TRUSTED® COMPUTING GROUP

SPECIFICATION

TCG Platform Certificate Profile

Version 2.1
January 27, 2026

Contact: admin@trustedcomputinggroup.org

PUBLISHED

DISCLAIMERS, NOTICES, AND LICENSE TERMS

THIS SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Without limitation, TCG disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and TCG disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.

Contact the Trusted Computing Group at www.trustedcomputinggroup.org for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.

ACKNOWLEDGEMENT

The TCG wishes to thank those who contributed to this specification. This document builds on considerable work done in the various working groups in the TCG.

Special thanks to the members of the Platform Certificate Interoperability subgroup and others contributing to this document:

Name	Affiliation
Eugene Cho	Dell, Inc.
Ludovic Jacquin	NVIDIA Corporation
Theo Koulouris (editor)	Hewlett Packard Enterprise
Amy Nelson	Dell, Inc.
Lawrence Reinert	United States Government
Joshua Schiffman	HP Inc.
Dick Wilkins	Phoenix Technologies

CONTENTS

FIGURES

TABLES

1 Introduction

1.1 Purpose

This specification defines the TCG Platform Certificate Profile Version 2.1. It defines the certificate information model and its encoding in ITU-T X.509 standard format. The encoding is defined for both X.509 attribute certificates and X.509 public key certificates.

1.2 Key Words

The key words “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” in this document normative statements are to be interpreted as described in RFC-2119 [4], Key words for use in RFCs to Indicate Requirement Levels.

1.3 Statement Type

Please note a very important distinction between different sections of text throughout this document. There are two distinctive kinds of text: informative comment and normative statements. Because most of the text in this specification will be of the kind normative statements, the authors have informally defined it as the default and, as such, have specifically called out text of the kind informative comment. They have done this by flagging the beginning and end of each informative comment and highlighting its text in gray. This means that unless text is specifically marked as of the kind informative comment, it can be considered a kind of normative statement.

EXAMPLE: Start of informative comment

This is the first paragraph of 1–n paragraphs containing text of the kind *informative comment* ...

This is the second paragraph of text of the kind *informative comment* ...

This is the nth paragraph of text of the kind *informative comment* ...

To understand the TCG specification the user must read the specification. (This use of MUST does not require any action).

End of informative comment

1.4 Document Structure

Start of informative comment

This document provides a complete definition of the Platform Certificate. The abstract definition of the information contained in the Platform Certificate (i.e., the information model) is in Section 2. Section 3 specifies how a Platform Certificate is concretely encoded as an X.509 certificate. Section 4 presents the definition of the **`TRAIT`** class and **`Trait`** instance types. Relevant ASN.1 structures and OIDs are in Section 5.

End of informative comment

1.5 Relationship to Other TCG Specifications

Start of informative comment

This specification references the TCG Infrastructure Working Group Reference Architecture for Interoperability [2], the TCG TPM Main Specification [3], the TCG Credential Profiles for TPM Family 1.2 [6], the EK Credential Profile Specification [7], the PC Client Platform Firmware Profile Specification (PFP) [11], TCG Glossary [10], and the TCG Algorithm Registry Specification [13]. This specification replaces the Platform Credential Specification defined in the TCG Credential Profiles for TPM Family 1.2 [6].

End of informative comment

1.6 Intended Audiences

Start of informative comment

The intended audience for this document is entities, such as Verifiers and Attestation CAs, which are expected to participate in the TCG infrastructure. Computer OEMs and the companies in the OEM supply chain, such as TPM vendors and components vendors, are also intended audiences for this document.

End of informative comment

1.7 Definition of Terms

Start of informative comment

The TCG Glossary [1] contains definitions that are fundamental to this specification. Rather than repeat those definitions, the reader is assumed to be familiar with the terms in the TCG glossary.

The following operational definitions, however, are specific to this specification.

Certificate – An artifact that cryptographically binds a subject's identity to its public key or attributes using the industry-standard certificate structure from ISO/IEC/ITU-T X.509 version 3. Certificate generation consists of (a) assembling values for the certificate fields and (b) signing over the assembled fields.

Note: The term "Credential", used in previous versions of the TCG Platform Certificate Profile and Platform Attribute Credential Profile specifications, has been replaced with "Certificate" throughout this document. Certificate is a more precise term to describe this artifact, as a credential is a combination of a certificate and a private key, accessible for the required purpose (e.g., creating a signature). Any uses of the word "Credential" in this document refer to titles of previously published specifications, attributes, or extensions.

End of informative comment

1.8 Abbreviations

Abbreviation	Description
AC	Attribute Certificate
OID	Object IDentifier
PKC	Public Key Certificate

2 TCG Platform Certificate Information Model

Start of informative comment

A Platform Certificate is a signed statement describing characteristics of a platform that can affect the platform's trustworthiness. Concretely, a Platform Certificate provides the foundation for binding the identity of the platform to one or more Roots of Trust and Trusted Building Block(s) of the platform.

This section describes the Platform Certificate Information Model. The Platform Certificate Information Model abstractly defines the information fields in a Platform Certificate.

End of informative comment

2.1 Platform Certificate

Start of informative comment

A Platform Certificate asserts that a specific platform contains one or more unique Roots of Trust (TPM, DICE, etc.), Trusted Building Block(s), and a specific set of components. As of this Platform Certificate Profile specification 2.1, the Root of Trust (RoT) is not limited to the TPM Endorsement Key.

A Trusted Building Block (TBB) consists of the parts of the RoT that do not have shielded locations or protected capabilities. Normally, this includes just the Core Root of Trust for Measurement (CRTM) and the TPM initialization functions. The definition of a TBB is typically platform-specific. One example of a TBB, specific to the PC Client platform, is the combination of CRTM, connection of the CRTM storage to the motherboard, and mechanisms for determining Physical Presence.

Platform Certificates contain assertions about trust made by a platform manufacturer. The Base certificate asserts the platform's security properties and configuration as shipped. Delta Platform Certificates are used to reflect platform changes made by system integrators, resellers, and other entities after the platform has left the manufacturer's facility.

Platform Certificates may be in the form of a Key Certificate compliant with RFC 5280 [14] or may be in the form of an Attribute Certificate compliant with RFC 5755 [12]. An Attribute Certificate is warranted when the attributes listed in the certificate have a different lifecycle than the key they would be bound to in a Key Certificate. It may not be necessary or practical to issue a new Platform Certificate, in the form of a Key Certificate, whenever an attribute of the platform - such as the addition, removal of change of a component - changes. A Key certificate can be used when there is no impediment to Proof of Possession as defined by PKCS #10 [26], for instance, in the case of a platform with an IDevID key and associated certificate. By definition, the IDevID key can sign and thus can provide proof of possession to a certifying authority. In contrast, a PC Client platform with a TPM complying with the TCG Endorsement Key Credential Profile Specification contains an EK that is a restricted decryption key and thus cannot provide Proof of Possession using a PKCS #10 Certificate Signing Request.

End of informative comment

2.1.1 Who Uses a Platform Certificate?

Start of informative comment

One consumer of a Platform Certificate can be a Verifier. A Platform Certificate contains information that the Verifier can use in verifying the integrity characteristics of a platform. Platform Certificates, including Delta Platform Certificates, are Endorsements that a Verifier uses when evaluating the Evidence provided by an Attester. A Verifier determines whether the issuer of a Platform Certificate or Delta Platform Certificate is trusted, and by extension whether the assertions contained therein can be trusted, based on the Appraisal Policy it uses that can be provided by the Relying Party.

Another consumer of a Platform Certificate can be an Attestation CA, which can copy field entries from the Platform Certificate to a new Platform or Attestation Key Certificate that the Attestation CA creates. The new certificates contain just the information the Attestation CA deems sufficient for the resulting certificate to meet its intended privacy level.

Another consumer of a Platform Certificate can be an Enterprise, which wishes to remotely provision multiple devices that belong to it. Typically, in this case, the Enterprise knows the serial numbers of the systems it owns, and the Platform Certificate is used to associate those serial numbers with particular EK [6] [7], IDevID or IAK certificates [8]. This way, for example, a VPN can be provisioned using the TPM to provide keys securely to clients of an Enterprise. In order to support this use case, the optional Platform Serial Number attribute needs to be included in the certificate. In addition, an Enterprise could use the Platform Certificate to assert non-security related properties, such as platform components, included optionally by the platform manufacturer in the certificate.

For other users of the Platform Certificate, refer to section 6.2 Platform Endorsement Credential of Reference Architecture for Interoperability Specification [2].

End of informative comment

2.1.2 Who Issues a Platform Certificate?

Start of informative comment

In general, any entity can issue a Platform Certificate: the initial Platform Certificate is typically issued by the platform manufacturer (for example, an OEM). An entity should not generate a Platform Certificate unless the entity is satisfied that the platform contains the Root of Trust referenced inside the certificate. Before using a Platform Certificate, relying entities need to establish trust in the issuer of the Platform Certificate. Such trust establishment mechanisms are out of the scope of this specification.

2.1.3 Types of Platform Certificates

A Platform Certificate can be a “Base”, “Delta” or “Rebase” Platform Certificate. A Base Platform Certificate is a Platform Certificate that does not reference any other Platform Certificate for a particular platform and contains the complete set of assertions that its issuer makes for this platform. A Delta Platform Certificate asserts specific changes made to the platform that are not reflected in the existing Platform Certificate. It requires and references a previously-issued Base or other Delta Platform Certificate for the complete set of platform assertions to be obtained. A Rebase Platform Certificate is functionally equivalent to a Base Platform Certificate in that it is a self-contained Platform Certificate that contains the complete set of assertions specified by its issuer. In addition, a Rebase Platform Certificate references a previously-issued Platform Certificate (either a Base or a Delta Platform Certificate) for the purpose of providing transparency to the operations and transformations previously applied to the platform.

