TPM in Compliance mode, for added convenience, Atmel ships the TPM with these five flags in the following default states:

TPM Permanent Flags Default State: Physical Presence and Disable/Deactivated State

```
TPM_PERMANENT_FLAGS.physicalPresenceHWEnable      :[FALSE]
TPM_PERMANENT_FLAGS.physicalPresenceCMDEnable     :[TRUE]
TPM_PERMANENT_FLAGS.physicalPresenceLifetimeLock  :[FALSE]
TPM_PERMANENT_FLAGS.disable                       :[FALSE]
TPM_PERMANENT_FLAGS.deactivated                   :[FALSE]
```

Shipping the Atmel TPM with Physical Presence Software Command Enabled (physicalPresenceCMDEnable flag – *TRUE*), as well as, enabled and activated (disable and deactivated flags – *FALSE*), allows the manufacturer to perform known answer Compliance mode testing without being required to first alter the state of these flags.

The TPM_PERMANENT_FLAGS.physicalPresenceCMDEnable remains in a state of *TRUE* unless the manufacturer or TPM user chooses to change the state via the TSC_PhysicalPresence command. It is quite doubtful that a manufacturer or TPM user would ever choose to change the state of this flag, since it must be *TRUE* in order for Physical Presence to be asserted on an Atmel TPM.

As long as the TPM_PERMANENT_FLAGS.physicalPresenceLifetimeLock flag remains in its default state of *FALSE*, the state of the other two TPM Permanent Flags(TPM_PERMANENT_FLAGS.physicalPresenceHWEnable and TPM_PERMANENT_FLAGS.physicalPresenceCMDEnable) can be changed. Once the TPM_PERMANENT_FLAGS.physicalPresenceLifetimeLock flag is set to *TRUE*, the state of the other two TPM Permanent Flags(TPM_PERMANENT_FLAGS.physicalPresenceHWEnable and TPM_PERMANENT_FLAGS.physicalPresenceCMDEnable) cannot be modified during the lifetime of the TPM.

As a result, it is important during the manufacturing process, the physicalPresenceLifetimeLock flag is set to *TRUE* while the physicalPresenceCMDEnable flag is also *TRUE*, thereby, permanently enabling the ability to assert PhysicalPresence via the TSC_PhysicalPresence command.

The TPM_PERMANENT_FLAGS.disable and TPM_PERMANENT_FLAGS.deactivated will change to a state of *TRUE*, meaning disabled and deactivated, as a result of the Force Clear or Owner Clear operation that clears the Compliance mode from the TPM.

See Section 4., “TCG 1.2 TPM Physical Presence Management” for a detailed description of how these physical presence permanent flags are used to manage TPM Physical Presence, as well as, recommendations for how the management of the volatile stclear flags along with the enabled and activated states should be handled by a PC BIOS or low level embedded system initialization software.

2. Sample Manufacturing Sequence

According to the TCG specifications, there are many different ways the TPM can be configured; therefore, it is not possible to provide a single solution as to how to setup the TPM during manufacturing. The target application, security concerns and usage model of the TPM will ultimately determine how the TPM needs to be setup. Relevant TCG documentation is available at: https://www.trustedcomputinggroup.org.

The following sample command sequence is one way the TPM can be setup during manufacturing. This sample command sequence provides an example of how to clear Compliance mode, set the Permanent Flags, and generate the Endorsement Key Pair. Please refer to the TCG specifications for additional details and optional setup methods to ensure the best setup method is utilized for a custom target application.

If shipped in Compliance mode, Atmel recommends the following steps to place the TPM into a fully operational state.

Step 1 Physically boot the TPM by performing a normal hardware power-up or reset sequence.

Step 2 TPM_Startup(ST_CLEAR)

TPM_Startup(ST_CLEAR) initializes the TPM after a power-up or reset. After the TPM receives a hardware reset either by power-up or by the assertion of the reset pin, a TPM_Startup command needs to be sent to the TPM before the majority of TPM commands will be accepted by the TPM.

TPM_Startup(ST_CLEAR)
Incoming Operands and Sizes:

```
00 C1 00 00 00 0C 00 00 00 99 00 01
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 0C
ordinal             4 Bytes, Offset       6:  00 00 00 99
startupType         2 Bytes, Offset      10:  00 01
```

TPM_Startup(ST_CLEAR)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 0A
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

Step 3 TSC_PhysicalPresence

During the manufacturing process, use TSC_PhysicalPresence to set the following three TPM_PERMANENT_FLAGS:

TPM_PERMANENT_FLAGS.physicalPresenceHWEnable to *FALSE*
TPM_PERMANENT_FLAGS.physicalPresenceCMDEnable to *TRUE*
TPM_PERMANENT_FLAGS.physicalPresenceLifetimeLock to *TRUE*

The Atmel recommendation during the manufacturing process is to set in the persistent TPM_PERMANENT_FLAGS to:

physicalPresenceHWEnable to *FALSE*
physicalPresenceCMDEnable to *TRUE*
physicalPresenceLifetimeLock to *TRUE*

This will permanently disable hardware access to physicalPresence, permanently enable software command access to physicalPresence, and permanently set the physicalPresenceLifetimeLock flag so these TPM_PERMANENT_FLAGS cannot be changed during the lifetime of the TPM. Software command access to physicalPresence is turned on and then permanently locked so this software command capability to access physicalPresence is never inadvertently lost if the execution of rogue software attempts to set these flags to an undesired state.

TSC_PhysicalPresence
(Set Physical Presence Permanent Flags)
Incoming Operands and Sizes:

```
00 C1 00 00 00 0C 40 00 00 0A 02 A0
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 0C
ordinal             4 Bytes, Offset       6:  40 00 00 0A
physicalPresence    2 Bytes, Offset      10:  02 A0
```

TSC_PhysicalPresence
(Set Physical Presence Permanent Flags)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 0A
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

An optional addition to Step 3 would be to read the permanent flags out of the TPM via the TPM_GetCapability command and confirm the Physical Presence Permanent Flags are set as requested.

