ETSI TS 104 875 V1.1.1 (2026-05)

TECHNICAL SPECIFICATION

Cyber Security (CYBER);
Hardware-Based Root of Trust Specification

Reference
DTS/CYBER-00177

Keywords
cyber security

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights 5
Foreword 5
Modal verbs terminology 5
Introduction 5
1 Scope 7
2 References 7
2.1 Normative references 7
2.2 Informative references 7
3 Definition of terms, symbols and abbreviations 8
3.1 Terms 8
3.2 Symbols 8
3.3 Abbreviations 8
4 Applicability of the HBRT specification 8
4.0 Description 8
4.1 Use Case 1: Secure Boot 9
4.1.1 Description 9
4.1.2 Reference Process 9
4.2 Use Case 2: Secure Update 9
4.2.1 Description 9
4.2.2 Reference Process 10
4.3 Use Case 3: Trusted (Measured) Boot 10
4.3.1 Description 10
4.3.2 Reference Process 11
4.4 Use Case 4: Secure Identification 11
4.4.1 Description 11
4.4.2 Reference Process 12
4.5 Use Case 5: Secure Storage 12
4.5.1 Description 12
4.5.2 Reference Process 13
4.6 Use Case 6: Firmware Resilience 13
4.6.1 Description 13
4.6.2 Reference Process 14
4.7 Use Case 7: Trusted I/O and Device Attestation 14
4.7.1 Description 14
4.7.2 Reference Process 15
5 Security Functions/Controls in Hardware-Based Root of Trust 15
5.1 RTM - Root of Trust for Measurement 15
5.2 RTV - Root of Trust for Verification 16
5.3 RTI - Root of Trust for Integrity 16
5.4 RTR - Root of Trust for Reporting 16
5.5 RTS - Root of Trust for Storage 17
5.6 RTU - Root of Trust for Update 17
5.7 RTD - Root of Trust for Detection 17
5.8 RTRec - Root of Trust for Recovery 17
5.9 RTC - Root of Trust for Confidentiality 18
5.10 RTId - Root of Trust for Identification 18
5.11 RTAuthen - Root of Trust for Authentication 18
5.12 RTAuthor - Root of Trust for Authorization 19
5.13 RTComm - Root of Trust for Communication 19
Annex A (informative): Relation to existing and related specifications 20
Annex B (informative): Application Scenarios of RoTs 21

B.1 Trusted Orchestration 21
Annex C (informative): Change history 23
History 24

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Cyber Security (CYBER).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

The concept of a Hardware-Based Root of Trust (HBRT) is fundamental to establish secure and trustworthy computing environments. However, the definition and implementation of Root of Trust (RoT) technologies remain fragmented across various industries and specifications, such as those from ETSI, ISO/IEC, TCG, and GlobalPlatform. This fragmentation is evident in the diverse approaches to RoT functions - such as measurement, reporting, and storage - and in the varying capabilities of HBRT solutions, ranging from discrete to embedded Trusted Platform Module (TPM) implementations. Existing specifications, such as TCG's TPM, often focus on a limited subset of RoT functions, typically covering only measurement, reporting, and storage, leaving gaps in addressing the full spectrum of security needs.

The absence of a comprehensive and unified HBRT standard hinders the industry's ability to meet evolving security demands, including the integration of all (commonly used) RoT functions and the adoption of future-proof technologies like quantum-safe cryptography and crypto agility. Furthermore, customers require a flexible specification that enables clear communication of enhanced security features and supports innovative security technologies through advanced HBRT capabilities.

While HBRT is referenced in existing ETSI specifications, such as those from ETSI ISG NFV, ETSI TC SAI, and even ETSI TC CYBER, there is no cohesive specification to guide its implementation. The present document addresses these challenges by defining a comprehensive HBRT framework that consolidates scattered definitions and encompasses all essential RoT functions as security controls. Note that the present document assumes an HBRT to provide one RoT function. There are indeed variants as described by Global Platform [i.1] where one HBRT might provide several RoT functions or multiple HBRTs provide one RoT function. More information on the related specifications to this document are shown in Table A.1.

1 Scope

The present document defines a comprehensive Hardware-Based Root of Trust (HBRT) standard that consolidates scattered definitions and technologies across industries and specifications. It aims to define all essential Root of Trust functions as security controls, including but not limited to functions such as measurement, reporting, storage, update.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] GlobalPlatform: "Root of Trust - Definitions and Requirements v1.1.1", Public Release, June 2022.

[i.2] Trusted Computing Group: "Roots of Trust Specification", v0.20, Public Review, July 2018.

[i.3] ISO/IEC 11889-1:2015: "Information Technology - Trusted Platform Module Library - Part 1: Architecture".

[i.4] NIST SP 800-164 (2012): "Guidelines on Hardware-Rooted Security in Mobile Devices".

[i.5] NIST SP 800-193 (2018): "Platform Firmware Resiliency Guidelines".

[i.6] ETSI GR NFV-SEC 009 (V1.3.1): "Network Functions Virtualisation (NFV); NFV Security; Report on use cases and technical approaches for multi-layer host administration".