Unless explicitly stated otherwise, all clauses applying to Base Platform Certificates in this specification also apply to Rebase Platform Certificates. To avoid tedious repetition, when the type of Platform Certificate is not explicitly identified in the text, a Base or Rebase Platform Certificate is implied. Delta Platform Certificates are always identified explicitly.

When an entity issues a Platform Certificate for a platform that already has a Platform Certificate or Delta Platform Certificate, linkage between the new Platform Certificate and existing Platform Certificate and/or Delta Platform Certificate can be provided. **Figure 1** illustrates the relationship between Platform Certificates issued by four different stakeholders in the lifecycle of a single platform. The System Integrator and Value-Added Retailer (VAR) link their Platform Certificate to the previous entity that modified the platform, thus creating a chain of custody in the supply chain. Later, the owner of the platform can issue their own Platform Certificate, with or without linkage, to the previously issued Platform Certificates. When deciding whether to reference previous Platform Certificates, an entity issuing a new Platform Certificate needs to understand that excluding references to previous Platform Certificates removes the visibility of the chain of custody through the supply chain.

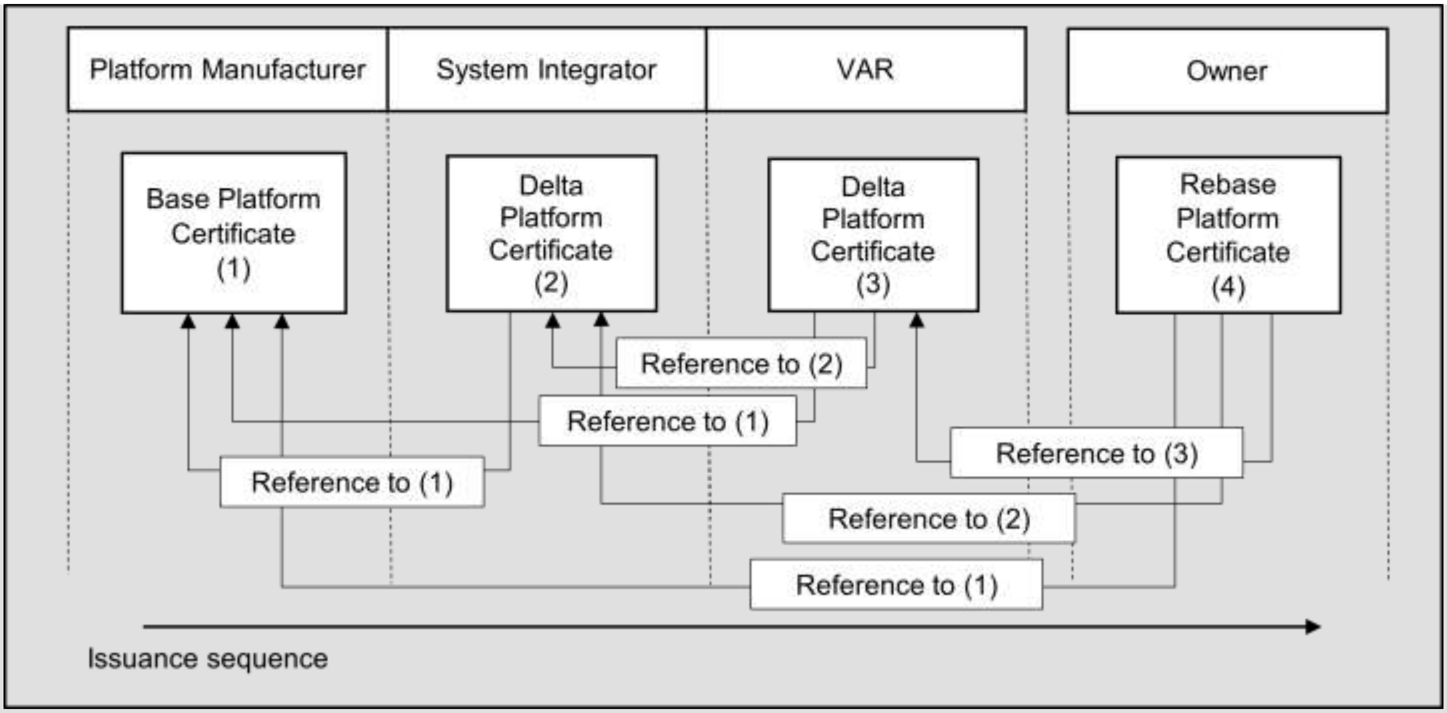

Figure 1: Platform Certificate Relationships

Other types of entities in the platform manufacturing supply chain could issue a Platform Certificate or a Delta Platform Certificate. For more information, refer to section 3 The Trusted Platform Lifecycle of Reference Architecture for Interoperability Specification [2].

End of informative comment

2.1.4 Contents of a Platform Certificate

Start of informative comment

A Platform Certificate contains assertions, made by the issuer, about the certificate itself and about the platform to which the certificate refers.

Table 1 specifies the Information Model of a Platform Certificate.

Certain information fields defined in the Platform Certificate information model can contain information that may be deemed privacy-sensitive in specific deployment scenarios. For example, the Platform Ownership field may be considered privacy-sensitive in a personal device use scenario, or privacy-insensitive in an enterprise deployment scenario. Given that such considerations may apply to different instances of the same class of computing equipment (i.e. different manufactured units of the same make and model of computer) this Platform Certificate Profile 2.1 specification does not distinguish information model fields according to the potential privacy impact of their contents. Binding specifications for specific classes of platforms or deployment scenarios may restrict the use of certain optional fields of the Platform Certificate Profile information model, or define specific guidelines for their use.

End of informative comment

Table 1: Platform Certificate Information Model (normative)

	Field Name	Description	Field Status
Assertions about the Platform Certificate	Certificate Type	A statement identifying the certificate type	REQUIRED
	Issuer	A statement identifying the issuer of the certificate	REQUIRED
	Certificate Specification	The Platform Certificate Specification to which the Platform Certificate is compliant	REQUIRED
	Validity Period	The time period when the certificate is valid	REQUIRED
	Signature	The signature of the issuer over the other fields	REQUIRED
	Policy Reference	A reference to the issuing policy of the Platform Certificate issuer	REQUIRED
	Previous Platform Certificate	A statement identifying existing Platform Certificates or Delta Platform Certificates for the platform	See section 2.1.4.6
	Revocation Locator	A statement identifying source of revocation status information	OPTIONAL
Assertions about the Platform	Cryptographic Anchors	A statement identifying the certificates or keys associated with the platform Root(s) of Trust	REQUIRED
	Platform Manufacturer	The name of the platform manufacturer	REQUIRED
	Platform Model	A manufacturer-specific identifier	REQUIRED
	Platform Version	A manufacturer-specific identifier	REQUIRED
	Platform Specification	The platform Specification to which the platform is compliant	REQUIRED
	Platform Serial Number	The platform's unique serial number	OPTIONAL
	Platform Assertions	Security assertions about the platform	OPTIONAL
	Platform Configuration	Non-security related platform properties	OPTIONAL
	Platform Ownership	A statement identifying the owner of the platform	OPTIONAL
	Manufacturing Assertions	Assertions about the platform's manufacturing and supply chain	OPTIONAL

2.1.4.1 Certificate Type

Start of informative comment

This field identifies the type of the Platform Certificate (either a Platform Certificate or a Delta Platform Certificate) and its encoding (X.509 Attribute certificate or Public Key certificate).

End of informative comment

2.1.4.2 Issuer

Start of informative comment

This field identifies the entity that signed and issued the Platform Certificate.

End of informative comment

2.1.4.3 Certificate Specification

Start of informative comment

This field identifies the Platform Certificate Profile Specification version, which includes this specification's Version, Level, and Revision.

End of informative comment

2.1.4.4 Validity Period

Start of informative comment

This field enables the certificate user to determine whether the Platform Certificate has begun to be valid or has expired.

A Platform Certificate is not expected to expire during the normal life expectancy of the platform, but it may become unrepresentative of the state of the platform when maintenance occurs.

End of informative comment

2.1.4.5 Signature

Start of informative comment

This field contains the signature of the issuer over the other fields in the certificate.

End of informative comment

2.1.4.6 Previous Platform Certificate

Start of informative comment

This field identifies the Platform Certificates previously issued to the platform.

A new Platform Certificate can capture any change to the platform, such as security assertion, cryptographic anchor, or component changes, without any dependencies on the previous Platform Certificate.

End of informative comment

The Previous Platform Certificate field SHALL only be included when the Certificate Type identifies the Platform Certificate as referencing another Platform Certificate.

When present, this SHALL contain unambiguous references to the previously issued Platform Certificates, encoded as defined in Section 3.3.11.

2.1.4.7 Policy Reference

Start of informative comment

This field enables the certificate user to identify the certificate issuance policy of the Platform Certificate issuer. For example, an issuer of a Platform Certificate that references another Platform Certificate or Delta Platform Certificate can indicate that the previous Platform Certificate or Delta Platform Certificate was valid at the time of issuance of the new Platform Certificate.

End of informative comment

2.1.4.8 Revocation Locator

Start of informative comment

This field enables the certificate consumer to determine whether the Platform Certificate has been revoked and should no longer be used as the basis for a trust decision.

A Platform Certificate could be revoked by the platform manufacturer if there is evidence of CA compromise. Other reasons for revocation include replacement of a platform's TPM, replacement of the Endorsement Key, or reissuance of the EK certificate. Platform configuration changes made after the platform is shipped can be addressed by the issuance of a Delta Platform Certificate.

End of informative comment

2.1.4.9 Cryptographic Anchors

Start of informative comment

The Cryptographic Anchors attribute is used by a Verifier to determine whether the platform contains one or more unique RoTs referenced by the Platform Certificate. The RoT can be a TPM, or a component with equivalent security properties provided through technologies such as DICE or MARS.

When the cryptographic anchor is a TPM, this assertion can reference the EK Certificates since the "TCG Infrastructure Working Group Reference Architecture for Interoperability (Part I)" [2] requires the TPM Manufacturer to issue an EK Certificate for each TPM Endorsement Key. The Platform Certificate contains references to all TCG required Endorsement Key (EK) Certificates. The Platform Certificate can also contain references to optional EK Certificates, such as those issued by the Platform OEM or Platform Owner. Alternatively, this assertion can reference the IDevID and IAK certificates specified in [8].