TPM_GetCapability
(Read Permanent Flags)
Incoming Operands and Sizes:

```
00 C1 00 00 00 16 00 00 00 65 00 00 00 04 00 00 00 04 00 00 01 08
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 16
ordinal             4 Bytes, Offset       6:  00 00 00 65
capArea             4 Bytes, Offset      10:  00 00 00 04
subCapSize          4 Bytes, Offset      14:  00 00 00 04
subCap              4 Bytes, Offset      18:  00 00 01 08
```

TPM_GetCapability
(Read Permanent Flags)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 23 00 00 00 00 00 00 00 15 00 07 00 01 00 01 00 01 01 00 01 00 00
00 00 00 00 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 23
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
respSize            4 Bytes, Offset      10:  00 00 00 15
resp               21 Bytes, Offset      14:  00 07 00 01 00 01 00 01 0100 01 00
                                              00 00 00 00 00 00 00 00 00
```

Permanent Flag Settings:

```
disable                                 :  00 [FALSE]
ownership                               :  01 [TRUE]
deactivated                             :  00 [FALSE]
readPubek                               :  01 [TRUE]
disableOwnerClear                       :  00 [FALSE]
allowMaintenance                        :  01 [TRUE]
physicalPresenceLifetimeLock            :  01 [TRUE]
physicalPresenceHWEnable                :  00 [FALSE]
physicalPresenceCMDEnable               :  01 [TRUE]
CEKPUsed                                :  00 [FALSE]
TPMpost                                 :  00 [FALSE]
TPMpostLock                             :  00 [FALSE]
FIPS                                    :  00 [FALSE]
operator                                :  00 [FALSE]
enableRevokeEK                          :  00 [FALSE]
nvLocked                                :  00 [FALSE]
readSRKPub                              :  00 [FALSE]
tpmEstablished                          :  00 [FALSE]
maintenanceDone                         :  00 [FALSE]
```

Step 4 TSC_PhysicalPresence

When Physical Presence is needed, such as to execute the TPM_ForceClear command which follows, use TSC_PhysicalPresence to setTPM_STCLEAR_FLAGS.physicalPresence to *TRUE*.

TSC_PhysicalPresence
(Set Physical Presence STCLEAR Flag To *TRUE*)
Incoming Operands and Sizes:

```
00 C1 00 00 00 0C 40 00 00 0A 00 08
tag                2 Bytes, Offset       0:  00 C1
paramSize          4 Bytes, Offset       2:  00 00 00 0C
ordinal            4 Bytes, Offset       6:  40 00 00 0A
physicalPresence   2 Bytes, Offset      10:  00 08
```

TSC_PhysicalPresence
(Set Physical Presence STCLEAR Flag To *TRUE*)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                2 Bytes, Offset       0:  00 C4
paramSize          4 Bytes, Offset       2:  00 00 00 0A
returnCode         4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

An optional addition to Step 4 would be to read the STCLEAR flags out of the TPM via the TPM_GetCapability command and confirm the Physical Presence STCLEAR flag is set as requested.

TPM_GetCapability
(Read STCLEAR Flags)
Incoming Operands and Sizes:

```
00 C1 00 00 00 16 00 00 00 65 00 00 00 04 00 00 00 04 00 00 01 09
tag                2 Bytes, Offset       0:  00 C1
paramSize          4 Bytes, Offset       2:  00 00 00 16
ordinal            4 Bytes, Offset       6:  00 00 00 65
capArea            4 Bytes, Offset      10:  00 00 00 04
subCapSize         4 Bytes, Offset      14:  00 00 00 04
subCap             4 Bytes, Offset      18:  00 00 01 09
```

TPM_GetCapability
(Read STCLEAR Flags)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 15 00 00 00 00 00 00 00 07 00 20 00 00 01 00
tag                2 Bytes, Offset       0:  00 C4
paramSize          4 Bytes, Offset       2:  00 00 00 15
returnCode         4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
respSize           4 Bytes, Offset      10:  00 00 00 07
resp               7 Bytes, Offset      14:  00 20 00 00 01 00 00
```

STCLEAR Flags:

```
deactivated                              : 00 [FALSE]
disableForceClear                        : 00 [FALSE]
physicalPresence                         : 01 [TRUE]
physicalPresenceLock                     : 00 [FALSE]
bGlobalLock                              : 00 [FALSE]
```

Step 5 TPM_ForceClear

TPM_ForceClear will clear the compliance data from the TPM. Clearing compliance data from the TPM during manufacturing is recommended if compliance devices are purchased. This step is not necessary when initializing Real mode devices.

TPM_ForceClear
Incoming Operands and Sizes:

```
00 C1 00 00 00 0A 00 00 00 5D
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 0A
ordinal             4 Bytes, Offset       6:  00 00 00 5D
```

TPM_ForceClear
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 0A
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

Step 6 TPM_PhysicalEnable

Enable the TPM under Physical Presence so the Endorsement Key Pair can be created. Use TPM_PhysicalEnable to set TPM_PERMANENT_FLAGS.disable to *FALSE*.

TPM_PhysicalEnable
Incoming Operands and Sizes:

```
00 C1 00 00 00 0A 00 00 00 6F
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 0A
ordinal             4 Bytes, Offset       6:  00 00 00 6F
```

TPM_PhysicalEnable
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                 2 Bytes, Offset       0:   00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 0A
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

Step 7 TPM_PhysicalSetDeactivated (FALSE)

The TPM *must* be activated in order for the Endorsement Key Pair to be created. Although presently activated, deactivation of the TPM at the next reset is scheduled by the TPM_ForceClear command from Step 5. Cancel this deactivation request by using TPM_PhysicalSetDeactivated to set TPM_PERMANENT_FLAGS.deactivated to *FALSE*.