[i.7] ETSI GS NFV-SEC 023: "Network Functions Virtualisation (NFV) Release 5; Security; Container Security Specification".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
AVS	Attestation Verifier Service
CA	Certificate Authority
CoT	Chain of Trust
eDRTM	external Dynamic Root of Trust
HBRT	Hardware Based Root of Trust
HMEE	Hardware Mediated Execution Environment
KGV	Known Good Values
NV	Non Volatile
PEnE	Policy Enforcement Engine
RoT	Root of Trust
RoTU	Root of Trust for Update
RTAuthen	Root of Trust for Authentication
RTAuthor	Root of Trust for Authorization
RTC	Root of Trust for Confidentiality
RTComm	Root of Trust for Communication
RTD	Root of Trust for Detection
RTE	Run Time Environment
RTI	Root of Trust for Integrity
RTId	Root of Trust for Identification
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTRec	Root of Trust for Recovery
RTS	Root of Trust for Storage
RTU	Root of Trust for Update
RTV	Root of Trust for Verification
SE	Secure Element
SecID	Security Identifier
SRK	Storage Root Key
TCB	Trusted Computing Base
TCG	Trusted Computing Group
TEE	Trusted Execution Environment
TMF	TEE Management Framework
TPM	Trusted Platform Module
VM	Virtual Machine

4 Applicability of the HBRT specification

4.0 Description

This clause aims to provide a single use case for every RoT function described in clause 5.

4.1 Use Case 1: Secure Boot

4.1.1 Description

During the Boot phase of a device, Secure Boot cryptographically verifies all Boot Components' integrity and authenticity before executing them.

4.1.2 Reference Process

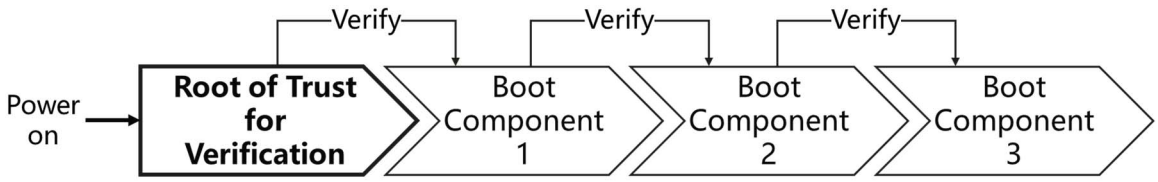

Figure 1

Typical steps for a Secure Boot process are shown in Figure 1 and described as follows:

1) After power-on, the Root of Trust for Verification (RTV) verifies the integrity and authenticity of the first mutable Boot Component (Boot Component 1) against a known-good cryptographic reference stored in protected memory under control of the hardware-based Root of Trust and then executes it.

2) Afterwards, Boot Component 1 verifies Boot Component 2 using the corresponding known-good cryptographic reference and then executes it; Boot Component 2 verifies Boot Component 3, and so on.

3) After all Boot Components are sequentially verified and executed, the overall platform reaches an authorized boot state and overall integrity can be attested by the RTR using signed measurements against known-good references.

4) If any verification fails, the device will fail securely and either remain in a secure state, enter recovery to a known-good image, or restart the verification sequence.

4.2 Use Case 2: Secure Update

4.2.1 Description

During the update phase of a device, Secure Update cryptographically verifies the integrity and authenticity of all Update Packages before installing them.

4.2.2 Reference Process

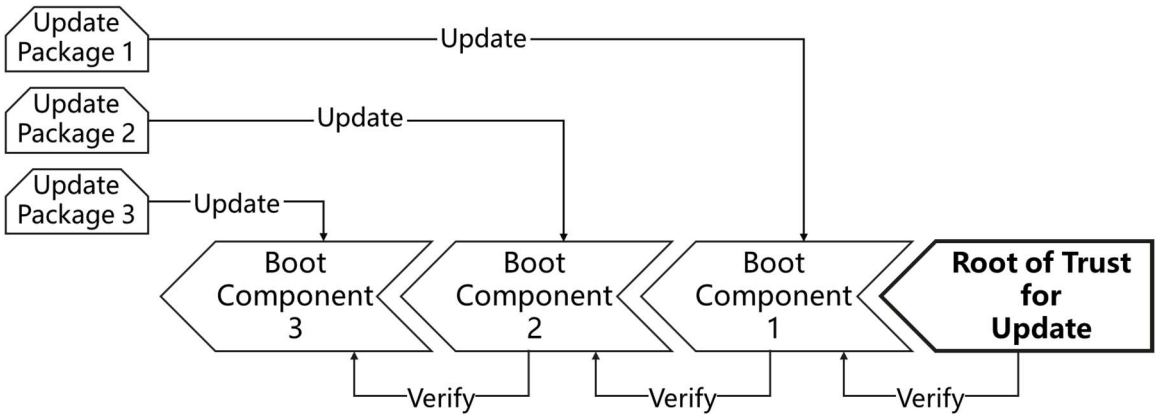

Figure 2

Typical steps for a Secure Update process are shown in Figure 2 and described as follows:

1) When the owners or administrators initiate an update, they build and sign different Update Packages for the respective Boot Components.

2) Each Update Package is transmitted to the device, where a higher trust level Boot Component verifies the Update Package's integrity and authenticity and then installs it.

EXAMPLE: When the device uses Update Package 3 to securely update Boot Component 3, the verification only needs to be performed by Boot Component 2.

3) The verification could be performed at runtime, upon reboot, or as a combination of both.

4) If the verification fails, the device will refuse the update package, and maintain the status before update.

4.3 Use Case 3: Trusted (Measured) Boot

4.3.1 Description

During the Boot phase of a device, Trusted Boot cryptographically measures all Boot Components before executing them.

Each measurement is securely stored in protected storage and later cryptographically signed and reported to a verifier.

The verifier authenticates the signed measurements and compares them against Known Good Values (KGVs) to determine whether the device booted in a trusted and unmodified state.

