# Protocols TCG2 Test

## TCG2 Protocol Test

Reference Document:

*EFI Protocol Specification*, EFI\_TCG2\_PROTOCOL Chapter 6

Test in this chapter support TCG PC Client Platform TPM Profile Specification for TPM 2.0. v.103

### GetCapability()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
|  | 0xca93b02a, 0xe897, 0x4400, 0x81, 0x38, 0xc8, 0xa8, 0xcb, 0x2f, 0xc1, 0xed | EFI\_TCG2\_PROTOCOL. GetCapability() - GetCapabilty()  returns EFI\_INVALID\_PARAMTER with NULL pointer Capability Struct Passed in. | Call GetCapability() with NULL for capability struct pointer.  a. Verify Status returned == EFI\_INVALID\_PARAMETER. |
|  | 0xda8821d9, 0x3d2c, 0x4698, 0x8c, 0xd5, 0x0f, 0x0c, 0x82, 0x94, 0x1d, 0x0c | EFI\_TCG2\_PROTOCOL.GetCapability() –GetCapability() shall populate the included structure elements and return with a Status of EFI\_BUFFER\_TOO\_SMALL when structure size is set to less than the size of EFI\_TCG\_BOOT\_SERVICE\_CAPABILITY. | Invoke GetCapability() with ProtocolCapability.Size set to only include StructureVersion and ProtocolVersion.  a. Verify that StructureVersion->Major == 1  b. Verify that StructureVersion->Minor == 1  c. Verify that ProtocolVersion->Major == 1  d. .Verify that ProtocolVersion->Minor == 1  e. Verify Status returned == EFI\_BUFFER\_TOO\_SMALL.  f. Verify returned Size equal to size of the EFI\_TCG2\_BOOT\_SERVICE\_CAPABILITY up to and including the vendor ID field. |
|  | 0xfdee7001, 0x7e28, 0x4e35, 0x99, 0x66, 0x98, 0x0b, 0xeb, 0xba, 0xf1, 0x57 | EFI\_TCG2\_PROTOCOL. GetCapability() - GetCapability() shall populate all structure elements and return with a Status of EFI\_SUCCESS when the structure size includes all of the EFI\_TCG\_BOOT\_SERVICE\_CAPABILITY structure. | Invoke GetCapabilty() with Protocol Capability. Size set to sizeof (EFI\_TCG2\_BOOT\_SEVICE\_CAPABILTY)  a. Verify that StructureVersion->Major == 1  b. Verify that StructureVersion->Minor == 1  c. Verify that ProtocolVersion->Major == 1  d. . Verify that ProtocolVersion->Minor == 1  e. Verify that HashAlgorithmBitmap includes SHA256.  f. Verify that SupportedEventLogs is EFI\_TCG2\_EVENT\_LOG\_FORMAT\_TCG\_2  g. Verify that ~ActivePcrBanks & HashAlgorithmBitMap == 0  h. Verify that NumberofPcrBanks is at least 1  i. Verify that ActivePcrBanks includes SHA256, SHA384, or SHA512  j. Verify returned Status == EFI\_SUCCESS. |

### GetActivePcrBanks()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
|  | 0x7a1e79a3, 0x4064, 0x4372, 0xbb, 0x64,0x55, 0xb8, 0xf2, 0xa5, 0xa3, 0x26 | EFI\_TCG2\_PROTOCOL. GetActivePcrBanks() - GetActivePcrBanks() returns EFI\_INVALID\_PARAMETER with NULL pointer Passed in. | Invoke GetActivePcrBanks() with ActivePcrBanks=NULL.  a. Verify Status returned == EFI\_INVALID\_PARAMETER. |
|  | 0xb0e717c4, 0xb1e2, 0x49f7, 0xb2, 0xd7,0x60, 0x58,0x97, 0x7d, 0x09, 0x2c | EFI\_TCG2\_PROTOCOL. GetActivePcrBanks() - GetActivePcrBanks() should return with EFI\_SUCCESS and have SHA256/384/512 Algorithms in its Bitmap. | 1. Invoke GetActivePcrBanks() with valid ActivePcrBanks buffer. Should return EFI\_SUCCESS.  a. Verify Status returned == EFI\_SUCCESS  b. Verify that returned ActivePcrBanks bitmap includes SHA256, SHA384, or SHA512.  c. Verify that returned ActivePcrBanks bitmap matches one returned by GetCapabilty().  . |