When the cryptographic anchor is based on a different technology, for example DICE [34] or MARS [35], this assertion can reference a certificate and/or key that the Platform Certificate issuer determines to act as a cryptographic anchor for the Platform Certificate. The certificate and/or key should have comparable properties to a TPM EK or IDevID certificate, for example a DICE DeviceID certificate endorsed by the DICE manufacturer's Root CA.

The cryptographic anchor attribute can contain any cryptographic certificate or key that can identify the platform.

End of informative comment

2.1.4.10 Platform Manufacturer

Start of informative comment

This assertion identifies the platform manufacturer, which can be expressed as a string or a globally unique and verifiable value, such as an IANA Private Enterprise Number identifier.

End of informative comment

2.1.4.11 Platform Model

Start of informative comment

This assertion identifies the specific model of the platform. This is used, in combination with the Platform Version, by a Verifier to assess that the platform contains specific root of trust implementations.

The platform model is manufacturer-specific.

End of informative comment

2.1.4.12 Platform Version

Start of informative comment

This assertion identifies the specific version of the platform. This is used, in combination with the Platform Model, by a Verifier to assess that the platform contains specific root of trust implementations.

The platform version is manufacturer-specific.

End of informative comment

2.1.4.13 Platform Specification

Start of informative comment

This assertion identifies the relevant TCG platform specific specification to which the platform was designed. This describes the platform class as well as the major and minor version number and the revision level.

End of informative comment

2.1.4.14 Platform Serial Number

Start of informative comment

This assertion is a value that, in combination with other assertions such as the platform manufacturer and platform model, uniquely identifies the platform. This is used by the verifier to correlate the certificate to a physical platform.

The Platform Serial Number is manufacturer-specific.

End of informative comment

When present, this field SHALL contain a customer-visible serial number as the identifier.

Start of informative comment

Even though this field is optional, the field will be included when enabling Enterprise use cases.

End of informative comment

2.1.4.15 Platform Assertions

Start of informative comment

This field contains assertions about the general security properties of the platform. This is used by a Verifier to assess that the platform implements acceptable security policies.

For more information, see Section 5 Entities, Assertions and Signed Structures in [2].

End of informative comment

2.1.4.16 Platform Configuration

Start of informative comment

These properties could include platform component information such as serial numbers, network adapter MAC addresses, motherboard serial number and Uniform Resource Identifier where valid PCR and platform configuration information can be obtained.

End of informative comment

2.1.4.17 Platform Ownership

Start of informative comment

This field could contain assertions of ownership or usage of the platform.

End of informative comment

2.1.4.18 Manufacturing Assertions

Start of informative comment

This field contains assertions about the manufacturing and supply chain properties of the platform. This information can be used to identify supply chain processes and actors for assessing compliance or addressing supply chain risks.

End of informative comment

2.2 Delta Platform Certificate

Start of informative comment

A system integrator or VAR can make modifications to a platform resulting in the Platform Certificate inaccurately reflecting its current configuration. A Delta Platform Certificate asserts specific changes made to the platform that are not reflected in the original Platform Certificate.

The entity making platform modifications could issue a Delta Platform Certificate to reflect those changes. A chain consisting of a Platform Certificate followed by multiple Delta Platform Certificates is supported in cases where multiple entities make valid modifications to a platform. A Delta Platform Certificate only includes additions, modifications and deletions of certain platform attributes. The issuer of the Delta Platform Certificate endorses that the changes it made to the platform are adequately represented by the Delta Platform Certificate and that the Delta Platform Certificate references the appropriate Base/Rebase Platform Certificate or previously-issued Delta Platform Certificate.

Figure 2 illustrates how a chain of Platform Certificate and Delta Platform Certificates can be constructed by linking the certificates using a base certificate reference.

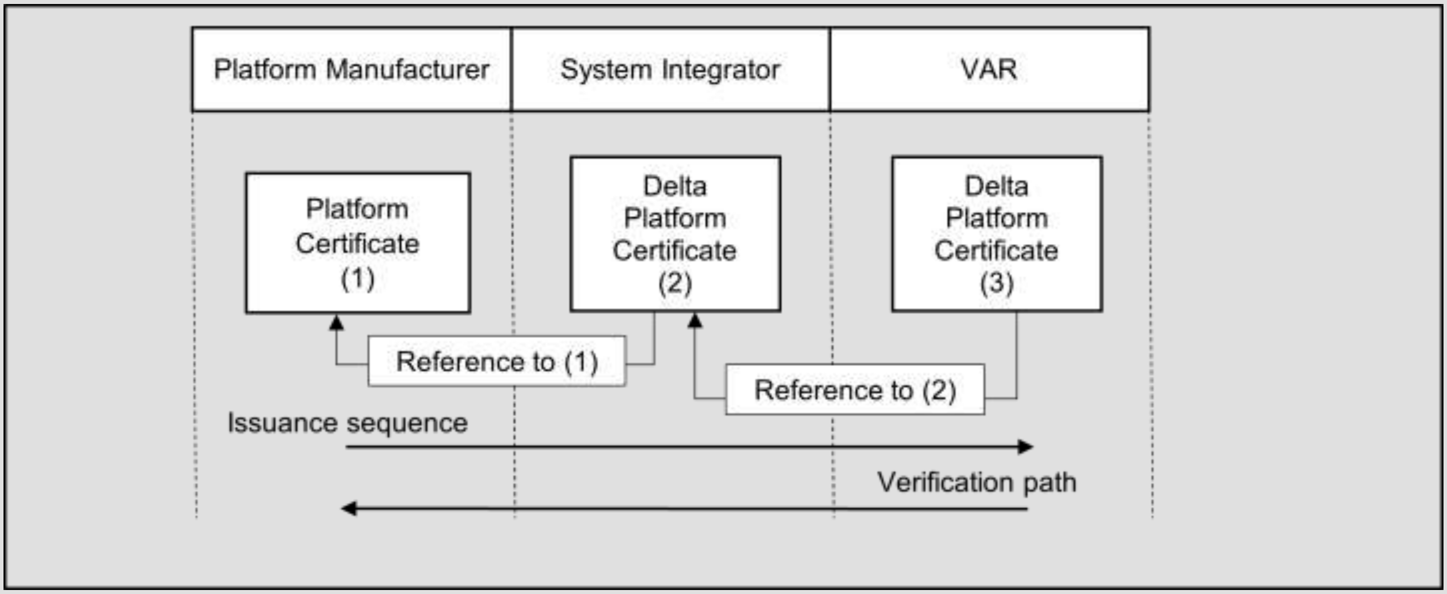

Figure 2: Delta Platform Certificate chain

A Platform Certificate can reference a Base Platform Certificate and its associated Delta Platform Certificates using the Previous Platform Certificate field, as illustrated in **Figure 3**. Note that “Platform Certificate (4)” in **Figure 3** is a Rebase Platform Certificate, as it references previously-issued Base and Delta Platform Certificates.

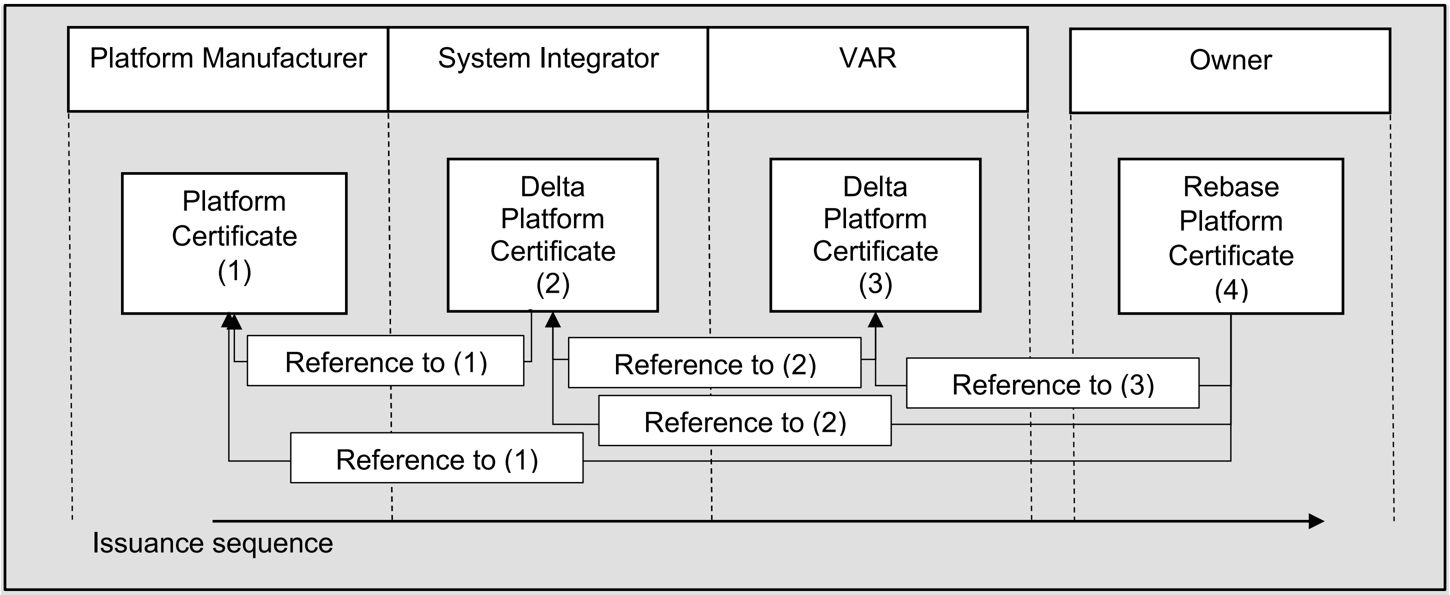

Figure 3: Platform Certificate referencing a Platform Certificate and associated Delta Platform Certificates

End of informative comment

2.2.1 Who Uses a Delta Platform Certificate?

Start of informative comment

A Delta Platform Certificate may be issued when an authorized change is made to the platform. A Delta Platform Certificate is used by Verifiers, Attestation CAs and Enterprises wanting to verify authorized changes in platform attributes.