TPM_PhysicalSetDeactivated (FALSE)
Incoming Operands and Sizes:

```
00 C1 00 00 00 0B 00 00 00 72 00
tag                2 Bytes, Offset       0:  00 C1
paramSize          4 Bytes, Offset       2:  00 00 00 0B
ordinal            4 Bytes, Offset       6:  00 00 00 72
state              1 Bytes, Offset      10:  00
```

TPM_PhysicalSetDeactivated (FALSE)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                2 Bytes, Offset       0:  00 C4
paramSize          4 Bytes, Offset       2:  00 00 00 0A
returnCode         4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

Step 8 TPM_SelfTestFull

The LPC and SPI Standard Mode TPMs support a split SelfTest feature as defined by the TCG PC Client specification. The initial partial self-test is executed immediately upon power-up or reset and includes verification of SHA capabilities allowing early access to the SHA engine required for secure boot operations. The remaining tests of critical internal cryptographic resources are performed with the TPM_SelfTestFull command. If this command is not executed, any command requiring TPM resources will automatically complete testing of cryptographic resources and respond with the error code TPM_DOING_SELFTEST; the requesting software would then be required to resend the original command.

A TPM in FIPS Mode and the I^2C TPM execute the SelfTest completely on power-up before any commands are processed; therefore, this command is not necessary.

TPM_SelfTestFull
Incoming Operands and Sizes:

```
00 C1 00 00 00 0A 00 00 00 50
tag                2 Bytes, Offset       0:  00 C1
paramSize          4 Bytes, Offset       2:  00 00 00 0A
ordinal            4 Bytes, Offset       6:  00 00 00 50
```

TPM_SelfTestFull
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                2 Bytes, Offset       0:  00 C4
paramSize          4 Bytes, Offset       2:  00 00 00 0A
returnCode         4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

Step 9 TPM_CreateEndorsementKeyPair

TPM_CreateEndorsementKeyPair will create the Endorsement Key Pair. This command can only be successfully executed one time. If an EK already exists, the TPM will respond with a return code of TPM_DISABLED_CMD.

TPM_CreateEndorsementKeyPair
Incoming Operands and Sizes:

```
00 C1 00 00 00 36 00 00 00 78 E6 59 0F A4 B0 C0 E9 65 39 27 CF 76 60 6E F9 95 CE 35 E3 13
00 00 00 01 00 03 00 01 00 00 00 0C 00 00 08 00 00 00 00 02 00 00 00 00
tag                                                 2 Bytes,Offset 0:00 C1
paramSize                                           4 Bytes,Offset 2:00 00 00 36
ordinal                                             4 Bytes,Offset 6:00 00 00 78
antiReplay.nonce                                    20 Bytes,Offset 10:E6 59 0F A4 B0 C0
E9 65 39 27 CF 76 60 6E F9 95 CE 35 E3 13
keyinfo.algorithmID                                 4 Bytes,Offset 30:00 00 00 01
keyinfo.encScheme                                   2 Bytes,Offset 34:00 03
keyinfo.sigScheme                                   2 Bytes,Offset 36:00 01
keyinfo.parmsize                                    4 Bytes,Offset 38:00 00 00 0C
keyinfo.parms.keylength                             4 Bytes,Offset 42:00 00 08 00
keyinfo.parms.numprimes                             4 Bytes,Offset 46:00 00 00
02keyinfo.parms.exponentsize                        4 Bytes,Offset 50:00 00 00
00keyinfo.parms.exponent                            0 Bytes,Offset 54:
```

TPM_CreateEndorsementKeyPair
Outgoing Operands and Sizes:

```
00 C4 00 00 01 3A 00 00 00 00 00 00 00 01 00 03 00 01 00 00 00 0C 00 00 08 00 00 00 00 02
00 00 00 00 00 00 01 00 C0 0A 98 C1 04 E6 AA 33 E4 0B 21 DA C4 3F B1 FB 2F 7F E7 15 CA 59
D6 90 E8 91 2F 3D 67 1F 82 DD 17 03 DA 43 BD C6 B4 43 9E 62 30 AA 56 DD 83 CB F3 EC A4 67
21 8F 79 15 B8 2D D6 CA E3 53 CE 9E B9 C0 2D F4 30 29 BD E6 85 A2 95 F1 61 CA ED 16 A2 F2
8B 4E 92 00 E3 9F 83 76 F5 F6 AF 9A 04 95 83 33 51 AF 51 AA 66 54 84 04 64 43 4E B0 8A 4E
3D 63 8F 5A 2A 53 FE 19 10 33 DA 22 7A 38 9C B1 20 EA 66 41 4A 2A 00 DE BB E1 BA 10 36 75
11 20 25 06 22 7A F1 E6 4C A5 77 D7 74 7D E3 DD 27 75 AF D4 04 04 E4 19 A8 A3 04 0D CD 9B
8A 10 4B 7D 1D B5 9D B4 30 D5 86 4B 1B 9E 3D 11 83 E8 3E D4 31 F3 71 69 3F F8 DA B6 40 DB
45 04 EC E7 E4 70 3C A3 D9 26 A2 CA D9 0F 5B 13 5B 9B 70 E9 8F 94 5A 85 D4 44 E4 C8 94 17
77 1C 20 A3 D5 E6 14 6B 21 DB 21 02 B6 A1 1E 77 F1 07 B0 B9 14 B1 33 8F 49 69 13 81 83 0E
94 57 92 66 80 7E DD F5 B4 42 3B 19 CB 29
tag                                                  2 Bytes,Offset 0:00 C4
paramSize                                            4 Bytes,Offset 2:00 00 01 3A
returnCode                                           4 Bytes,Offset 6:00 00 00 00 [TPM_SUCCESS]
pubEndorsementKey.algorithmparms.algorithmID         4 Bytes,Offset 10:00 00 00 01
pubEndorsementKey.algorithmparms.encScheme           2 Bytes,Offset 14:00 03
pubEndorsementKey.algorithmparms.sigScheme           2 Bytes,Offset 16:00 01
pubEndorsementKey.algorithmparms.parmsize            4 Bytes,Offset 18:00 00 00 0C
pubEndorsementKey.algorithmparms.parms.keylength     4 Bytes,Offset 22:00 00 08 00
pubEndorsementKey.algorithmparms.parms.numprimes     4 Bytes,Offset 26:00 00 00 02
pubEndorsementKey.algorithmparms.parms.exponentsize4 Bytes,Offset 30:00 00 00 00
pubEndorsementKey.algorithmparms.parms.exponent      0 Bytes,Offset 34:
pubEndorsementKey.pubkey.keylength                   4 Bytes,Offset 34:00 00 01 00
```

```
pubEndorsementKey.pubkey                          256 Bytes,Offset 38:C0 0A 98C1 04 E6 AA
33 E4 0B 21 DA C4 3F B1 FB 2F 7F E7 15 CA 59 D6 90 E8 91 2F 3D 67 1F 82 DD 17 03 DA 43 BD
C6 B4 43 9E 62 30 AA 56 DD 83 CB F3 EC A4 67 21 8F 79 15 B8 2D D6 CA E3 53 CE 9E B9 C0 2D
F4 30 29 BD E6 85 A2 95 F1 61 CA ED 16 A2 F2 8B 4E 92 00 E3 9F 83 76 F5 F6 AF 9A 04 95 83
33 51 AF 51 AA 66 54 84 04 64 43 4E B0 8A 4E 3D 63 8F 5A 2A 53 FE 19 10 33 DA 22 7A 38 9C
B1 20 EA 66 41 4A 2A 00 DE BB E1 BA 10 36 75 11 20 25 06 22 7A F1 E6 4C A5 77 D7 74 7D E3
DD 27 75 AF D4 04 04 E4 19 A8 A3 04 0D CD 9B 8A 10 4B 7D 1D B5 9D B4 30 D5 86 4B 1B 9E 3D
11 83 E8 3E D4 31 F3 71 69 3F F8 DA B6 40 DB 45 04 EC E7 E4 70 3C A3 D9 26 A2 CA D9 0F 5B
13 5B 9B 70 E9 8F 94 5A 85 D4 44 E4 C8 94 17 77 1C 20 A3 D5 E6 14 6B 21 DB 21 02 B6 A1 1E
77 F1 07 B0 B9 14 B1 33 8F
checksum.digest                                   20 Bytes,Offset 294: 49 69 13 81 83
0E 94 57 92 66 80 7E DD F5 B4 42 3B 19 CB 29
```