### 30.1.3 HashLogExtendEvent()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
|  | 0xa8e1b5e6, 0xfc09, 0x461c, 0xb0, 0xe9, 0x2a, 0x49, 0xcd, 0x25, 0xc1, 0x24 | EFI\_TCG2\_PROTOCOL. HashLogExtendEvent() - HashLogExtendEvent()Test with NULL DataToHash Pointer should return EFI\_INVALID\_PARAMETER. | Invoke HashLogExtendEvent() with DataToHash=NULL.  a. Verify Status returned == EFI\_INVALID\_PARAMETER. |
|  | 0x26f04a9b, 0x7b7a, 0x4f47, 0xbe, 0xa8, 0xb1, 0xa6, 0x02, 0x65, 0x19, 0x8a | EFI\_TCG2\_PROTOCOL. HashLogExtendEvent() - HashLogExtendEvent()Test with NULL EfiTcgEvent Pointer should return EFI\_INVALID\_PARAMETER. | Invoke HashLogExtendEvent() with EfiTcgEvent=NULL.  a. Verify Status returned == EFI\_INVALID\_PARAMETER. |
|  | 0x4d1d9985, 0x91e2, 0x4948, 0x89, 0x16, 0xbb, 0x98, 0x13, 0x62, 0x39, 0x1d | EFI\_TCG2\_PROTOCOL. HashLogExtendEvent() - HashLogExtendEvent()Test with Event.Size < Event.Header.HeaderSize + sizeof(UINT32) should return EFI\_INVALID\_PARAMETER. | Invoke HashLogExtendEvent() with EfiTcgEvent.Size is less than EfiTcgEvent.Header.HeaderSize + sizeof(UINT32).  a. Verify Status returned == EFI\_INVALID\_PARAMETER. |
|  | 0xfb59cab7, 0x4f8c, 0x4ded, 0xa4, 0x1c, 0xc8, 0x41, 0x20, 0x1c, 0x37, 0x22 | EFI\_TCG2\_PROTOCOL. HashLogExtendEvent() - HashLogExtendEvent()Test with PCRIndex > 23 should return EFI\_INVALID\_PARAMETER. | Invoke HashLogExtendEvent with EfiTcgEvent.Header.PCRIndex=24.  a. Verify Status returned == EFI\_INVALID\_PARAMETER. |
|  | 0x0363d22f, 0xc66a, 0x4872, 0xa5, 0x46, 0x06, 0x7f, 0x6a, 0x0d, 0xdb, 0xcd | EFI\_TCG2\_PROTOCOL. HashLogExtendEvent() - HashLogExtendEvent() Test with valid parameters should return EFI\_SUCCESS. | Invoke HashLogExtendEvent() with:  i. DataToHash = "The quick brown fox jumps over the lazy dog"  ii. PCRIndex = 16  iii. EventType = EV\_POST\_CODE  iv. Event data = “TCG2 Protocol Test”  a. Verify Status returned == EFI\_SUCCESS. |
|  | 0x9cd6d636, 0x603a, 0x4b78, 0x80, 0xa3, 0xa3, 0xb9, 0xcc, 0x6a, 0x0b, 0x08 | EFI\_TCG2\_PROTOCOL. HashLogExtendEvent() - HashLogExtendEvent()Test Handling of PE\_COFF\_IMAGE flag. | Invoke HashLogExtendEvent() with:  i. DataToHash = "The quick brown fox jumps over the lazy dog"  ii. PCRIndex = 16  iii. EventType = EV\_POST\_CODE  iv. Event data = “TCG2 Protocol Test”  v. Flags = PE\_COFF\_IMAGE  a. Verify Status returned == EFI\_UNSUPPORTED. |