End of informative comment

2.2.2 Who Issues a Delta Platform Certificate?

Start of informative comment

Any authorized entity, typically a system integrator, VAR or the platform's owner that modifies a platform's configuration can issue a Delta Platform Certificate to reflect platform attribute changes. Entities using a Delta Platform Certificate need to establish trust in the issuing entity. Such trust establishment mechanisms are out of the scope of this specification.

End of informative comment

2.2.3 Requirements for Issuing a Delta Platform Certificate

Start of informative comment

A Delta Platform Certificate can be issued if the following conditions are satisfied:

- Changes made to the platform do not invalidate the TBB security claims made by the original platform manufacturer.
- Changes made to the platform do not invalidate the TCG Platform Specification compliance claims made by the platform manufacturer.
- The platform's RoTs as identified by a previous Platform Certificate or Delta Platform Certificate are not altered or replaced. For example, the issuer may not call TPM2_ChangeEPS on a TPM. Doing so would break the binding between the Base Platform Certificate and the TPM.

If any one of those conditions is not true, then a new Base or Rebase Platform Certificate is recommended to be issued. Ultimately, a Verifier implements its own policy to determine whether the issuer of a Delta Platform Certificate can be trusted to change assertions present in the Base Platform Certificate.

The Delta Platform Certificate can include references to new Cryptographic Anchors, as long as the Delta Platform Certificate does not invalidate (i.e., change the values, or remove) the Cryptographic Anchors listed in the Base Platform Certificate and any prior Delta Platform Certificates.

End of informative comment

Any Delta Platform Certificate created SHALL adhere to the following set of requirements:

1. The Delta Platform Certificate SHALL list the same Platform Manufacturer, Platform Model and Platform Serial Number listed in the Base Platform Certificate.
2. The Delta Platform Certificate SHALL NOT change or remove the Cryptographic Anchors listed in the Base Platform Certificate.
3. The Delta Platform Certificate SHALL NOT change or remove the Cryptographic Anchors defined by a prior Delta Platform Certificate.

2.2.4 Contents of a Delta Platform Certificate

Start of informative comment

A Delta Platform Certificate contains assertions, made by the issuer, about the certificate itself and about the platform to which the certificate refers.

Table 2 specifies the Information Model of a Delta Platform Certificate.

End of informative comment

Table 2: Delta Platform Certificate Information Model (normative)

	Field Name	Description	Field Status
Assertions about the Delta Platform Certificate	Certificate Type	A statement identifying the certificate type	REQUIRED
	Issuer	A statement identifying the issuer of certificate	REQUIRED
	Validity Period	The time period when the certificate is valid	REQUIRED
	Signature	The signature of the issuer over the other fields	REQUIRED
	Previous Platform Certificate	A statement identifying the prior Base, Rebase or Delta Platform Certificate	REQUIRED
	Policy Reference	A reference to the issuing policy of the Platform Certificate issuer	REQUIRED
	Certificate Specification	The Platform Certificate Specification to which the Delta Platform Certificate is compliant	REQUIRED
	Revocation Locator	A statement identifying source of revocation status information	OPTIONAL
Assertions about the Platform	Platform Manufacturer	The name of the platform manufacturer	REQUIRED
	Platform Model	A manufacturer-specific identifier	REQUIRED
	Platform Version	A manufacturer-specific identifier	REQUIRED

	Field Name	Description	Field Status
	Cryptographic Anchors	A statement identifying the newly issued certificates or keys associated with the platform Root of Trust	OPTIONAL
	Platform Specification	The platform Specification to which the platform is compliant	OPTIONAL
	Platform Serial Number	The platform's unique serial number	OPTIONAL
	Platform Assertions	Security assertions about the platform	OPTIONAL
	Platform Configuration	Non-security related platform properties	OPTIONAL
	Platform Ownership	A statement identifying the owner of the platform	OPTIONAL

2.2.4.1 Certificate Type

Start of informative comment

This field identifies the type of the Platform Certificate (in this case, a Delta Platform Certificate) and its encoding (X509 Attribute certificate or Public Key certificate).

End of informative comment

2.2.4.2 Issuer

Start of informative comment

This field identifies the entity that signed and issued the Delta Platform Certificate.

End of informative comment

2.2.4.3 Validity Period

Start of informative comment

This field enables the certificate user to determine whether the Delta Platform Certificate has begun to be valid or has expired.

End of informative comment

The validity period's "Not After" date SHALL match that of the Base Platform Certificate.

2.2.4.4 Signature

Start of informative comment

This field is the signature of the issuer over the other fields in the certificate.

End of informative comment

2.2.4.5 Previous Platform Certificate

Start of informative comment

This field enables the verifier to bind the certificate to the previously issued Base (or Rebase) Platform Certificate or Delta Platform Certificate.

End of informative comment

The Previous Platform Certificate field SHALL contain an unambiguous reference to the Base or Rebase Platform Certificate directly or indirectly through another Delta Platform Certificate.

2.2.4.6 Certificate Specification

Start of informative comment

This field identifies the Platform Certificate Profile Specification version, which includes the Platform Certificate Profile specification's Version, Level, and Revision.

The value of this field is identical to the one in the Base Platform Certificate this Delta Platform Certificate directly or indirectly references. If a different version of the Certificate Specification is to be used, then a new Base Platform Certificate or Rebase Platform Certificate is appropriate.

End of informative comment

This field SHALL equal that of the Base Platform Certificate.

2.2.4.7 Policy Reference

Start of informative comment

This field enables the certificate user to identify the certificate issuance policy of the Delta Platform Certificate issuer. For example, an issuer of a Delta Platform Certificate that references a Base Platform Certificate or Delta Platform Certificate can indicate that the Base Platform Certificate or Delta Platform Certificate was valid at the time of issuance of the new Delta Platform Certificate.

End of informative comment

2.2.4.8 Revocation Locator

Start of informative comment

This field enables the certificate consumer to determine whether the Delta Platform Certificate has been revoked and should no longer be used as the basis for a trust decision.

A Delta Certificate could be revoked if there is evidence of CA compromise, or in cases where the Base Platform Certificate or prior Delta Platform Certificate are revoked.

End of informative comment

2.2.4.9 Platform Manufacturer

Start of informative comment

This field identifies the platform manufacturer, which can be expressed as a string or a globally unique and verifiable value such as an IANA identifier.

End of informative comment

This field SHALL equal that of the Base Platform Certificate.

2.2.4.10 Platform Model

Start of informative comment

This field identifies the specific platform model implementation. This is used, in combination with the Platform Version, by a Verifier to assess whether the platform contains a specific root of trust implementation.

The platform model is manufacturer-specific.

End of informative comment

This field SHALL equal that of the Base Platform Certificate.

2.2.4.11 Platform Version

Start of informative comment

This field identifies the specific version of the platform. This is used, in combination with the Platform Model, by a Verifier to assess whether the platform contains a specific root of trust implementation.

The platform version is manufacturer-specific.

End of informative comment

This field SHALL equal that of the Base Platform Certificate.

2.2.4.12 Cryptographic Anchors

Start of informative comment

This field is used to reference new certificates or keys issued by the Delta Platform Certificate issuer that are not present in the Base Platform Certificate or prior Delta Platform Certificate.

End of informative comment

If present in the Delta Platform Certificate, this field SHALL contain references to the unique Cryptographic Anchors asserted by the issuer of this Delta Platform Certificate that are not listed in the Base Platform Certificate or prior Delta Platform Certificate.

2.2.4.13 Platform Serial Number

Start of informative comment

This field holds a value that, in combination with other assertions such as the platform manufacturer and platform model, uniquely identifies the platform. This is used by the verifier to correlate the certificate to a physical platform.

The Platform Serial Number is manufacturer-specific.

End of informative comment

If present in the Delta Platform Certificate, this field SHALL equal that of the Base Platform Certificate.

2.2.4.14 Platform Configuration

Start of informative comment

This field contains assertions of properties that are not security related.

The Delta Platform Certificate can only include platform properties that have changed (added, modified, or deleted) with respect to the previously-issued Base or Delta Platform Certificate.

End of informative comment

2.2.4.15 Platform Ownership

Start of informative comment

This field contains assertions about ownership or usage of the platform.

End of informative comment

2.2.4.16 Manufacturing Assertions

Start of informative comment

This field contains assertions about the manufacturing and supply chain properties of the platform as they relate to processes that occurred after issuance of the Base Platform Certificate. This information can be used to identify supply chain processes and actors for assessing compliance or addressing supply chain risks.

End of informative comment

3 TCG Platform Certificate X.509 encoding

Start of informative comment

This section defines the specification of a TCG Platform Certificate instantiated as an X.509 certificate. All the common and information fields required by this specification are defined in ASN.1 and encoded using DER.

Tables 3-6 define how the information model elements identified in Chapter 2 for Platform Certificates and Delta Platform Certificates are encoded as Attribute Certificate (AC) and Public Key Certificate (PKC) X.509 certificates.

A future version of this specification might explicitly disallow the modification of existing sequences and attributes.

End of informative comment

3.1 X.509 Attribute Certificate encoding

Start of informative comment

This section contains the definition of the X.509 Attribute Certificate encoding, version 3.