4.3.2 Reference Process

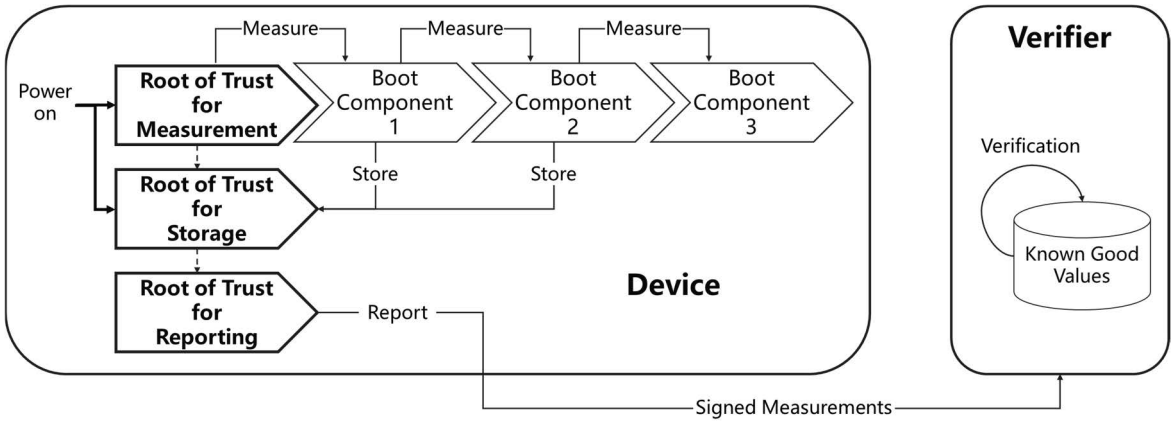

Figure 3

Typical steps for a Trusted Boot process are shown in Figure 3 and described as follows:

1) After power-on, the Root of Trust for Measurement (RTM) measures the integrity of the first mutable Boot Component, Boot Component 1. The resulting measurement (e.g. a cryptographic hash) is to be stored by the Root of Trust for Storage (RTS). Afterwards, Boot Component 1 is to be executed.

2) Each subsequent boot component measures the next one in the chain. Boot Component 1 measures Boot Component 2 and stores the result in the RTS, then executes it. Boot Component 2 measures Boot Components 3 and stores the result, and so on, until all Boot Components have been measured and executed in sequence.

3) After all measurements have been recorded, the Root of Trust for Reporting (RTR) binds the complete set of measurements to a device-unique private key inside a protected execution environment. The RTR then creates a cryptographically signed measurement report and transmits it to the verifier.

4) The verifier authenticates the Signed Measurement report, checks its integrity and freshness (e.g. using a nonce or timestamp), and compares each measurement against the set of Known Good Values (KGVs).

5) If the comparisons success, the verifier concludes that the device has powered on securely and no boot component has been modified.
If any comparison fails, the verifier identifies which component deviates from its expected state, indicating possible tampering or corruption.

4.4 Use Case 4: Secure Identification

4.4.1 Description

The device establishes its identity using a hardware-protected Device Identifier generated by RTId. A Certificate Authority signs the public part of this identifier. Other devices authenticate the SecID Generator Device by validating the Identity Certificate and verifying signatures created with the Device Identifier.

4.4.2 Reference Process

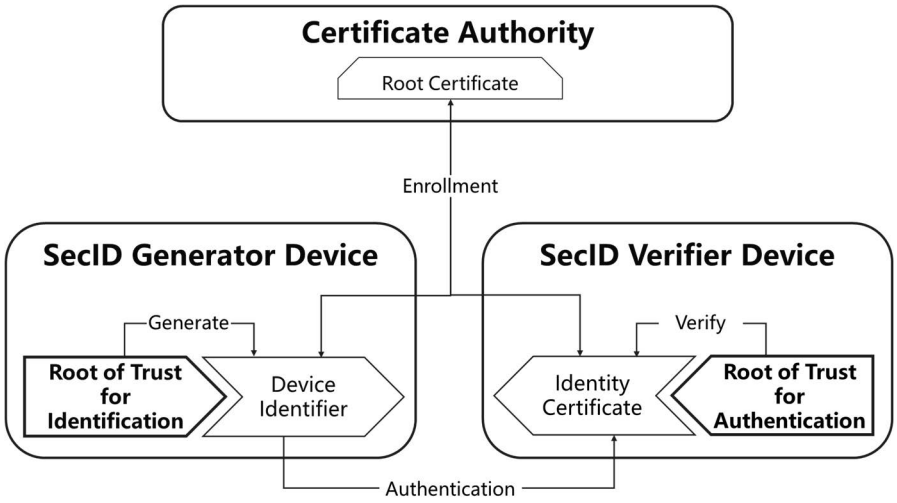

Figure 4

Typical steps for a Secure Identification process are shown in Figure 4 and described as follows:

1) During Identity Generation, RTId of the SecID Generator Device generates a cryptographic Device Identifier, for example a key pair. The private key remains inside RTId. The public key is prepared for signing by the Certificate Authority.

2) During Certificate Authorization, the Certificate Authority signs the public key of the Device Identifier with its Root CA private key. This produces the Identity Certificate of the SecID Generator Device.

3) During Identity Verification, RTAuthen of the SecID Verifier Device verifies the authenticity and integrity of the Root Certificate and then verifies the Identity Certificate using the Root Certificate's public key.

4) During the Authentication Process, the SecID Generator Device signs a challenge value, such as a nonce, with the private key of the Device Identifier. The SecID Verifier Device verifies the signature using the public key from the Identity Certificate.

5) The authentication process includes a freshness mechanism, for example a nonce, to prevent replay.

4.5 Use Case 5: Secure Storage

4.5.1 Description

Secure Storage protects critical data at rest by encrypting the data and generating integrity information. All storage keys are generated and protected by RTC and RTI.