### 30.1.4 GetEventLog()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
|  | 0xfc80408e, 0x9a3c, 0x4054, 0x96, 0xf9, 0x31, 0x23, 0x35, 0xc2, 0x31, 0x35 | EFI\_TCG2\_PROTOCOL. GetEventLog() - GetEventLog() should return EFI\_INVALID\_PARAMTER when passed in invalid EventLog Format. | Invoke GetEventLog() with invalid EventLogFormat.  a. Verify Status returned == EFI\_SUCCESS. |
|  | 0x45fa1a42, 0x912a, 0x5124, 0x84, 0xf4, 0x41, 0x67, 0xab, 0xb5, 0x89, 0x90 | EFI\_TCG2\_PROTOCOL. GetEventLog() - GetEventLog() shall return EFI\_SUCCESS when a valid EventLogFormat is passed in. | Invoke GetEventLog() with EventLogFormat=EFI\_TCG2\_EVENT\_LOG\_FORMAT\_TCG\_2. Should return EFI\_SUCCESS.  a. Verify Status returned == EFI\_SUCCESS. |
|  | 0x1689bc3a, 0x2298, 0xa116, 0x28, 0x4c, 0xc1, 0xdd, 0xaa, 0xd8, 0xef, 0x51 | EFI\_TCG2\_PROTOCOL. GetEventLog() - GetEventLog() should return correct EventLogHeader | Verify that the returned event log is present at EventLogLocation address by verifying event log header. |
|  | 0x126a789a, 0x1932, 0x3234, 0x21, 0xab, 0x42, 0x64, 0x8a, 0x7b, 0x63, 0x76 | EFI\_TCG2\_PROTOCOL. GetEventLog() - GetEventLog() should record Event from Test 0x0363d22f as last EventLogEntry. | Verify that an event log entry is present at EventLogLastEntry by verifying the last entry. The last entry should be the one added with the HashLogExtendEvent in test 0x0363d22f:  a. Verify TCG\_PCR\_EVENT2.PCRIndex = 16  b. Verify TCG\_PCR\_EVENT2.EventType = EV\_POST\_CODE  c. Verify TCG\_PCR\_EVENT2.Digests.Count = [must equal number of active PCR banks] |

### 30.1.5 SubmitCommmand()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
|  | 0x3aac8b9a, 0x312a, 0x4dcf, 0x12, 0x76, 0x54, 0x55, 0x32, 0xcd, 0x3a, 0xea | EFI\_TCG2\_PROTOCOL. SubmitComand() - SubmitCommand() shall populate the response buffer and return with a status of EFI\_SUCCESS when valid command parameters are passed in. | Invoke SubmitCommand() with a command buffer containing Command TPM2\_HASH Command, and Data to Hash The quick brown fox jumps over the lazy dog".   1. Verify Status returned == EFI\_SUCCESS. 2. Verify returned outHash matches expected result |

### 30.1.6 SetActivePcrBanks()

|  |  |  |  |
| --- | --- | --- | --- |
| Number | GUID | Assertion | Test Description |
|  |  | EFI\_TCG2\_PROTOCOL. SetActivePcrBanks () - SetActivePcrBanks() should return EFI\_INVALID\_PARAMETER with NULL pointer Passed in. | Invoke SetActivePcrBanks() with ActivePcrBanks=0. a. Verify Status returned == EFI\_INVALID\_PARAMETER. |
|  |  | EFI\_TCG2\_PROTOCOL. SetActivePcrBanks () - SetActivePcrBanks() should return EFI\_INVALID\_PARAMETER with banks not defined in HashAlgorithmBitMap. | 1. Invoke GetCapabilty() with valid data size and Structure.  2. Invoke SetActivePcrBanks() with any bit not set in HashAlgorithmBitMap retrieved via GetCapabilities().  a. Verify Status returned == EFI\_INVALID\_PARAMETER. |
|  |  | EFI\_TCG2\_PROTOCOL. SetActivePcrBanks () - SetActivePcrBanks() should return EFI\_SUCESS for all bank permutations defined in HashAlgorithmmBitmap. | Invoke SetActivePcrBanks() with ActivePcrBanks from all non-zero permutations of banks  reported in HashAlgorithmBitMap retrieved via GetCapabilities(). For each Permutation:   1. Verify Status returned == EFI\_SUCCESS. |
|  |  | EFI\_TCG2\_PROTOCOL. SetActivePcrBanks () - SetActivePcrBanks() should return correct banks set by last SetActivePcrBanks () call. | 1. Reboot system  2. Invoke GetActivePcrBanks() with valid ActivePcrBanks buffer. Should return EFI\_SUCCESS, and ActivePcrBanks should be equal to the banks last set in test 30.1.6.3. |