RFC5755 [12] and the ITU-T X.509 specification [27] defines the syntax of an attribute certificate as follows:

```
AttributeCertificate ::= SIGNED{TBSAttributeCertificate}

TBSAttributeCertificate ::= SEQUENCE {
    version                AttCertVersion, -- version is v2
    holder                 Holder,
    issuer                 AttCertIssuer,
    signature              AlgorithmIdentifier{{SupportedAlgorithms}},
    serialNumber           CertificateSerialNumber,
    attrCertValidityPeriod AttCertValidityPeriod,
    attributes             SEQUENCE OF Attribute{{SupportedAttributes}},
    issuerUniqueID         UniqueIdentifier OPTIONAL,
    ...,
    ...,
    extensions OPTIONAL }
```

Unless stated otherwise, the **`Attribute`** structures defined in this specification are included in the **`attributes`** component.

The fields defined in Table 1 and Table 2 of the Information Model of this specification need to be encoded in one of the existing **`TBSAttributeCertificate`** components such as the **`holder`**, **`attributes`** or **`extensions`** components. This specification makes use of existing **`components`**, **`Attribute(s)`** and **`extensions`** defined in [27] and [12], and specifies new **`Attribute(s)`** and the required ASN.1 structures.

Table 3 and Table 4 specify the mapping between the Information Model's fields and their encoding, version 3, in a X.509 Attribute Certificate.

End of informative comment

Table 3: X.509 Attribute Certificate Encoding (version 3) of the Platform Certificate Information Model (normative)

	Information Model’s Field Name	Field Status	X.509 Attribute Certificate encoding, version 3
Assertions about the Platform Certificate	Certificate Type	REQUIRED	`tCGCredentialType` attribute
	Issuer	REQUIRED	`issuer` component and `authorityKeyIdentifier` extension
		OPTIONAL	`authorityInfoAccess` extension
		PROHIBITED	`issuerUniqueID` component
	Certificate Specification	REQUIRED	`tCGCredentialSpecification` attribute
	Validity Period	REQUIRED	`attrCertValidityPeriod` component
	Signature	REQUIRED	`signature` component, `signatureAlgorithm` and `signatureValue` fields
	Policy Reference	REQUIRED	`certificatePolicies` extension
	Previous Platform Certificate	See section 2.1.4.6	`previousPlatformCertificates` attribute
	Revocation Locator	OPTIONAL	`cRLDistributionPoints` extension
Assertions about the Platform	Cryptographic Anchors	REQUIRED	`holder` component
		OPTIONAL	`cryptographicAnchors` attribute
	Platform Manufacturer	REQUIRED	`subjectAltName` extension
	Platform Model	REQUIRED	`subjectAltName` extension
	Platform Version	REQUIRED	`subjectAltName` extension
	Platform Specification	REQUIRED	`tCGPlatformSpecification` attribute
	Platform Serial Number	OPTIONAL	`subjectAltName` extension
	Platform Assertions	OPTIONAL	`tBBSecurityAssertions` attribute
	Platform Configuration	OPTIONAL	`platformConfiguration` and `platformConfigUri` attributes
	Platform Ownership	OPTIONAL	`platformOwnership` attribute
	Manufacturing Assertions	OPTIONAL	`manufacturingAssertions` attribute

Table 4: X.509 Attribute Certificate Encoding (version 3) of the Delta Platform Certificate Information Model (normative)

	Information Model's Field Name	Field Status	X.509 Attribute Certificate encoding, version 3
Assertions about the Delta Platform Certificate	Certificate Type	REQUIRED	`tCGCredentialType` attribute
	Issuer	REQUIRED	`issuer` component and `authorityKeyIdentifier` extension
		OPTIONAL	`authorityInfoAccess` extension
		PROHIBITED	`issuerUniqueID` component
	Validity Period	REQUIRED	`attrCertValidityPeriod` component
	Signature	REQUIRED	`signature` component, `signatureAlgorithm` and `signatureValue` fields
	Policy Reference	REQUIRED	`certificatePolicies` extension
	Previous Platform Certificate	REQUIRED	`previousPlatformCertificates` attribute
	Certificate Specification	REQUIRED	`tCGCredentialSpecification` attribute
	Revocation Locator	OPTIONAL	`cRLDistributionPoints` extension
Assertions about the Platform	Platform Manufacturer	REQUIRED	`subjectAltName` extension
	Platform Model	REQUIRED	`subjectAltName` extension
	Platform Version	REQUIRED	`subjectAltName` extension
	Platform Serial Number	OPTIONAL	`subjectAltName` extension
	Platform Specification	OPTIONAL	`tCGPlatformSpecification` attribute
	Cryptographic Anchors	REQUIRED	`holder` component
		OPTIONAL	`cryptographicAnchors` attribute
	Platform Assertions	OPTIONAL	`tBBSecurityAssertions` attribute
	Platform Configuration	OPTIONAL	`platformConfiguration` and `platformConfigUri` attributes
	Platform Ownership	OPTIONAL	`platformOwnership` attribute
	Manufacturing Assertions	OPTIONAL	`manufacturingAssertions` attribute

3.2 X.509 Public Key Certificate encoding

Start of informative comment

This section contains the definition of the X.509 Public Key Certificate encoding, version 2.

RFC5280 [14] and the ITU-T X.509 specification [27] defines the syntax of a key certificate as follows:

```
Certificate ::= SIGNED {TBSCertificate}

   TBSCertificate  ::=  SEQUENCE  {
        version         [0]  EXPLICIT Version DEFAULT v1,
        serialNumber         CertificateSerialNumber,
        signature            AlgorithmIdentifier,
        issuer               Name,
        validity             Validity,
        subject              Name,
        subjectPublicKeyInfo SubjectPublicKeyInfo,
        issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                             -- If present, version MUST be v2 or v3
        subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                             -- If present, version MUST be v2 or v3
        extensions      [3]  EXPLICIT Extensions OPTIONAL
                             -- If present, version MUST be v3
        }
```

The fields defined in Table 1 and Table 2 of the Information Model of this specification need to be encoded in one of the existing **`TBSCertificate`** components such as the **`subjectPublicKeyInfo`** and **`extensions`** components. This specification makes use of existing **`components`** and **`extensions`** defined in [27] and [14], and specifies new **`Extensions`** and the required ASN.1 structures.

Table 5 and Table 6 specify the mapping between the Information Model's fields and their encoding, version 2, in an X.509 Public Key Certificate.

Unless stated otherwise, the **`Attribute`** structures defined in this specification are included in the **`subjectDirectoryAttributes`** component.

End of informative comment

Table 5: X.509 Public Key Certificate Encoding (version 2) of the Platform Certificate Information Model (normative)

	Information Model's Field Name	Field Status	X.509 Public Key Certificate encoding
Assertions about the Platform Certificate	Certificate Type	REQUIRED	**`tCGCredentialType`** attribute in the **`subjectDirectoryAttribute`** extension
	Issuer	REQUIRED	**`issuer`** component and **`authorityKeyIdentifier`** extension
		OPTIONAL	**`authorityInfoAccess`** extension
		PROHIBITED	**`issuerUniqueID`** component
	Certificate Specification	REQUIRED	**`tCGCredentialSpecification`** attribute in the **`subjectDirectoryAttribute`** extension
	Validity Period	REQUIRED	**`validity`** component
	Signature	REQUIRED	**`signature`** component, **`signatureAlgorithm`** and **`signatureValue`** fields

	Information Model's Field Name	Field Status	X.509 Public Key Certificate encoding
	Policy Reference	REQUIRED	**`certificatePolicies`** extension
	Previous Platform Certificate	See section 2.1.4.6	**`previousPlatformCertificates`** attribute in the **`subjectDirectoryAttribute`** extension
	Revocation Locator	OPTIONAL	**`cRLDistributionPoints`** extension
Assertions about the Platform	Cryptographic Anchors	REQUIRED	**`subjectPublicKeyInfo`** component
		OPTIONAL	**`cryptographicAnchors`** attribute in the **`subjectDirectoryAttribute`** extension
	Platform Manufacturer	REQUIRED	**`subjectAltName`** extension
	Platform Model	REQUIRED	**`subjectAltName`** extension
	Platform Version	REQUIRED	**`subjectAltName`** extension
	Platform Specification	REQUIRED	**`tCGPlatformSpecification`** attribute in the **`subjectDirectoryAttribute`** extension
	Platform Serial Number	OPTIONAL	**`subjectAltName`** extension
	Platform Assertions	OPTIONAL	**`tBBSecurityAssertions`** attribute in the **`subjectDirectoryAttribute`** extension
	Platform Configuration	OPTIONAL	**`platformConfiguration`** and **`platformConfigUri`** attributes in the **`subjectDirectoryAttribute`** extension
	Platform Ownership	OPTIONAL	**`platformOwnership`** attribute in the **`subjectDirectoryAttribute`** extension
	Manufacturing Assertions	OPTIONAL	**`manufacturingAssertions`** attribute in the **`subjectDirectoryAttribute`** extension

Table 6: X.509 Public Key Certificate Encoding (version 2) of the Delta Platform Certificate Information Model (normative)

	Information Model's Field Name	Field Status	X.509 Public Key Certificate encoding, version 3
Assertions about the Delta Platform Certificate	Certificate Type	REQUIRED	**`tCGCredentialType`** attribute in the **`subjectDirectoryAttribute`** extension
	Issuer	REQUIRED	**`issuer`** component and **`authorityKeyIdentifier`** extension
		OPTIONAL	**`authorityInfoAccess`** extension
		PROHIBITED	**`issuerUniqueID`** component
	Validity Period	REQUIRED	**`validity`** component

	Information Model's Field Name	Field Status	X.509 Public Key Certificate encoding, version 3
	Signature	REQUIRED	`signature` component, `signatureAlgorithm` and `signatureValue` fields
	Policy Reference	REQUIRED	`certificatePolicies` extension
	Previous Platform Certificate	REQUIRED	`previousPlatformCertificates` attribute
	Certificate Specification	REQUIRED	`tCGCredentialSpecification` attribute in the `subjectDirectoryAttribute` extension
	Revocation Locator	OPTIONAL	`cRLDistributionPoints` extension
Assertions about the Platform	Cryptographic Anchors	REQUIRED	`subjectAltName` extension
		OPTIONAL	`cryptographicAnchors` attribute in the `subjectDirectoryAttribute` extension
	Platform Manufacturer	REQUIRED	`subjectAltName` extension
	Platform Model	REQUIRED	`subjectAltName` extension
	Platform Version	REQUIRED	`subjectAltName` extension
	Platform Serial Number	OPTIONAL	`subjectAltName` extension
	Platform Specification	OPTIONAL	`tCGPlatformSpecification` attribute in the `subjectDirectoryAttribute` extension
	Platform Assertions	OPTIONAL	`tBBSecurityAssertions` attribute in the `subjectDirectoryAttribute` extension
	Platform Configuration	OPTIONAL	`platformConfiguration` and `platformConfigUri` attributes in the `subjectDirectoryAttribute` extension
	Platform Ownership	OPTIONAL	`platformOwnership` attribute in the `subjectDirectoryAttribute` extension
	Manufacturing Assertions	OPTIONAL	`manufacturingAssertions` attribute in the `subjectDirectoryAttribute` extension

3.3 TCG Platform Certificate fields, attributes, and extensions

Start of informative comment

This section defines how information is encoded in a TCG Platform Certificate. It specifies the encoding for both X.509 AC and X.509 PKC encodings. Where differences exist between the two types of encodings, they are clearly defined.