4.5.2 Reference Process

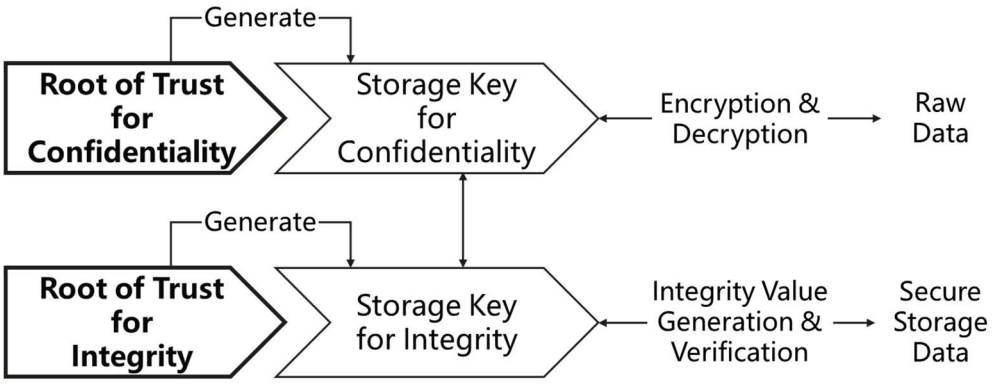

Figure 5

Typical steps for a Secure Storage process are shown in Figure 5 and described as follows:

1) Data Storage:

   a) The Device requests RTC to generate or derive a Storage Key for Confidentiality and uses this key to encrypt the Raw Data, producing Encrypted Data.

   b) The Device requests RTI to generate or derive a Storage Key for Integrity and uses this key to compute an Integrity Value over the Raw Data.

   c) The Device stores the Encrypted Data and the Integrity Value together as Secure Storage Data in memory.

2) Data Load:

   a) The Device retrieves the Secure Storage Data from memory.

   b) The Device uses RTC to decrypt the Encrypted Data to obtain Raw Data.

   c) The Device uses RTI to recompute the Integrity Value over the Raw Data and compares it with the stored Integrity Value. A match indicates unmodified data; a mismatch indicates modification.

3) In practice, encryption and integrity generation may be combined using authenticated encryption schemes.

4.6 Use Case 6: Firmware Resilience

4.6.1 Description

Firmware Resilience protects Boot Components and their Back-ups from unauthorized modification, detects tampering, and restores firmware to a Known Good Back-up if corruption is detected.

4.6.2 Reference Process

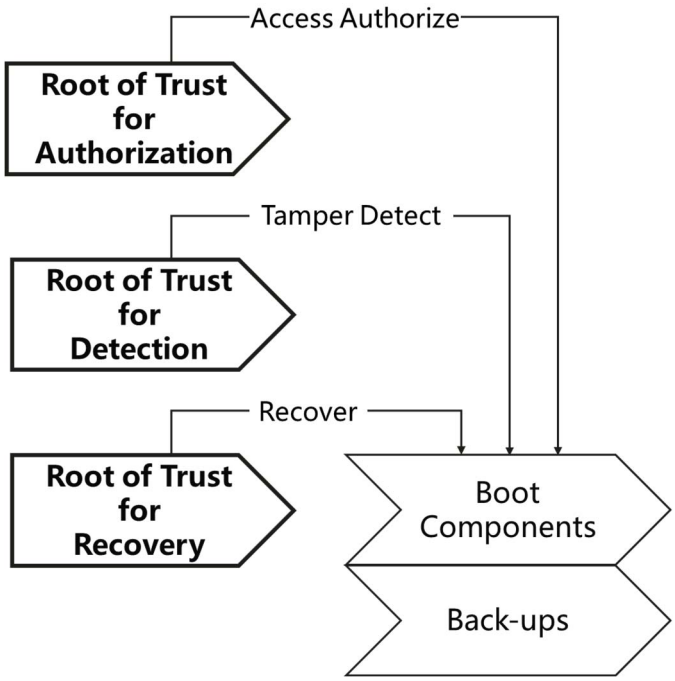

Figure 6

Typical steps for a Firmware Resilience process are shown in Figure 6 and described as follows:

1) In the boot-time, the Device requests RTAuthor to configure access control for Boot Components and their Back-ups. During runtime, access is limited according to this configuration.

2) In the boot-time or runtime, the Device requests RTD to detect directly or indirectly whether Boot Components or their Back-ups have been tampered or corrupted. Detection may use Known Good Values.

NOTE: "Directly" means that detection is performed by RTD, "indirectly" means that detection is performed by the chain of trust rooted in RTD.

3) If Boot Components or their Back-ups are identified as tampered or corrupted, the Device requests RTRec to recover them using Known Good Back-ups. Recovery Back-ups may be verified by RTV or RTU before they are used.

In practice, A/B Back-ups or similar schemes may be used to prevent a tampered or interrupted update from bricking the system and to allow fallback to a previous good state.

4.7 Use Case 7: Trusted I/O and Device Attestation

4.7.1 Description

Trusted I/O protects communication between HMEEs of the Host and the Device by mutual authentication, key exchange and sensitive data encryption based on the Root of Trust for Communication, and with this secure communication the Host could collect the Device Measurements Report from the Device and push it to the Verifier for Device Attestation.

