**# Title**: Confidential Computing Extension for ACPI

**# Status**: Submitted to industry standard forum

**# Document**: ACPI Specification

**# License**: SPDX-License-Identifier: CC-BY-4.0

**# Submitter**: [TianoCore Community](<https://www.tianocore.org>)

**# Summary of the change**

**[Background]**

We are adding confidential computing (CC) virtual firmware interface in EDKII. Those interfaces will be consumed by OS. Current implementation is for Intel Trust Domain Extension (TDX). It may be used for other CC solution such as ARM Realm.

However, we cannot use TCG defined interface for TPM, because the CC environment might not produce TPM-like hardware capability. Instead the CC environment may define its own measurement register, such as TDX Runtime Measurement Register.

**[Proposal]**

This proposal adds

1. A new ACPI table - **CCEL**. This ACPI table abstracts the measurement event log report. It is similar to TPM2 ACPI table defined in [TCG ACPI specification].
2. A new ACPI table - **SVKL**. This ACPI table abstracts the storage volume key location.

NOTE: UEFI related extension is added to “CodeFirst - Confidential Computing Extension for UEFI”

Reference:

1. Intel TDX GHCI Specification - <https://www.intel.com/content/www/us/en/developer/articles/technical/intel-trust-domain-extensions.html>
2. UEFI Specification 2.9 - [www.uefi.org](http://www.uefi.org)
3. ACPI Specification 6.4 - [www.uefi.org](http://www.uefi.org)
4. TCG EFI Protocol specification - <https://trustedcomputinggroup.org/resource/tcg-efi-protocol-specification/>
5. TCG ACPI specification - <https://trustedcomputinggroup.org/resource/tcg-acpi-specification/>
6. TCG PFPI specification - <https://trustedcomputinggroup.org/resource/pc-client-specific-platform-firmware-profile-specification/>
7. EDKII - <https://github.com/tianocore/edk2>

**# Benefits of the change**

The OS kernel may use the ACPI table to get measurement event log.

**# Impact of the change**

This is an optional protocol. It will be only available in virtual firmware confidential computing environment.

# Detailed description of the change [normative updates]

### 5.2.31 CC Event Log ACPI Table

This section describes the format of confidential computing (CC) event log ACPI table. A virtual firmware with CC capability may set up an ACPI table to pass the CC event log information. The event log created by the virtual firmware owner contains the hashes to reconstruct the confidential computing (CC) measurement registers.

Table 5.148: CC Event Log ACPI Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Header |  |  |  |
| Signature | 4 | 0 | ‘CCEL’ Signature. |
| Length | 4 | 4 | Length, in bytes, of the entire Table |
| Revision | 1 | 8 | 1 |
| Checksum | 1 | 9 | Entire table must sum to zero. |
| OEMID | 6 | 10 | Standard ACPI header |
| OEM Table ID | 8 | 16 | Standard ACPI header |
| OEM Revision | 4 | 24 | Standard ACPI header |
| Creator ID | 4 | 28 | Standard ACPI header |
| Creator Revision | 4 | 32 | Standard ACPI header |
| CC Type | 1 | 36 | Confidential computing (CC) type.  0: Reserved  1: AMD SEV  2: Intel TDX  3~0xFF: Reserved |
| CC Subtype | 1 | 37 | Confidential computing (CC) type specific sub type. |
| Reserved | 2 | 38 | Reserved. Must be 0. |
| Log Area Minimum Length (LAML) | 8 | 40 | Identifies the minimum length (in bytes) of the system’s pre-boot CC event log area |
| Log Area Start Address (LASA) | 8 | 48 | Contains the 64-bit-physical address of the start of the system's pre-boot CC event log area in QWORD format. Note: The log area ranges from address LASA to LASA+(LAML-1). |

### 5.2.32 Storage Volume Key Location Table

This section describes the format of confidential computing (CC) storage volume key location ACPI table. In CC environment, the storage volume will typically be an encrypted volume. In that case, the virtual firmware may need to support quote generation and attestation to be able to fetch a set of storage-volume key(s) from a remote-key server during boot and pass the key to the guest kernel. Typically, the key is stored in the memory, and the information of the key is passed from virtual firmware via an ACPI table.

Table 5.149: Storage Volume Key Location ACPI Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Header |  |  |  |
| Signature | 4 | 0 | ‘SVKL’ Signature. |
| Length | 4 | 4 | Length, in bytes, of the entire Table |
| Revision | 1 | 8 | 1 |
| Checksum | 1 | 9 | Entire table must sum to zero. |
| OEMID | 6 | 10 | Standard ACPI header |
| OEM Table ID | 8 | 16 | Standard ACPI header |
| OEM Revision | 4 | 24 | Standard ACPI header |
| Creator ID | 4 | 28 | Standard ACPI header |
| Creator Revision | 4 | 32 | Standard ACPI header |
| Key Count (C) | 4 | 36 | The count of key structure |
| Key Structure | 16 \* C | 40 | The key structure |

Table 5.150: Storage-Volume-Key Structure

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Key Type | 2 | 0 | The type of the key.  0: the main storage volume key  1~0xFFFF: reserved. |
| Key Format | 2 | 2 | The format of the key.  0: raw binary.  1~0xFFFF: reserved. |
| Key Size | 4 | 4 | The size of the key in bytes. |
| Key Address | 8 | 8 | The guest-physical address (GPA) of the key. The address must be in ACPI-Reserved Memory. |

**# Special Instructions**

NO