Fields not specified here but required by an X509 certificate, such as `version` and `serialNumber`, are defined in the RFC 5280 specification [14].

End of informative comment

3.3.1 tCGCredentialType attribute

Start of informative comment

This attribute identifies the type of the Platform Certificate: it can be either a Platform Certificate (Base or Rebase) or a Delta Platform Certificate. The values are encoded as TCG registered OIDs.

End of informative comment

A Platform Certificate that does not reference another Platform Certificate or Delta Platform Certificate SHALL list **tcg-kp-PlatformAttributeCertificate** if it is an AC, or SHALL list **tcg-kp-PlatformKeyCertificate** if it is a PKC in its **tCGCredentialType** attribute.

A Platform Certificate that references another Platform Certificate or Delta Platform Certificate SHALL list **tcg-kp-AdditionalPlatformAttributeCertificate** if the reference certificate is an AC, or SHALL list **tcg-kp-AdditionalPlatformKeyCertificate** if the reference certificate is a PKC in its **tCGCredentialType** attribute.

A Delta Platform Certificate SHALL list **tcg-kp-DeltaPlatformAttributeCertificate** if the reference certificate is an AC, or SHALL list **tcg-kp-DeltaPlatformKeyCertificate** if the reference certificate is a PKC in its **tCGCredentialType** attribute.

```
tCGCredentialType ATTRIBUTE ::= {
    WITH SYNTAX TCGCredentialType
    ID          tcg-at-tcgCredentialType}

TCGCredentialType::= SEQUENCE {
    certificateType CredentialType}

CredentialType ::= OBJECT IDENTIFIER (tcg-kp-PlatformAttributeCertificate | tcg-kp-
    PlatformKeyCertificate |
    tcg-kp-AdditionalPlatformAttributeCertificate | tcg-kp-AdditionalPlatformKeyCertificate |
    tcg-kp-DeltaPlatformAttributeCertificate | tcg-kp-DeltaPlatformKeyCertificate)
)
```

3.3.2 issuer component

Start of informative comment

This component contains the distinguished name of the entity that issued the Platform Certificate. This is the entity that asserts that a specific platform contains one or more unique Root(s) of Trust (TPM, DICE, etc.) and Trusted Building Block(s) (TBB) and a specific set of components in a manner that conforms to the relevant TCG Platform Specific specification.

End of informative comment

3.3.3 authorityKeyIdentifier extension

Start of informative comment

This extension identifies the subject public key of the certificate issuer.

As a reminder to the Reader, an **Extension** is defined in [14] as follows.

```
Extension ::= SEQUENCE  {
    extnID    OBJECT IDENTIFIER,
    critical  BOOLEAN DEFAULT FALSE,
    extnValue OCTET STRING
              -- contains the DER encoding of an ASN.1 value
              -- corresponding to the extension type identified
              -- by extnID
```

```
        }

        -- ISO arc for standard certificate and CRL extensions
        id-ce OBJECT IDENTIFIER  ::=  {joint-iso-ccitt(2) ds(5) 29}
```

The **authorityKeyIdentifier** extension is defined in [14] as follows.

```
        id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::=  { id-ce 35 }

        AuthorityKeyIdentifier ::= SEQUENCE {
            keyIdentifier             [0] KeyIdentifier           OPTIONAL,
            authorityCertIssuer       [1] GeneralNames            OPTIONAL,
            authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL  }

        KeyIdentifier ::= OCTET STRING
```

End of informative comment

The **critical** field of the **authorityKeyIdentifier** extension SHALL be **FALSE**.

If the issuer's certificate contains a **subjectKeyIdentifier** extension, the **keyIdentifier** field of the **authorityKeyIdentifier** extension SHALL contain the **subjectKeyIdentifier** value from the issuer's certificate.

3.3.4 authorityInfoAccess extension

Start of informative comment

This extension contains information about how to access additional Certificate Authority related information about the issuer of the certificate and the services provided by that issuer.

The **authorityInfoAccess** extension is defined in [14] as follows:

```
        id-pe-authorityInfoAccess OBJECT IDENTIFIER ::= { id-pe 1 }

        AuthorityInfoAccessSyntax  ::=
            SEQUENCE SIZE (1..MAX) OF AccessDescription

        AccessDescription  ::=  SEQUENCE {
            accessMethod          OBJECT IDENTIFIER,
            accessLocation        GeneralName  }

        id-ad OBJECT IDENTIFIER ::= { id-pkix 48 }

        id-ad-caIssuers OBJECT IDENTIFIER ::= { id-ad 2 }

        id-ad-ocsp OBJECT IDENTIFIER ::= { id-ad 1 }
```

MAX is to be interpreted as described in [14], to mean the upper bound is unspecified.

End of informative comment

If an **authorityInfoAccess** extension is present, its **critical** field SHALL be **FALSE**.

If an **authorityInfoAccess** extension is present, the **accessMethod** field SHALL be set to **id-ad-ocsp** and the **accessLocation** field SHALL contain the URI of the OCSP responder.

3.3.5 issuerUniqueID component

Start of informative comment

This component uniquely identifies certificates that share names with other certificates issued by the same issuer. This component cannot be present in a Platform Certificate or Delta Platform Certificate.

End of informative comment

The **issuerUniqueID** component SHALL NOT be present in a Platform Certificate or Delta Platform Certificate.

3.3.6 tCGCredentialSpecification attribute

Start of informative comment

This attribute identifies the major version, minor version, and revision of the certificate specification with which a certificate is compliant. The values are encoded as three integers in this attribute.

End of informative comment

If a Platform Certificate references another Platform Certificate or Delta Platform Certificate, the **tCGCredentialSpecification** attribute SHALL be identical to the **tCGCredentialSpecification** attribute of the referenced Platform Certificate or Delta Platform Certificate.

```
tCGCredentialSpecification ATTRIBUTE ::= {
    WITH SYNTAX TCGSpecificationVersion
    ID          tcg-at-tcgCredentialSpecification }

TCGSpecificationVersion ::= SEQUENCE {
    majorVersion INTEGER,
    minorVersion INTEGER,
    revision     INTEGER }
```

3.3.7 attrCertValidityPeriod component

Start of informative comment

This component is applicable to only the AC encoding of the TCG Platform Certificate.

This component contains the period during which the binding between the assertions contained in the Platform Certificate and referenced Root(s) of Trust certificates is considered valid. It is represented by two date values named **notBefore** and **notAfter**. Issuers assign **notBefore** to the current time when the certificate is issued and **notAfter** to the last date that the certificate is considered valid. Both **notBefore** and **notAfter** use the appropriate time format as indicated by RFC 5755 [12], section 4.2.6 Validity Period.

End of informative comment

3.3.8 validity component

Start of informative comment

This component is applicable to only the PKC encoding of the TCG Platform Certificate.

This component contains the time frame during which the binding between the assertions contained in the Platform Certificate and RoT's Public Key(s) is considered valid. It is represented by two date values named **notBefore** and **notAfter**. Issuers should assign **notBefore** to the current time when the certificate is issued and **notAfter** to the last date that the certificate is considered valid. Both **notBefore** and **notAfter** use the appropriate time format as indicated by RFC 5280 [14]

End of informative comment

3.3.9 signature component, signatureAlgorithm and signatureValue fields

Start of informative comment

The **signature** component contains the OID that identifies the **algorithm**, used by the platform certificate issuer to sign the certificate. Optionally, the **signature** component also lists the relevant **parameters**. The **signature** component needs to match the **signatureAlgorithm** field.

The **`signatureAlgorithm`** field defined in [12] and [14] contains the OID that identifies the **`algorithm`**, used by the platform certificate issuer to sign the certificate. Optionally, the **`signature`** field also lists the relevant **`parameters`**. In [27] this field is named **`algorithmIdentifier`** in the **`SIGNATURE`** sequence.

The **`signatureValue`** field defined in [12] and [14] contains the bit string representing the signature’s value. In [27] this field is named **`signature`** in the **`SIGNATURE`** sequence.

End of informative comment

3.3.10 `certificatePolicies` extension

Start of informative comment

This extension defines the policy terms under which the Platform Certificate was issued.