4.7.2 Reference Process

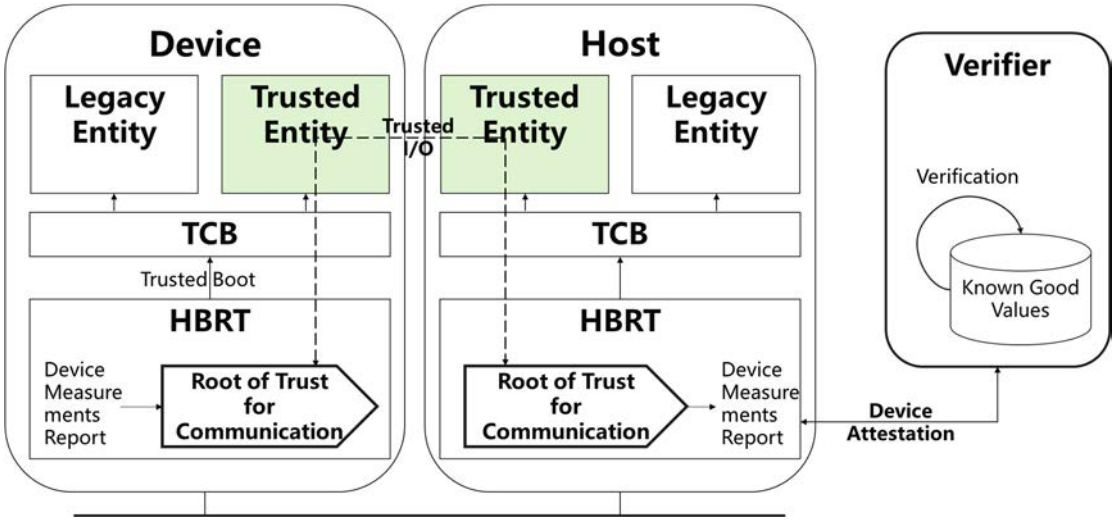

Figure 7

Typical steps for a Trusted I/O and Device Attestation process are shown in Figure 7 and described as follows:

1) In the boot-time, the Device performs the Trusted Boot based on its own HBRT. The Device Measurements Report is generated by the HBRT of the Device.

2) In the runtime, HMEEs of the Host and the Device securely communicate with each other by mutual authentication, key exchange and optional data encryption based on their Root of Trust for Communication.

3) Afterwards, the HMEE of the Host receives the Device Measurements Report from the HMEE of the Device over the secure communication and then attests the Device Measurements Report to the Verifier.

4) The Verifier could verify the integrity and authenticity of the Device Measurements Report, and then tells the Host if the Device is authenticated and its integrity verified or not.

5) After the verification, the Host could know whether the I/O could be trusted or not.

5 Security Functions/Controls in Hardware-Based Root of Trust

5.1 RTM - Root of Trust for Measurement

Description:

The Root of Trust for Measurement (RTM) measures the integrity of code, configuration, and other critical data before execution and records digests in a chained structure ("measure-before-execute"). It corresponds to: GlobalPlatform v1.1.1 [i.1] (section 4.5), ISO/IEC 11889-1:2015 [i.3] (clauses 10 and 22), NIST SP 800-164 [i.4] (section 3.1.1).

Requirements:

[RTM-1]: The RTM SHALL measure each mutable component prior to execution.

[RTM-2]: The RTM SHALL store each measurement in protected storage controlled by the HBRT (RTS).

[RTM-3]: The RTM SHALL chain measurements (each component measures the next) to maintain sequence integrity.

[RTM-4]: The RTM SHALL ensure measurement data are integrity-protected and non-modifiable after storage.

[RTM-5]: The RTM SHOULD protect against replay, tampering, or fault injection during measurement.

5.2 RTV - Root of Trust for Verification

Description:

The Root of Trust for Verification validates the integrity and authenticity of code and data objects before execution. It accepts an object together with its cryptographic evidence and verifies signatures or hashes against trusted references. This function anchors secure boot, since the first mutable component may only execute after successful verification.

The Root of Trust for Verification (RTV) validates the integrity and authenticity of signed code or data before execution against trusted references maintained under HBRT control. RTV provides the verification anchor for Secure Boot and supports key revocation and rollback prevention. It corresponds to GlobalPlatform v1.1.1 [i.1] (sections 4.1 to 4.4, 4.8 to 4.9, 7.2.1, 7.2.2), TCG Roots of Trust v0.20 [i.2] (sections 4.1 to 4.2.2), NIST SP 800-164 [i.4] (sections 3.1.1 to 3.1.3).

Requirements:

[RTV-1]: The RTV SHALL verify the first mutable component at power-on against a known-good reference before execution.

[RTV-2]: The RTV SHALL verify all subsequent components before execution (verify-before-execute chain).

[RTV-3]: Trusted reference material SHALL be stored in protected memory under HBRT control.

[RTV-4]: The RTV SHALL fail securely and prevent further booting on verification failure.

[RTV-5]: The RTV SHOULD support key revocation and update of reference material under policy.

5.3 RTI - Root of Trust for Integrity

Description:

The Root of Trust for Integrity (RTI) maintains the authenticity of critical data and prevents unauthorized modification or downgrade. It corresponds to: GlobalPlatform v1.1.1 [i.1] (section 4.4), ISO/IEC 11889-1:2015 [i.3] (clause 10), NIST SP 800-164 [i.4] (section 3).

Requirement:

[RTI-1]: Integrity metadata SHALL be cryptographically bound (e.g. signed or hashed) to prevent substitution.

[RTI-2]: The RTI SHALL offer controlled update interfaces.

[RTI-3]: The RTI SHOULD support verification of firmware or configuration integrity during runtime.

5.4 RTR - Root of Trust for Reporting

Description:

The Root of Trust for Reporting (RTR) generates signed integrity reports binding system measurements or claims to the device identity and freshness data (e.g. nonce). It corresponds to: GlobalPlatform v1.1.1 [i.1] (section 4.7), ISO/IEC 11889-1:2015 [i.3] (clause 10), NIST SP 800-164 [i.4] (sections 3.1.1 to 3.1.3).

Requirements:

[RTR-1]: The RTR SHALL produce signed reports containing integrity measurements, device identity, and freshness data.

[RTR-2]: The RTR SHALL use keys protected by HBRT and marked non-exportable.

[RTR-3]: External verifiers SHALL authenticate report origin and compare values to known-good references.

[RTR-4]: The RTR SHOULD support timestamping or monotonic counters for replay prevention.