Step 10 TPM_PhysicalSetDeactivated (*TRUE*)

Deactivate the TPM under Physical Presence. Use TPM_PhysicalSetDeactivated to set TPM_PERMANENT_FLAGS.deactivated to *TRUE* which requests deactivation of the TPM at the next reset. Step 10 is normally seen in a PC product since PC products are normally shipped in the deactivated state. In an embedded non-PC design, this command is optional depending on the application and security requirements.

TPM_PhysicalSetDeactivated (*TRUE*)
Incoming Operands and Sizes:

```
00 C1 00 00 00 0B 00 00 00 72 01
tag                2 Bytes, Offset       0:  00 C1
paramsize          4 Bytes, Offset       2:  00 00 00 0B
ordinal            4 Bytes, Offset       6:  00 00 00 72
state              1 Bytes, Offset      10:  01
```

TPM_PhysicalSetDeactivated (*TRUE*)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                2 Bytes, Offset       0:  00 C4
paramsize          4 Bytes, Offset       2:  00 00 00 0A
returncode         4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

Step 11 TPM_PhysicalDisable

Disable the TPM under Physical Presence. Use TPM_PhysicalDisable to set TPM_PERMANENT_FLAGS.disable to *TRUE*. Step 10 is normally seen in a PC product since PC products are normally shipped in the disabled state. In an embedded non-PC design, this command is optional depending on the application and security requirements.

TPM_PhysicalDisable
Incoming Operands and Sizes:

```
00 C1 00 00 00 0A 00 00 00 70
tag                2 Bytes, Offset       0:  00 C1
paramsize          4 Bytes, Offset       2:  00 00 00 0A
ordinal            4 Bytes, Offset       6:  00 00 00 70
```

TPM_PhysicalDisable
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                2 Bytes, Offset       0:  00 C4
paramsize          4 Bytes, Offset       2:  00 00 00 0A
returncode         4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

Step 12 TSC_PhysicalPresence

After commands that require Physical Presence are no longer needed, and a system reboot is not imminent, it is good practice to turn off Physical Presence. Use TSC_PhysicalPresence to set TPM_STCLEAR_FLAGS.physicalPresence to *FALSE*. The TPM_STCLEAR_FLAGS.physicalPresence will automatically be set to *FALSE* by the next execution of the TPM_Startup(ST_CLEAR) command, which normally occurs after a system reboot. Step 12 is usually not necessary during manufacturing setup since the TPM will normally be rebooted before use.

TSC_PhysicalPresence
(Set physical presence STCLEAR flag to *FALSE*)
Incoming operands and sizes:

```
00 C1 00 00 00 0C 40 00 00 0A 00 10
tag                2 Bytes, Offset       0:  00 C1
paramSize          4 Bytes, Offset       2:  00 00 00 0C
ordinal            4 Bytes, Offset       6:  40 00 00 0A
physicalPresence   2 Bytes, Offset      10:  00 10
```

TSC_PhysicalPresence
(Set physical presence STCLEAR flag to *FALSE*)
Outgoing operands and sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                2 Bytes, Offset       0:  00 C4
paramSize          4 Bytes, Offset       2:  00 00 00 0A
returnCode         4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

An optional addition to Step 12 would be to read the STCLEAR flags out of the TPM via the TPM_GetCapability command and confirm the Physical Presence STCLEAR Flag is set as requested.