The **`certificatePolicies`** extension is defined in [14] as follows.

```
id-ce-certificatePolicies OBJECT IDENTIFIER ::=  { id-ce 32 }

certificatePolicies ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier CertPolicyId,
    policyQualifiers SEQUENCE SIZE (1..MAX) OF
                         PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId PolicyQualifierId,
    qualifier         ANY DEFINED BY policyQualifierId }

id-pkix  OBJECT IDENTIFIER  ::=
    { iso(1) identified-organization(3) dod(6) internet(1)
            security(5) mechanisms(5) pkix(7) }

id-qt          OBJECT IDENTIFIER ::=  { id-pkix 2 }
id-qt-cps      OBJECT IDENTIFIER ::=  { id-qt 1 }
id-qt-unotice  OBJECT IDENTIFIER ::=  { id-qt 2 }

PolicyQualifierId ::= OBJECT IDENTIFIER ( id-qt-cps | id-qt-unotice )

Qualifier ::= CHOICE {
    cPSuri           CPSuri,
    userNotice       UserNotice }

CPSuri ::= IA5String

UserNotice ::= SEQUENCE {
     noticeRef        NoticeReference OPTIONAL,
     explicitText     DisplayText OPTIONAL }

NoticeReference ::= SEQUENCE {
     organization     DisplayText,
     noticeNumbers    SEQUENCE OF INTEGER }

DisplayText ::= CHOICE {
     ia5String        IA5String      (SIZE (1..200)),
     visibleString    VisibleString  (SIZE (1..200)),
     bmpString        BMPString      (SIZE (1..200)),
     utf8String       UTF8String     (SIZE (1..200)) }
```

End of informative comment

The **`critical`** field of the **`certificatePolicies`** extension SHALL be **`FALSE`**.

When the issuer of a Delta Platform Certificate or additional Platform Certificate has verified the validity of the Base and any previous Delta Platform Certificate, the OID `tcg-cap-verifiedPlatformCertificate` MAY be set in one of the `policyIdentifier` fields.

At least one `policyInformation` field SHALL be present and SHALL contain a `cPSuri`, identified by the `id-qt-cps` OID in the `policyQualifierId` field, and a `userNotice`, identified by the `id-qt-unotice` OID in the `policyQualifierId` field.

The `cPSuri` field SHALL contain an HTTP URL at which a plain language version of the platform endorsement entity's certificate policy could be obtained.

The `explicitText` field of `userNotice` SHALL contain "TCG Trusted Platform Endorsement" encoded as a UTF8String.

3.3.11 `previousPlatformCertificates` attribute

Start of informative comment

This attribute identifies the Platform Certificates or Delta Platform Certificates that were previously issued for this platform. The certificates are referenced using `Trait`, which is defined in section 4.13.2.

The `traitCategory` field is set to either `tcg-tr-cat-PlatformCertificate`, `tcg-tr-cat-DeltaPlatformCertificate` or `tcg-tr-cat-RebasePlatformCertificate` as appropriate to reflect the type of Platform Certificate the trait describes, as specified in Section 4.2.2.

Use of the `CertificateIdentifierTrait` `Trait` is recommended to enable verifier interoperability.

End of informative comment

The certificates referenced in the `previousPlatformCertificates` attribute SHALL appear in the `PreviousPlatformCertificates` sequence in the order they were issued.

A `Trait` in a `PreviousPlatformCertificates` sequence that represents a Base Platform Certificate SHALL use `tcg-tr-cat-PlatformCertificate` in its `traitCategory` field.

A `Trait` in a `PreviousPlatformCertificates` sequence that represents a Delta Platform Certificate SHALL use `tcg-tr-cat-DeltaPlatformCertificate` in its `traitCategory` field.

A `Trait` in a `PreviousPlatformCertificates` sequence that represents a Platform Certificate that references a previous Platform Certificate or Delta Platform Certificate SHALL use `tcg-tr-cat-RebasePlatformCertificate` in its `traitCategory` field.

```
previousPlatformCertificates ATTRIBUTE ::= {
    WITH SYNTAX PreviousPlatformCertificates
    ID          tcg-at-previousPlatformCertificates }

PreviousPlatformCertificates ::= SEQUENCE(SIZE(1..MAX) OF Trait
```

3.3.12 `cRLDistributionPoints` extension

Start of informative comment

This extension provides the location of the issuer's revocation information. This extension is optional.

End of informative comment

When a `cRLDistributionPoints` extension is present, its `critical` field SHALL be `FALSE`.

3.3.13 `holder` component

Start of informative comment

This component is applicable to only the AC encoding of the TCG Platform Certificate.

In a Platform Certificate, the **`holder`** component contains a reference to one of the RoT's certificates. When multiple keys or certificates are referenced for a particular RoT, or multiple RoTs are referenced in the Platform Certificate, the **`holder`** component is populated with a single reference and the **`cryptographicAnchors`** attribute is used for all additional references. This Platform Certificate Profile specification does not endow special meaning to the certificate referenced by the **`holder`** component over the certificates or keys referenced in the **`cryptographicAnchors`** attribute.

In a Delta Platform Certificate, the **`holder`** component mirrors the contents of the **`holder`** component of the Base Platform Certificate. If the Delta Platform Certificate references new certificates or keys issued by the Delta Platform Certificate issuer that are not present in the prior Base Platform Certificate or prior Delta Platform Certificate(s), the **`cryptographicAnchors`** attribute is used.

As a reminder to the Reader, the **`holder`** component is defined in [12] as follows:

```
Holder ::= SEQUENCE {
    baseCertificateID [0] IssuerSerial OPTIONAL,
        -- the issuer and serial number of
        -- the holder's Public Key Certificate
    entityName        [1] GeneralNames OPTIONAL,
        -- the name of the claimant or role
    objectDigestInfo  [2] ObjectDigestInfo OPTIONAL
        -- used to directly authenticate the holder,
        -- for example, an executable
}

IssuerSerial  ::=  SEQUENCE {
    issuer         GeneralNames,
    serial         CertificateSerialNumber,
    issuerUID      UniqueIdentifier OPTIONAL
}
```

NOTE: This specification does not stipulate the order in which the RoT's certificate references must appear in the Platform Certificate. The RoT's certificates can appear in any order. If the RoT has multiple certificates, the other RoT certificates are included in the **`cryptographicAnchors`** attribute.

End of informative comment

Only the **`baseCertificateID`** field of the **`holder`** component SHALL be used as specified below. Other fields in the **`holder`** component SHALL NOT be used.

In a Platform Certificate:

- The **`issuer`** field of **`baseCertificateID`** SHALL contain a Distinguished Name (DN) that matches the **`issuer`** component of the Root of Trust certificate being referenced.
- The **`serial`** field of **`baseCertificateID`** SHALL contain the serial number of the Root of Trust certificate being referenced.

In a Delta Platform Certificate:

- The **`holder`** component SHALL match the content of the **`holder`** component of the Base Platform Certificate being referenced.

3.3.14 `subject` component

Start of informative comment

This field is applicable to only the PKC encoding of the TCG Platform Certificate.

The **`subject`** component identifies the entity associated with the public key stored in the subject public key field.

End of informative comment

The **`subject`** component SHALL contain a non-empty X.500 Distinguished Name (DN) in accordance with RFC 5280 [14].

3.3.15 `cryptographicAnchors` attribute

Start of informative comment

The **`cryptographicAnchors`** attribute contains references to additional cryptographic anchors present in a platform other than the one referenced by the **`holder`** component of an AC Platform Certificate or the **`subjectPublicKeyInfo`** component of a PKC Platform Certificate.

The cryptographic anchors are referenced using **`Trait`**, which is defined in section 4.13.2.

The certificates or keys referenced in the **`cryptographicAnchors`** attribute can appear in the **`CryptographicAnchors`** sequence in any order.

End of informative comment

A **`Trait`** in a **`CryptographicAnchors`** sequence that references a certificate SHALL use the appropriate OID value in its **`traitCategory`** field as defined in Section 4.2.2 to specify the type of certificate.

A **`Trait`** in a **`CryptographicAnchors`** sequence that references a public key SHALL use **`tcg-tr-cat-PublicKey`** in its **`traitCategory`** field.

```
cryptographicAnchors ATTRIBUTE ::= {
    WITH SYNTAX CryptographicAnchors
    ID          tcg-at-cryptographicAnchors }

CryptographicAnchors ::= SEQUENCE(SIZE(1..MAX) OF Trait
```

3.3.16 `subjectAltName` extension

Start of informative comment

This extension defines the alternative name of the entity associated with this certificate, i.e., the platform. This extension is required.

The **`subjectAltName`** extension is defined in [14] as follows.

```
id-ce-subjectAltName OBJECT IDENTIFIER ::=  { id-ce 17 }

SubjectAltName ::= GeneralNames

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName                       [0]     OtherName,
    rfc822Name                      [1]     IA5String,
    dNSName                         [2]     IA5String,
    x400Address                     [3]     ORAddress,
    directoryName                   [4]     Name,
    ediPartyName                    [5]     EDIPartyName,
    uniformResourceIdentifier       [6]     IA5String,
    iPAddress                       [7]     OCTET STRING,
    registeredID                    [8]     OBJECT IDENTIFIER }
```

```
OtherName ::= SEQUENCE {
    type-id    OBJECT IDENTIFIER,
    value      [0] EXPLICIT ANY DEFINED BY type-id }
```

The **critical** field of the **subjectAltName** extension is required to be **FALSE**, otherwise any additional **Trait** that cannot be validated by a relying party would fail the certificate verification entirely.

End of informative comment

The **critical** field of the **subjectAltName** extension SHALL be **FALSE**.

The **GeneralNames** sequence SHALL contain an **OtherName** sequence, with the **type-id** field set to **tcg-at-platformIdentifier** and the **value** field containing a **PlatformIdentifier** sequence.

A **PlatformIdentifier** sequence SHALL contain a **Trait** whose **traitCategory** is **tcg-tr-cat-platformManufacturer**. This **Trait** SHALL identify the platform's manufacturer.

A **PlatformIdentifier** sequence SHALL contain a **Trait** whose **traitCategory** is **tcg-tr-cat-platformModel**. This **Trait** SHALL identify the platform's model.

A **PlatformIdentifier** sequence SHALL contain a **Trait** whose **traitCategory** is **tcg-tr-cat-platformVersion**. This **Trait** SHALL identify the platform's version.

A **PlatformIdentifier** sequence SHOULD contain a **Trait** whose **traitCategory** is **tcg-tr-cat-platformSerial**. When present, this **Trait** SHALL identify the platform's serial number value that is manufacturer-specific.