5.5 RTS - Root of Trust for Storage

Description:

The Root of Trust for Storage (RTS) protects secrets and sensitive data at rest and in use by other RoT functions via a key hierarchy anchored by the SRK. It corresponds to: GlobalPlatform v1.1.1 [i.1] (section 4.2), ISO/IEC 11889-1:2015 [i.3] (clauses 10, 11.7, 22), NIST SP 800-164 [i.4] (section 3).

Requirements:

[RTS-1]: The RTS SHALL provide shielded storage for private keys and critical security parameters.

[RTS-2]: The RTS SHALL implement a hardware-anchored key hierarchy rooted in the Storage Root Key (SRK).

[RTS-3]: The RTS SHALL support sealed storage tied to device state to prevent unauthorized use.

[RTS-4]: The RTS SHOULD protect against rollback or tampering of stored secrets.

5.6 RTU - Root of Trust for Update

Description:

The Root of Trust for Update authenticates, authorizes, and applies firmware or configuration updates, enforces anti-rollback, and performs installation atomically or recoverably. It corresponds to: GlobalPlatform v1.1.1 [i.1] (section 4.8), TCG Roots of Trust v0.20 [i.2] (sections 6.1 and 8.1), NIST SP 800-193 [i.5] (sections 3.1 to 3.3, 3.6.5, 4.1).

Requirement:

[RTU-1]: The RTU SHALL verify integrity and authenticity of all update packages before installation.

[RTU-2]: The RTU SHALL enforce rollback prevention via trusted versioning or deny-lists.

[RTU-3]: The RTU SHALL perform updates atomically or ensure verifiable recovery if interrupted.

[RTU-4]: The RTU SHALL support key rotation and revocation managed by HBRT.

[RTU-5]: The RTU SHALL generate audit logs of update events protected by HBRT.

5.7 RTD - Root of Trust for Detection

Description:

The Root of Trust for Detection (RTD) detects corruption or tampering of firmware/configuration or abnormal runtime conditions and triggers secure responses. It corresponds to NIST SP 800-193 [i.5] (sections 3.1 to 3.3, 3.6.5, 4.1).

Requirements:

[RTD-1]: RTD SHALL detect unauthorized firmware or configuration modification during runtime.

[RTD-2]: RTD SHOULD monitor environmental or fault conditions and enforce protective responses.

[RTD-3]: RTD SHOULD record detection events in tamper-protected logs for recovery.

5.8 RTRec - Root of Trust for Recovery

Description:

The Root of Trust for Recovery (RTRec) restores the platform to a known-good state when corruption or failure occurs and authenticates recovery images against HBRT keys. It corresponds to NIST SP 800-193 [i.5] (sections 3.3, 3.6.5, 4 and 4.4).

Requirements:

[RTRec-1]: Recovery images SHALL be authenticated with HBRT-bound keys before execution.

[RTRec-2]: The RTRec SHALL restore verified components to a known-good baseline.

[RTRec-3]: Recovery processes SHALL log restoration results under HBRT control.

[RTRec-4]: The RTRec SHOULD integrate with RTV verification on next boot.

5.9 RTC - Root of Trust for Confidentiality

Description:

The Root of Trust for Confidentiality (RTC) ensures secret material remains protected and released only in authorized contexts. It corresponds to: GlobalPlatform v1.1.1 [i.1] (section 4.2), ISO/IEC 11889-1:2015 [i.3] (clauses 10 and 22), NIST SP 800-164 [i.4] (section 3).

Requirements:

[RTC-1]: The RTC SHALL store and use secrets exclusively within shielded locations under HBRT control.

[RTC-2]: The RTC SHALL prevent unauthorized export or duplication of private keys.

[RTC-3]: The RTC SHALL restrict secret use to authenticated and authorized contexts.

[RTC-4]: The RTC SHOULD employ hardware encryption or memory isolation for additional confidentiality.

5.10 RTId - Root of Trust for Identification

Description:

The Root of Trust for Identification (RTId) establishes a unique device identity distinct from higher-level platform identities. It provisions identity keys in shielded storage and exposes cryptographic proofs of device identity. It corresponds to GlobalPlatform v1.1.1 [i.1] (sections 4.3 and 7.2).

Requirements:

[RTId-1]: Identity keys SHALL be generated or provisioned in shielded locations controlled by HBRT.

[RTId-2]: Identity credentials SHALL be certified by trusted issuers or owners.

[RTId-3]: RTId SHALL make identity proofs available to RTR and RTAuthen under policy.

[RTId-4]: RTId SHOULD support lifecycle management (renewal, revocation).

5.11 RTAuthen - Root of Trust for Authentication

Description:

The Root of Trust for Authentication (RTAuthen) verifies the identity of entities accessing protected services, protects credentials, and establishes secure channels. It corresponds to: GlobalPlatform v1.1.1 [i.1] (section 4.1), ISO/IEC 11889-1:2015 [i.3] (clauses 11 and 5).

Requirements:

[RTAuthen-1]: Authentication data SHALL be stored and processed in shielded locations.

[RTAuthen-2]: The RTAuthen SHALL support mutual authentication with challenge-response.

[RTAuthen-3]: The RTAuthen SHALL establish session keys for secure channels under HBRT control.

[RTAuthen-4]: The RTAuthen SHOULD support algorithm agility for future cryptographic transitions.

5.12 RTAuthor - Root of Trust for Authorization

Description:

The Root of Trust for Authorization (RTAuthor) enforces access-control policies using authorization tokens in shielded storage and exposes only decision results. It corresponds to: GlobalPlatform v1.1.1 [i.1] (section 4.6), ISO/IEC 11889-1:2015 [i.3] (clause 11.5), NIST SP 800-193 [i.5] (section 3.6.5).