TPM_GetCapability(Read STCLEAR Flags)
Incoming Operands and Sizes:

```
00 C1 00 00 00 16 00 00 00 65 00 00 00 04 00 00 00 04 00 00 01 09
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 16
ordinal             4 Bytes, Offset       6:  00 00 00 65
capArea             4 Bytes, Offset      10:  00 00 00 04
subCapSize          4 Bytes, Offset      14:  00 00 00 04
subCap              4 Bytes, Offset      18:  00 00 01 09
```

TPM_GetCapability(Read STCLEAR Flags)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 15 00 00 00 00 00 00 00 07 00 20 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 15
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
respSize            4 Bytes, Offset      10:  00 00 00 07
resp                7 Bytes, Offset      14:  00 20 00 00 00 00 00
```

STCLEAR Flags:

```
deactivated                              : 00 [FALSE]
disableForceClear                        : 00 [FALSE]
physicalPresence                         : 00 [FALSE]
physicalPresenceLock                     : 00 [FALSE]
bGlobalLock                              : 00 [FALSE]
```

Step 13 Locking NV

TPM_DefineSpace with a nv index of FF FF FF FF will set the permanent nvlocked flag. This prevents unauthorized use of user-accessible non-volatile memory. This step should be executed *only* during the end of the manufacturing process when unlimited access to the non-volatile memory is no longer needed for things such as insertion of platform certificates.

TPM_DefineSpace
Incoming Parameters

```
tag                                          2 Bytes, Offset  0: 00 C1
paramsize                                    4 Bytes, Offset  2: 00 00 00 65
ordinal                                      4 Bytes, Offset  6: 00 00 00 CC
pubinfo.tag                                  2 Bytes, Offset  10: 00 18
pubinfo.nvindex                              4 Bytes, Offset  12: FF FF FF FF
pubinfo.pcrinforead.pcrselection.sizeofselect 2 Bytes, Offset  16: 00 03
pubinfo.pcrinforead.pcrselection.pcrselect    3 Bytes, Offset 3:18: 00 00 00
pubinfo.pcrinforead.localityatrelease        1 Byte, Offset   21: 01
```

```
pubinfo.pcrinforead.digestatrelease.digest    20 Bytes, Offset 22: 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
pubinfo.pcrinfowrite.pcrselection.sizeofselect2 Bytes, Offset  42: 00 03
pubinfo.pcrinfowrite.pcrselection.pcrselect   3 Bytes, Offset  44: 00 00 00
pubinfo.pcrinfowrite.localityatrelease        1 Byte, Offset   047:01
pubinfo.pcrinfowrite.digestatrelease.digest   20 Bytes, Offset 048: 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
pubinfo.permission.tag                        2 Bytes, Offset  68: 00 17
pubinfo.permission.attributes                 4 Bytes, Offset  70: 00 02 00 00
pubinfo.breadstclear                          1 Byte, Offset   74: 00
pubinfo.bwritestclear                         1 Byte, Offset   75: 00
pubinfo.bwritedefine                          1 Byte, Offset   76: 00
pubinfo.datasize                              4 Bytes, Offset  77: 00 00 00 00
encauth.digest                                20 Bytes, Offset 81: 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
```

Outgoing Parameters

```
tag                                           2 Bytes, Offset  0: 00 C4
paramsize                                     4 Bytes, Offset  2: 00 00 00 0A
returncode                                    4 Bytes, Offset  6: 00 00 00 00 [TPM_SUCCESS]
```

At the conclusion of these 13 steps, the TPM will be in the following state:

```
Compliance Mode:                              Cleared
physicalPresenceHWEnable:                     False
physicalPresenceCMDEnable:                    True
physicalPresenceLifetimeLock:                 True
Endorsement Key Pair:                         Generated
Enable/Disable:                               Disabled
Activated/Deactivated:                        Activated until the next reset,
deactivated at the next power-up or reset is requested.
Physical Presence:                            Not Present
NV_Locked                                     True
```

This sample manufacturing sequence may be used as is, but is intended to be modified as needed by the system designer/manufacturer developing and manufacturing the product containing the TPM.

3. Real Mode Operational State

Some Atmel TPMs are shipped in Real mode operational states. A TPM shipped in Real mode has an EK generated by Atmel and is ready for normal operations. Because the generation of the EK is non-deterministic, it could take seconds to tens of seconds to generate an EK. Purchasing devices in Real mode simplifies the OEM/ODM/CM manufacturing process and reduces time required for device initialization.

If shipped in Real mode, Atmel recommends the following steps to place the TPM into a fully operational state.

Step 1 Physically boot the TPM by performing a normal hardware power-up or reset sequence.

Step 2 TPM_Startup(ST_CLEAR)

TPM_Startup(ST_CLEAR) initializes the TPM after a power-up or reset. After the TPM receives a hardware reset either by power-up or by the assertion of the reset pin, a TPM_Startup command needs to be sent to the TPM before the majority of TPM commands will be accepted by the TPM.

TPM_Startup(ST_CLEAR)
Incoming Operands and Sizes:

```
00 C1 00 00 00 0C 00 00 00 99 00 01
tag                2 Bytes, Offset       0:  00 C1
paramSize          4 Bytes, Offset       2:  00 00 00 0C
ordinal            4 Bytes, Offset       6:  00 00 00 99
startupType        2 Bytes, Offset      10:  00 01
```

TPM_Startup(ST_CLEAR)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                2 Bytes, Offset       0:  00 C4
paramSize          4 Bytes, Offset       2:  00 00 00 0A
returnCode         4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

Step 3 TSC_PhysicalPresence

During the manufacturing process, use TSC_PhysicalPresence to set the following three TPM_PERMANENT_FLAGS:

TPM_PERMANENT_FLAGS.physicalPresenceHWEnable to *FALSE*
TPM_PERMANENT_FLAGS.physicalPresenceCMDEnable to *TRUE*
TPM_PERMANENT_FLAGS.physicalPresenceLifetimeLock to *TRUE*

The Atmel recommendation during the manufacturing process is to set in the persistent TPM_PERMANENT_FLAGS to:

physicalPresenceHWEnable to *FALSE*
physicalPresenceCMDEnable to *TRUE*
physicalPresenceLifetimeLock to *TRUE*

This will permanently disable hardware access to physicalPresence, permanently enable software command access to physicalPresence, and permanently set the physicalPresenceLifetimeLock flag so these TPM_PERMANENT_FLAGS cannot be changed during the lifetime of the TPM. Software command access to physicalPresence is turned on and then permanently locked so this software command capability to access physicalPresence is never inadvertently lost if the execution of rogue software attempts to set these flags to an undesired state.

TSC_PhysicalPresence
(Set Physical Presence Permanent Flags)
Incoming Operands and Sizes:

```
00 C1 00 00 00 0C 40 00 00 0A 02 A0
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 0C
ordinal             4 Bytes, Offset       6:  40 00 00 0A
physicalPresence    2 Bytes, Offset      10:  02 A0
```

TSC_PhysicalPresence
(Set Physical Presence Permanent Flags)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 0A
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

An optional addition to Step 3 would be to read the permanent flags out of the TPM via the TPM_GetCapability command and confirm the Physical Presence Permanent Flags are set as requested.

TPM_GetCapability
(Read Permanent Flags)
Incoming Operands and Sizes:

```
00 C1 00 00 00 16 00 00 00 65 00 00 00 04 00 00 00 04 00 00 01 08
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 16
ordinal             4 Bytes, Offset       6:  00 00 00 65
capArea             4 Bytes, Offset      10:  00 00 00 04
subCapSize          4 Bytes, Offset      14:  00 00 00 04
subCap              4 Bytes, Offset      18:  00 00 01 08
```

TPM_GetCapability
(Read Permanent Flags)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 23 00 00 00 00 00 00 00 15 00 07 00 01 00 01 00 01 01 00 01 00 00
00 00 00 00 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 23
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
respSize            4 Bytes, Offset      10:  00 00 00 15
resp               21 Bytes, Offset      14:  00 07 00 01 00 01 00 01 0100 01 00
                                              00 00 00 00 00 00 00 00 00
```

Permanent Flag Settings:

```
disable                                 :  00 [FALSE]
ownership                               :  01 [TRUE]
deactivated                             :  00 [FALSE]
readPubek                               :  01 [TRUE]
disableOwnerClear                       :  00 [FALSE]
allowMaintenance                        :  01 [FALSE]
physicalPresenceLifetimeLock            :  01 [FALSE]
physicalPresenceHWEnable                :  00 [FALSE]
physicalPresenceCMDEnable               :  01 [TRUE]
CEKPUsed                                :  00 [FALSE]
TPMpost                                 :  00 [FALSE]
TPMpostLock                             :  00 [FALSE]
FIPS                                    :  00 [FALSE]
operator                                :  00 [FALSE]
enableRevokeEK                          :  00 [FALSE]
nvLocked                                :  00 [FALSE]
readSRKPub                              :  00 [FALSE]
tpmEstablished                          :  00 [FALSE]
maintenanceDone                         :  00 [FALSE]
```

Step 4 TPM_SelfTestFull

The LPC and SPI Standard Mode TPMs support a split SelfTest feature as defined by the TCG PC Client specification. The initial partial self-test is executed immediately upon power-up or reset and includes verification of SHA capabilities allowing early access to the SHA engine required for secure boot operations. The remaining tests of critical internal cryptographic resources are performed with the TPM_SelfTestFull command. If this command is not executed, any command requiring TPM resources will automatically complete testing of cryptographic resources and respond with the error code TPM_DOING_SELFTEST; the requesting software would then be required to resend the original command.

A TPM in FIPS Mode and the I^2C TPM execute the SelfTest completely on power-up before any commands are processed; therefore, this command is not necessary.

TPM_SelfTestFull
Incoming Operands and Sizes:

```
00 C1 00 00 00 0A 00 00 00 50
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 0A
ordinal             4 Bytes, Offset       6:  00 00 00 50
```

TPM_SelfTestFull
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 0A
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

Step 5 TPM_PhysicalDisable

Disable the TPM under Physical Presence. Use TPM_PhysicalDisable to set TPM_PERMANENT_FLAGS.disable to *TRUE*. Step 10 is normally seen in a PC product since PC products are normally shipped in the disabled state. In an embedded non-PC design, this command is optional depending on the application and security requirements.

TPM_PhysicalDisable
Incoming Operands and Sizes:

```
00 C1 00 00 00 0A 00 00 00 70
tag                 2 Bytes, Offset       0:  00 C1
paramsize           4 Bytes, Offset       2:  00 00 00 0A
ordinal             4 Bytes, Offset       6:  00 00 00 70
```

TPM_PhysicalDisable
Outgoing Operands and Sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramsize           4 Bytes, Offset       2:  00 00 00 0A
returncode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

Step 6 TSC_PhysicalPresence

After commands that require Physical Presence are no longer needed, and a system reboot is not imminent, it is good practice to turn off Physical Presence. Use TSC_PhysicalPresence to set TPM_STCLEAR_FLAGS.physicalPresence to *FALSE*. The TPM_STCLEAR_FLAGS.physicalPresence will automatically be set to *FALSE* by the next execution of the TPM_Startup(ST_CLEAR) command, which normally occurs after a system reboot. Step 12 is usually not necessary during manufacturing setup since the TPM will normally be rebooted before use.

TSC_PhysicalPresence
(Set physical presence STCLEAR flag to *FALSE*)
Incoming operands and sizes:

```
00 C1 00 00 00 0C 40 00 00 0A 00 10
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 0C
ordinal             4 Bytes, Offset       6:  40 00 00 0A
physicalPresence    2 Bytes, Offset      10:  00 10
```

TSC_PhysicalPresence
(Set physical presence STCLEAR flag to *FALSE*)
Outgoing operands and sizes:

```
00 C4 00 00 00 0A 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 0A
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
```

An optional addition to Step 12 would be to read the STCLEAR flags out of the TPM via the TPM_GetCapability command and confirm the Physical Presence STCLEAR Flag is set as requested.

