**# Title**: UEFI Cryptographic Algorithm Guideline

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

**# Document**: UEFI Specification Version 2.9

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

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

**[Background]**

Summary of current status for different crypto algorithms.

Table 1 – Crypto Algorithm Support in Standard

|  |  |
| --- | --- |
| **Standard** | **Algorithm** |
| [TCG – EFI protocol](https://trustedcomputinggroup.org/resource/tcg-efi-protocol-specification/) / [Platform Firmware Profile (PFP)](https://trustedcomputinggroup.org/resource/pc-client-specific-platform-firmware-profile-specification/) | SHA1/SHA-{256,384,512}/SM3 |
| [IETF – Network TLS (RFC8446)](https://datatracker.ietf.org/doc/rfc8446/) | TLS Cipher Suite – for negotiation |
| [DMTF – SPDM (Secure Protocol and Data Model) (DSP0274)](https://www.dmtf.org/sites/default/files/standards/documents/DSP0274_1.1.1.pdf) | RSA{SSA,PSS}\_{2048,3072,4096}  ECDSA\_P{256,384,521} + {SHA,SHA3}\_{256,384,512} |
| [PCI-SIG – CMA (Component Measurement and Authentication)](https://pcisig.com/specifications) | RSASSA\_3072/ECDSA\_P{256,384} + SHA{256,384} |
| [UEFI](https://uefi.org/) Secure Boot / Auth Variable | RSA 2048 (PKCS1\_1.5) + SHA 256  (\*Need allow more) |

Table 2 – Crypto Algorithm Requirement / Recommendation

|  |  |
| --- | --- |
| **Guideline** | **Minimal Requirement** |
| [Commercial National Security Algorithm Suite (CNSA Suite Guide)](https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm) | Minimal: SHA384, RSAPSS\_3072/ECDSA\_P384 |
| [PRC CESA Server BIOS Requirement](https://www.cesa.cn/news.aspx?id=bgHZbw0110I=&t=%E9%80%9A%E7%9F%A5%E5%85%AC%E5%91%8A) | Minimal: SHA256/SM3, RSA\_2048/SM2, ECDSA\_256/SM2 |
| [ARM Trusted Board Boot Requirements Client (TBBR-CLIENT) for A profile (DEN0006)](https://developer.arm.com/documentation/den0006/latest) | Minimal: AES\_128, SHA256, ECDSA\_256 (RSAPSS\_2048) |
| [ARM Platform Security Architecture (PSA) Trusted Base System Architecture (TBSA) for M profile (DEN0083)](https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad) | Minimal: AES\_128, SHA256, ECDSA\_256, RSA\_3072 |
| [OCP Secure Boot](https://docs.google.com/document/d/1Se1Dd-raIZhl_xV3MnECeuu_I0nF-keg4kqXyK4k4Wc/edit) | Minimal: SHA384, RSAPSS\_3072/ECDSA\_P384 |
| UEFI Secure Boot / Auth Variable | \*Need provide guidance for compatibility, and for future |

Table 3 – Crypto Algorithm Usage

|  |  |
| --- | --- |
| **Solution** | **Algorithm** |
| [coreboot – Google Binary Blob (GBB) firmware keys](https://link.springer.com/chapter/10.1007/978-1-4842-0070-4_5) | RSA 8192 + SHA512 |
| [IBM Power Secure Boot](https://developer.ibm.com/technologies/linux/articles/protect-system-firmware-openpower/) | ECDSA (512 bit) + SHA512 |
| [MSFT Secure Boot Key](https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-secure-boot-key-creation-and-management-guidance) | RSA 2048 + SHA256 |
| [EDKII](https://github.com/tianocore/edk2) (following UEFI) | RSA 2048 (PKCS1\_1.5) + SHA 256  (\*Need allow more) |

**Reference:**

1. UEFI Specification 2.9 - <https://uefi.org/specifications>
2. PI Specification 1.7 - <https://uefi.org/specifications>
3. Microsoft secure boot key requirement - [https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-secure-boot-key-creation-and-management-guidance#12-public-key-cryptography](https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-secure-boot-key-creation-and-management-guidance)
4. Commercial National Security Algorithm Suite (CNSA Suite Guide) - <https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm>
5. NIST FIPS 140-3 “**Security Requirements for Cryptographic Modules**” - <https://csrc.nist.gov/publications/detail/fips/140/3/final>

**# Detailed description of the change [normative updates]**

ADD means ADD, DELETE means DELETE

**2.6.5 Cryptographic Algorithm Requirement**

1. UEFI variable authentication

For EFI\_VARIABLE\_AUTHENTICATION\_3 or EFI\_VARIABLE\_AUTHENTICATION\_2 descriptor,

SignedData.digestAlgorithms shall support SHA-256 (oid: 2.16.840.1.101.3.4.2.1),

SignerInfo.digestEncryptionAlgorithm be support digest encryption algorithm of RSA with PKCS #1 v1.5 padding (RSASSA\_PKCS1v1\_5) (oid: sha256WithRSAEncryption: 1.2.840.113549.1.1.11).

2. EAP protocol

The cryptographic strength of EFI\_EAP\_TYPE\_TLS shall be at least of hash strength SHA-256 and RSA key length of at least 2048 bits.

3. TLS protocol

The recommended TLS version is 1.2 or 1.3.

4. Secure Boot

The platform key (PK) format shall be at least RSA-2048.

The hash of the UEFI image binary in the dbx shall be at least SHA-256.

5. Hash Protocol and Hash2 Protocol

SHA-1 and MD5 shall only be used for backwards compatibility. For example, SHA-1 shall only be used to support TPM1.2. MD5 shall only be used for iSCSI CHAP.

6. PKCS7 Verify Protocol.

Digest (Hash) Algorithm shall support SHA-256 (oid: 2.16.840.1.101.3.4.2.1).

Digest Encryption shall support sha256WithRSAEncryption (oid: 1.2.840.113549.1.1.11).