Requirements:

[RTAuthor-1]: Authorization decisions SHALL be computed using policy and token data stored in shielded storage.

[RTAuthor-2]: The RTAuthor SHALL return only decision results and no sensitive state.

[RTAuthor-3]: The RTAuthor SHALL support policy update under trusted control with verifiable provenance.

[RTAuthor-4]: The RTAuthor SHOULD log authorization events securely for audit.

5.13 RTComm - Root of Trust for Communication

Description:

RTComm is the Root of Trust for Communication that enables secure interactions between Host and Device HMEEs through mutual authentication (via public keys, certificates, or Pre-Shared Key), key exchange, and optional sensitive data encryption, while supporting the collection and attestation of Device Measurements Reports.

Requirements:

[RTComm-1]: The RTComm SHALL support mutual authentication by Public Key, Certificates or Pre-Shared Key.

[RTComm-2]: The RTComm SHALL support key exchange by Public Key or Pre-Shared Key.

[RTComm-3]: The RTComm SHALL support sensitive data encryption and decryption optionally.

[RTComm-4]: The RTComm SHALL support Measurements or Measurements Report collection from Device/Host.

[RTComm-5]: The cryptographic schemes and parameters used by RTComm SHALL be algorithm-agile and support coexistence of multiple trusted algorithms and keys.

Annex A (informative): Relation to existing and related specifications

Table A.1

Ref	Source	Sections / Clauses Used	Functional Contribution
[i.1]	GlobalPlatform Root of Trust - Definitions and Requirements v1.1.1 (2022)	4.1 Authentication 4.2 Confidentiality 4.3 Identification 4.4 Integrity 4.5 Measurement 4.6 Authorization 4.7 Reporting 4.8 Update 4.9 Verification 7.2.1 Boot of TEE RTE 7.2-7.2.4 TMF 7.2.2 (p. 62) Failure handling 8.9.2 (SE use case)	Defines RTAuthen primitives used for entity authentication and to bootstrap secure sessions (RTCom) and gate policy decisions (RTAuthor); Provides RTC/RTS foundation for secret handling; key protection; and sealed operations; Implements RTId for device/RoT identity binding used by RTR (attestation) and RTCom (channel binding); Realizes RTI to protect non-secret but critical data (keys; certs; configs) and detect substitution/downgrade; Defines RTM to measure-before-execute and produce chained measurements for attestation; Implements RTAuthor to evaluate policies/tokens and return decisions with minimal disclosure; Defines RTR for signed; fresh reports binding state/measurements to identity for remote attestation; Defines RTU to authenticate/authorize updates and support atomicity/rollback controls; Defines RTV for verify-before-execute; anchoring secure boot and verification chains; Implementation guidance for chaining RTV across boot phases and binding services to verified state; Shows RTR/RTId export patterns for management/remote discovery; Ensures fail-secure boot path underpins RTV/RTU error handling; Illustrates RTV/RTI usage for runtime verification in SE contexts
[i.2]	Trusted Computing Group Roots of Trust Specification v0.20 (2018)	4.1 Introduction 4.2.2 Trust Models 6.1 RoTU 6.2 Lifecycle 8.1 RoTU requirements	Establishes RoT taxonomy used to align RTM/RTR/RTS/RTU/RTV; Guides design choices for bootstrapped RTV and vendor-controlled RTU; Frames RTU as privileged update controller within device lifecycle; Details operational flow between Update Source; Owner; RoTU; Provides normative requirements for RTU including atomicity and rollback protections
[i.3]	ISO/IEC 11889-1:2015 (TPM 2.0 Part 1: Architecture)	10 TPM Protections 11.5 Authorization 11.7 NV Memory 22 Protected Storage	Underpins RTS/RTC/RTI (secure storage/use), supports RTV key protection and RTR attestation trust; Supports RTAuthor policy evaluation and RTCom session policy/downgrade resistance; Provides persistent RTS facilities and rollback-aware NV handling; Delivers RTS/RTC cryptographic protections (sealing; integrity) for stored objects and keys
[i.4]	NIST SP 800-164 (2012) Guidelines on Hardware-Rooted Security in Mobile Devices	3 (overview) 3.1.1 RoTs 3.1.2 API to RoTs 3.1.3 PEnE Executive context	Baseline model for RTM/RTR/RTS and platform API/PEnE integration; Maps core RoTs to functions (measurement, reporting, storage) used throughout; Guides RTCom/RTAuthor/RTR integration into higher layers; Context for platform policy flows leveraging RTAuthor/RTR; Reinforces adoption of RTS/RTM/RTR capabilities on devices
[i.5]	NIST SP 800-193 (2018) Platform Firmware Resiliency Guidelines	Executive summary 3.1 Principles 3.2 Resiliency properties 3.3 RoTs & CoTs 3.6.5 Event logging 4.1 Roots of Trust 4.4 Recovery	Frames objectives for RTU/RTD/RTRec; Sets functional goals and controls for RTU/RTD/RTRec; Classification criteria informing RoT/CoT implementation depth; Anchors RTU/RTD/RTRec and their chains of trust; Provides audit/logging requirements used by RTU/RTD/RTRec/RTCom/RTAuthor; Directly defines the device-level functional scope for RTU/RTD/RTRec; Specifies RTRec behaviour for authenticated recovery to a known-good state

Annex B (informative): Application Scenarios of RoTs

B.1 Trusted Orchestration

The trusted orchestration deals with the boot of a workload in a VM- or container-based virtualised environment. The environment can be implemented on one or more servers. The workload is constituted by components, which each of them can be a VM or a container. Implementing these components in Hardware Mediated Execution Environment (HMEE), as defined in ETSI GR NFV-SEC 009 [i.6], clause 6.16 provides a hardware-based and verifiable protection of these components.