TPM_GetCapability(Read STCLEAR Flags)
Incoming Operands and Sizes:

```
00 C1 00 00 00 16 00 00 00 65 00 00 00 04 00 00 00 04 00 00 01 09
tag                 2 Bytes, Offset       0:  00 C1
paramSize           4 Bytes, Offset       2:  00 00 00 16
ordinal             4 Bytes, Offset       6:  00 00 00 65
capArea             4 Bytes, Offset      10:  00 00 00 04
subCapSize          4 Bytes, Offset      14:  00 00 00 04
subCap              4 Bytes, Offset      18:  00 00 01 09
```

TPM_GetCapability(Read STCLEAR Flags)
Outgoing Operands and Sizes:

```
00 C4 00 00 00 15 00 00 00 00 00 00 00 07 00 20 00 00 00 00
tag                 2 Bytes, Offset       0:  00 C4
paramSize           4 Bytes, Offset       2:  00 00 00 15
returnCode          4 Bytes, Offset       6:  00 00 00 00 [TPM_SUCCESS]
respSize            4 Bytes, Offset      10:  00 00 00 07
resp                7 Bytes, Offset      14:  00 20 00 00 00 00 00
```

STCLEAR Flags:

```
deactivated                                 : 00 [FALSE]
disableForceClear                           : 00 [FALSE]
physicalPresence                            : 00 [FALSE]
physicalPresenceLock                        : 00 [FALSE]
bGlobalLock                                 : 00 [FALSE]
```

Step 7 Locking NV

TPM_DefineSpace with a nv index of FF FF FF FF will set the permanent nvlocked flag. This prevents unauthorized use of user-accessible non-volatile memory. This step should be executed *only* during the end of the manufacturing process when unlimited access to the non-volatile memory is no longer needed for things such as insertion of platform certificates.

TPM_DefineSpace
Incoming Parameters

```
tag                                         2 Bytes, Offset  0: 00 C1
paramsize                                   4 Bytes, Offset  2: 00 00 00 65
ordinal                                     4 Bytes, Offset  6: 00 00 00 CC
pubinfo.tag                                 2 Bytes, Offset  10: 00 18
pubinfo.nvindex                             4 Bytes, Offset  12: FF FF FF FF
pubinfo.pcrinforead.pcrselection.sizeofselect 2 Bytes, Offset  16: 00 03
pubinfo.pcrinforead.pcrselection.pcrselect  3 Bytes, Offset 3:18: 00 00 00
pubinfo.pcrinforead.localityatrelease       1 Byte, Offset   21: 01
pubinfo.pcrinforead.digestatrelease.digest  20 Bytes, Offset 22: 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
pubinfo.pcrinfowrite.pcrselection.sizeofselect2 Bytes, Offset  42: 00 03
pubinfo.pcrinfowrite.pcrselection.pcrselect  3 Bytes, Offset  44: 00 00 00
pubinfo.pcrinfowrite.localityatrelease      1 Byte, Offset   047:01
pubinfo.pcrinfowrite.digestatrelease.digest  20 Bytes, Offset 048: 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
pubinfo.permission.tag                      2 Bytes, Offset  68: 00 17
pubinfo.permission.attributes               4 Bytes, Offset  70: 00 02 00 00
pubinfo.breadstclear                        1 Byte, Offset   74: 00
pubinfo.bwritestclear                       1 Byte, Offset   75: 00
pubinfo.bwritedefine                        1 Byte, Offset   76: 00
pubinfo.datasize                            4 Bytes, Offset  77: 00 00 00 00
encauth.digest                              20 Bytes, Offset 81: 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
```

Outgoing Parameters

```
tag                                         2 Bytes, Offset  0: 00 C4
paramsize                                   4 Bytes, Offset  2: 00 00 00 0A
returncode                                  4 Bytes, Offset  6: 00 00 00 00 [TPM_SUCCESS]
```

At the conclusion of these 7 steps, the TPM will be in the following state:

```
Compliance Mode:                          Cleared
physicalPresenceHWEnable:                 False
physicalPresenceCMDEnable:                True
physicalPresenceLifetimeLock:             True
Endorsement Key Pair:                     Generated
Enable/Disable:                           Disabled
Activated/Deactivated:                    Activated until the next reset,
deactivated at the next power-up or reset is requested.
Physical Presence:                        Not Present
NV_Locked                                 True
```

This sample manufacturing sequence may be used as is, but is intended to be modified as needed by the system designer/manufacturer developing and manufacturing the product containing the TPM.

3.1 Real Mode Operational State with Signed EK Certificate

Atmel offers Real Mode TPMs shipped with pre-generated EKs and X.509 certificates stored in Non-volatile memory. The EK certificate is user accessible and useful in environments which require a certificate chain.

The Real mode operational state with Signed EK Certificate would follow the same recommended steps as in Section 3., “Real Mode Operational State”.

4. TCG 1.2 TPM Physical Presence Management

4.1 TPM_PERMANENT_FLAGS

In the persistent TPM_PERMANENT_FLAGS, the Atmel recommendation to all platform manufacturers during their manufacturing process is to set:

physicalPresenceHWEnable to *FALSE* unless hardware physicalPresense will be utilized.
physicalPresenceCMDEnable to *TRUE.*
physicalPresenceLifetimeLock to *TRUE.*

This will permanently:

- Disable hardware access to physicalPresence unless physicalPresenceHWEnable was set to *TRUE.*
- Enable software command access to physicalPresence.
- Set the physicalPresenceLifetimeLock flag.

As a result, these TPM_PERMANENT_FLAGS cannot be changed during the lifetime of the TPM.

Software command access to physicalPresence is turned on and then permanently locked so this software command capability to access physicalPresence is never inadvertently lost if the execution of rogue software attempts to set these flags to an undesired state.

4.2 TPM_ STCLEAR_FLAGS

For the PC platform manufacturers, Atmel recommends the PC BIOS, after processing any TPM commands that require the use of physicalPresence, set physicalPresenceLock to *TRUE*, and physicalPresence to *FALSE* in the volatile TPM_STCLEAR_FLAGS before turning control over to Operating System or Application Software. This will prevent a potential attack by a hacker in which the hacker could assert physicalPresence and take control of the TPM as if they were the owner. These TPM_STCLEAR_FLAGS are reset to their default state upon the execution of the TPM_Startup(ST_CLEAR) command, which normally occurs after a system reboot, and can only occur after the TPM receives a hardware reset either by power-up or by the assertion of the reset pin. Implementation directions for Physical Presence in PC BIOS code are given in the "TCG PC Client Specific Implementation Specification For Conventional BIOS Version 1.20, Section 15 Physical Presence".

Typically, Production PC BIOS set physicalPresenceLock is set to *TRUE* and physicalPresence to *FALSE* in the volatile TPM_STCLEAR_FLAGS before turning control over to the operating system or application software, thereby disabling access to Physical Presence until the next boot cycle.

For embedded non-PC platform manufacturers, the decision to set physicalPresenceLock to *TRUE* and physicalPresence to *FALSE* in the volatile TPM_STCLEAR_FLAGS during system software initialization will depend on the application's TPM usage model and desired security controls.

4.3 Enabled and Activated Considerations

In addition to the recommendation that a PC BIOS should disable Physical Presence before turning control over to the operating system or application software, it is important that a PC BIOS allow a user the option to enable and activate a TPM while under BIOS control with Physical Presence enabled. This is due to the fact that when the owner is cleared from a TPM, according to the TCG specification, the TPM is immediately disabled and deactivation is scheduled for the next TPM reset at the time the owner is cleared. Since the TPM can only be enabled and activated under Physical Presence if the BIOS always disables Physical Presence before turning control over to operating system or application software typically and does not provide a mechanism to enable and activate the TPM while under BIOS control with Physical Presence enabled, the TPM will be effectively unusable in the event the first owner is cleared by the operating system or application software.

Typically, Production PC BIOS provides the following user selectable functionality under Physical Presence at user request during the PC boot cycle:

- Enable the TPM (Enabled and Activated)
- Disable the TPM (Disabled and Deactivated)
- Clear the TPM (Clear the TPM Owner)

For embedded non-PC applications, the decision of how to manage Physical Presence including the settings of the persistent and volatile flags which includes the control of the TPM Enabled and Activated State, will depend on the application's TPM usage model and desired security controls.

4.4 TPM_PERMANENT_FLAGS and TPM_STCLEAR_FLAGS Definitions

The TPM_PERMANENT_FLAGS and TPM_STCLEAR_FLAGS related to physicalPresence are defined in the "TPM Main Part 2 TPM Structures Specification Version 1.2".

TPM_PERMANENT_FLAGS defined in Section 7.1, "TPM_PERMANENT_FLAGS"

```
typedef struct tdTPM_PERMANENT_FLAGS{
  TPM_STRUCTURE_TAG tag;
  BOOL disable;
  BOOL ownership;
  BOOL deactivated;
  BOOL readPubek;
  BOOL disableOwnerClear;
  BOOL allowMaintenance;
  BOOL physicalPresenceLifetimeLock;
  BOOL physicalPresenceHWEnable;
  BOOL physicalPresenceCMDEnable;
  BOOL CEKPUsed;
  BOOL TPMpost;
  BOOL TPMpostLock;
  BOOL FIPS;
  BOOL operator;
  BOOL enableRevokeEK;
  BOOL nvLocked;
  BOOL readSRKPub;
  BOOL tpmEstablished;
  BOOL maintenanceDone;
} TPM_PERMANENT_FLAGS;
```

TPM_STCLEAR_FLAGS defined in Section 7.2, “TPM_STCLEAR_FLAGS”

```
typedef struct tdTPM_STCLEAR_FLAGS{
  TPM_STRUCTURE_TAG tag;
  BOOL deactivated;
  BOOL disableForceClear;
  BOOL physicalPresence;
  BOOL physicalPresenceLock;
  BOOL bGlobalLock;
} TPM_STCLEAR_FLAGS;
```

Mask bits used to set the state of these flags with the TSC_PhysicalPresence command defined in Section 4.9, “TPM_PHYSICAL_PRESENCE”

```
TPM_PHYSICAL_PRESENCE_HW_DISABLE
0x0200h sets the physicalPresenceHWEnable to FALSE
TPM_PHYSICAL_PRESENCE_CMD_DISABLE
0x0100h Sets the physicalPresenceCMDEnable to FALSE
TPM_PHYSICAL_PRESENCE_LIFETIME_LOCK
0x0080h sets the physicalPresenceLifetimeLock to TRUE
TPM_PHYSICAL_PRESENCE_HW_ENABLE
0x0040h Sets the physicalPresenceHWEnable to TRUE
TPM_PHYSICAL_PRESENCE_CMD_ENABLE
0x0020h Sets the physicalPresenceCMDEnable to TRUE
TPM_PHYSICAL_PRESENCE_NOTPRESENT
0x0010h Sets PhysicalPresence = FALSE
TPM_PHYSICAL_PRESENCE_PRESENT
0x0008h Sets PhysicalPresence = TRUE
TPM_PHYSICAL_PRESENCE_LOCK
0x0004h Sets PhysicalPresenceLock = TRUE
```

All specifications referenced in this document are freely downloadable at:
https://www.trustedcomputinggroup.org/

5. Conclusion

The proper management of TPM Physical Presence can enable powerful security features in an end customer's application. Likewise, improper management of TPM Physical Presence can create usability problems and allow potential successful security attacks in the same end customer's application.

As a result, it is very important that all platform manufacturers utilizing the TPM in their products determine the appropriate management of TPM Physical Presence for their desired TPM application and set or reset the Physical Presence persistent and volatile flags accordingly during platform manufacturing and system software initialization.

6. Revision History

Revision	Date	Description
8882A	01/2014	Initial document release.

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**