While components under certain circumstances can be fully protected by HMEEs, there also exist setups where for reasons like performance, there also exist components which are not fully protected by HMEEs.

By including a measuring component in the Trusted Boot chain described in Use Case 3, this chain can be used to provide hardware-rooted trust to parts of workload components not protected by the HMEE. In this way, the complete workload component can be cryptologically measured and rooted in hardware, partly by the hardware-based RTM of the host (e.g. using a TPM) and partly by Root of Trust Services in the HMEE provided by the hardware platform for the HMEE.

The measurement having the trust origin from the Trusted Boot as well as the measurement having the trust origin from the HMEE are cryptographically signed and reported to an Attestation Verifier Service (AVS) as defined in ETSI GS NFV-SEC 023 [i.7]. The AVS authenticates the signed measurements and compares them against Known Good Values (KGVs) to determine whether each of the components in the workload are booted in a trusted and unmodified state.

Typical steps for a Trusted Orchestration process (consider that components of type A, which exists completely inside of an HMEE, can coexist with components of type B, which partly exists inside of an HMEE, as depictured but implementations can also have only components of type A or only components of type B) are shown in Figure B.1.

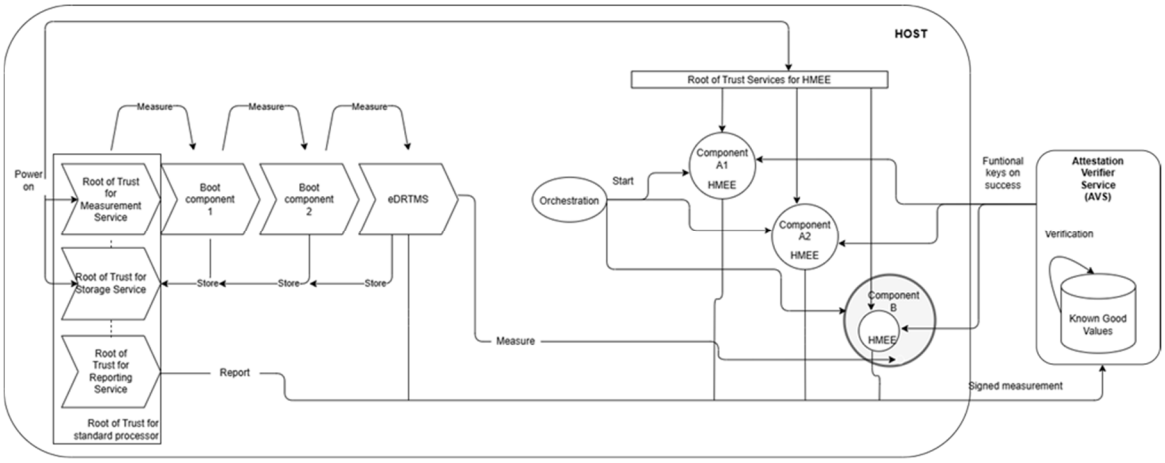

Figure B.1

1) As a result of the Trusted Boot process, the components included in the boot process are measured. A measuring component, here denoted external Dynamic Root of Trust Measurement (eDRTM) (see ETSI GS NFV-SEC 023 [i.7]), can be included in this Trusted Boot process measurement. This eDRTM will thus be anchored in the Root of Trust for Measurement (RTM) and Root of Trust for Reporting (RTR) of the Trusted Boot. The eDRTM will be used to dynamically measure parts of workload components not protected by an HMEE, illustrated for Component B, to create hardware-based and verifiable trust also for these parts of the components. The boot-time measurement of the Trusted Boot, including measurement of all boot-steps up to and including the eDRTM, will constitute an additional part of the measurement, to be able to verify the eDRTM and the execution environment of the workload component.

For workload components, or part of workload components, protected by an HMEE, the RTM and RTR are instead based on the HMEE-based attestation and thus related to the HMEE-protected component. The HMEE also provides attestable confidentiality and integrity isolation from the rest of the environment which is anchored in the RTC and the RTI rooted in the HBRT used for the HMEE. This is illustrated for Component A1 and Component A2 as well as parts of Component B. For these workload components, or part of workload components, the eDRTM is not used. RTM, RTR, RTC and RTI for the HMEE protected components are depicted as "Root of Trust Services for HMEE" in Figure B.1. The Orchestrator processes the workload components in an iterative process but is not involved in the trust process.

2) Each of the component's reports are transmitted to the verifier as a cryptographically signed measurement report. The AVS can be hierarchical and any AVS implementation existing in a cloud environment is to be protected, e.g. by an HMEE.

3) The AVS authenticates the Signed Measurement reports, checks their integrity and freshness (e.g. using a nonce or timestamp), and compares each measurement against the set of Known Good Values (KGVs).

4) If the comparisons success, the AVS concludes that the component has started on securely and has not been modified, thus it can then deliver the functional key the component needs to fulfil its role in the workload.

5) If any comparison fails, i.e. the AVS identifies that a component deviates from its expected state, indicating possible tampering or corruption, this implies that no functional keys are given to that component.

The orchestrator acts on readiness or liveness signal from each component either to fulfil its intent or according to its policy remove the entire workload.

Annex C (informative): Change history

Date	Version	Information about changes
03.09.2025	V0.0.1	Presentation of skeleton
31.10.2025	V0.0.2	Presentation of 3 use cases and first descriptions of some root of trust functions
26.11.2025	V0.0.3	Presentation of 1st draft with nearly full content
28.11.2025	V0.0.4	Revision of 1st draft with nearly full content
11.02.2026	V0.0.5	1st stable draft
30.03.2026	V0.0.6	Stable draft after final comment resolution
08.04.2026	V0.0.7	Final draft for approval

History

Version	Date	Status
V1.1.1	May 2026	Publication