A **PlatformIdentifier** sequence MAY contain a **Trait** whose **traitCategory** is **tcg-tr-cat-platformManufacturerIdentifier**. When present, this **Trait** SHALL identify the platform's manufacturer using an IANA-assigned Private Enterprise Number [9].

```
PlatformIdentifier ::= SEQUENCE (SIZE(1..MAX)) OF Trait
```

3.3.17 tCGPlatformSpecification attribute

Start of informative comment

The **tCGPlatformSpecification** attribute identifies the platform class, version and revision of the platform-specific specification with which a platform implementation is compliant. The platform specification can refer to the PC Client PFP Specification [11], for example. Standardized platform class values are defined in section 4 Platform Class of the Registry of Reserved TPM 2.0 Handles and Localities [23].

The **TCGSpecificationVersion** sequence is defined in 3.3.6.

End of informative comment

```
tCGPlatformSpecification ATTRIBUTE ::= {
    WITH SYNTAX TCGPlatformSpecification
    ID          tcg-at-tcgPlatformSpecification }

TCGPlatformSpecification ::= SEQUENCE {
    version       TCGSpecificationVersion,
    platformClass OCTET STRING (SIZE(4)) }
```

3.3.18 tBBSecurityAssertions attribute

Start of informative comment

This attribute contains security-related assertions about the platform's RoT(s) or TBB(s). The security assertions are structured as an array of **`Trait`**, which is defined in section 3.2.

A platform profile of this Platform Certificate Profile specification can define additional Traits that can be included in the **`tBBSecurityAssertions`** attribute. A platform profile can require the inclusion of the Traits listed below.

End of informative comment

The following Traits defined in this Platform Certificate Profile specification MAY be included in the **`tBBSecurityAssertions`** attribute: `Error! Reference source not found.`, `Error! Reference source not found.`, `Error! Reference source not found.`, `Error! Reference source not found.`, `Error! Reference source not found.`, `Error! Reference source not found.`, `Error! Reference source not found.`, `Error! Reference source not found.`, and `Error! Reference source not found.`. Any other Traits defined in this specification SHALL NOT be included in the **`tBBSecurityAssertions`** attribute.

```
tBBSecurityAssertions ATTRIBUTE ::= {
    WITH SYNTAX TBBSecurityAssertions-v3
    ID          tcg-at-tbbSecurityAssertions-v3 }

TBBSecurityAssertions-v3 ::= SEQUENCE (SIZE(1..MAX)) OF Trait
```

3.3.19 `platformConfiguration` attribute

Start of informative comment

This attribute defines the lists of platform component identifiers and platform properties.

The **`platformComponents`** field contains a list of individual components that constitute the platform, encoded as a SEQUENCE of type **`ComponentIdentifier-v2`**. The elements in **`ComponentIdentifier-v2`** are structured as an array of **`Trait`**.

Individual platform components can be represented by **`ComponentIdentifier-v2`** in two ways:

1) Using an array of **`Trait`**.

2) Using a single trait of type **`ComponentIdentifierV11Trait`**.

Using an array of **`Trait`** (option 1) provides maximum flexibility in describing various attributes of a component using the trait mechanism defined in section 4 of this specification. In this case, a variable number of traits can be used to describe each component's attributes as required, with certain traits being mandatory as specified in the normative statements below.

A trait of type **`ComponentIdentifierV11Trait`** (option 2), defined in section 4.2.5, can be used in cases where it is desirable to represent component information using a data structure that mirrors the **`ComponentIdentifier`** SEQUENCE defined in version 1.1 of the TCG Platform Certificate Profile specification [32]. In this case, **`ComponentIdentifier-v2`** contains a single trait of type **`ComponentIdentifierV11Trait`**.

The **`platformProperties`** field contains characteristics of the platform that the issuer considers of interest to the consumer of the Platform Certificate. This field is encoded as a SEQUENCE of type **`Property`**. Each **`Property`** essentially contains a key-value pair, with a conditionally-present status flag.

End of informative comment

A **`ComponentIdentifier-v2`** sequence that contains a **`Trait`** of type **`componentIdentifierV11Trait`** SHALL NOT contain any other **`Trait`**.

A **ComponentIdentifier-v2** sequence that contains more than one **Trait** SHALL NOT contain a **Trait** of type **componentIdentifierV11Trait**.

For each **ComponentIdentifier-v2** sequence that does not contain a **componentIdentifierV11Trait** the following requirements apply:

1. The **ComponentIdentifier-v2** sequence SHALL contain a **Trait** whose **traitCategory** is **tcg-tr-cat-componentClass**. This **Trait** SHALL identify the type of component.
2. The **ComponentIdentifier-v2** sequence SHALL contain a **Trait** whose **traitCategory** is **tcg-tr-cat-componentManufacturer**. This **Trait** SHALL identify the component's manufacturer.
3. The **ComponentIdentifier-v2** sequence SHALL contain a **Trait** whose **traitCategory** is **tcg-tr-cat-componentModel**. This **Trait** SHALL identify the component's model.
4. The **ComponentIdentifier-v2** sequence SHOULD contain a **Trait** whose **traitCategory** is **tcg-tr-cat-componentSerial**. When present, this **Trait** SHALL identify the component's serial number.
5. The **ComponentIdentifier-v2** sequence SHOULD contain a **Trait** whose **traitCategory** is **tcg-tr-cat-componentFieldReplaceable**. When present, this **Trait** SHALL indicate whether the component is field replaceable or not.
6. The **ComponentIdentifier-v2** sequence MAY contain a **Trait** whose **traitCategory** is **tcg-tr-cat-componentRevision**. When present, this **Trait** SHALL identify the component's revision.
7. In a Delta Platform Certificate, the **ComponentIdentifier-v2** sequence SHALL contain a **Trait** whose **traitCategory** is **tcg-tr-cat-componentStatus**. This **Trait** SHALL identify whether the component was added, modified or removed from the previous Platform Certificate.
8. The Platform Certificate or Delta Platform Certificate issuer MAY include other **Trait** in the **ComponentIdentifier-v2** sequence to identify the component.

In a Delta Platform Certificate, each **Property** sequence SHALL contain the **status** field, which identifies whether the component was added, modified or removed from the base Platform Certificate.

```
platformConfiguration ATTRIBUTE ::= {
    WITH SYNTAX PlatformConfiguration-v3
    ID          tcg-at-platformConfiguration-v3 }

PlatformConfiguration-v3 ::= SEQUENCE {
    platformComponents [0] IMPLICIT SEQUENCE(SIZE(1..MAX)) OF ComponentIdentifier-v2 OPTIONAL,
    platformProperties [1] IMPLICIT SEQUENCE(SIZE(1..MAX)) OF Property OPTIONAL }

ComponentIdentifier-v2 ::= SEQUENCE(SIZE(1..MAX)) OF Trait

Property ::= SEQUENCE {
    propertyName  UTF8String (SIZE (1..STRMAX)),
    propertyValue UTF8String (SIZE (1..STRMAX)),
    status        [0] IMPLICIT AttributeStatus OPTIONAL }

AttributeStatus ::= ENUMERATED {
    added    (0),
    modified (1),
    removed  (2) }
```

3.3.20 platformConfigUri attribute

Start of informative comment

This attribute defines the location where the reference integrity measurements can be obtained by the Verifier. For example, the **URITrait** defined in 4.2.17 can be used in the **platformConfigUri** attribute.

The **platformConfigUri** attribute can be included in the Delta Platform Certificate to assert changes to the PCR values of the Base Platform Certificate.

End of informative comment

```
platformConfigUri ATTRIBUTE ::= {
    WITH SYNTAX PlatformConfigUri-v3
    ID          tcg-at-platformConfigUri-v3
}

PlatformConfigUri-v3 ::= SEQUENCE(SIZE(1..MAX)) OF Trait
```

3.3.21 `platformOwnership` attribute

Start of informative comment

This attribute identifies the owner or user of the platform, at the time of issuance of the Platform Certificate or Delta Platform Certificate.

End of informative comment

A **PlatformOwnership** sequence SHALL contain a **Trait** whose **traitCategory** is **tcg-tr-cat-platformOwnership**. This **Trait** SHALL identify the platform's owner.

```
platformOwnership ATTRIBUTE ::= {
    WITH SYNTAX PlatformOwnership
    ID          tcg-at-platformOwnership
}

PlatformOwnership ::= SEQUENCE(SIZE(1..MAX)) OF Trait
```

3.3.22 `manufacturingAssertions` attribute

Start of informative comment

This attribute provides a mechanism for platform manufacturers to make platform manufacturer-specific assertions about the manufacturing and supply chain properties of the platform. These assertions are expressed as **Trait(s)**.

End of informative comment

```
manufacturingAssertions ATTRIBUTE ::= {
    WITH SYNTAX ManufacturingAssertions
    ID          tcg-at-manufacturingAssertions
}

ManufacturingAssertions ::= SEQUENCE(SIZE(1..MAX)) OF Trait
```

3.3.23 `subjectPublicKeyInfo` component

Start of informative comment

This component is applicable to only the PKC encoding of the TCG Platform Certificate.

The **subjectPublicKeyInfo** component contains the public key associated with a Platform Certificate and identifies the algorithm with which the key is used (e.g., RSA, DSA, or Diffie-Hellman).

End of informative comment

3.3.24 `subjectKeyIdentifier` extension

Start of informative comment

This extension is applicable to only the PKC encoding of the TCG Platform Certificate.

The Subject Key Identifier extension identifies the public key present in the **subjectPublicKeyInfo** component.

End of informative comment

3.3.25 `keyUsage` extension

Start of informative comment

This extension is applicable to only the PKC encoding of the TCG Platform Certificate.

The **`keyUsage`** extension defines the intended purpose of the subject public key in the Platform Certificate.

End of informative comment

This field MUST comply with the Key Usage requirement defined in the TCG EK Credential Profile Specification [7].

This extension MUST be critical.

3.3.26 `subjectDirectoryAttributes` extension

Start of informative comment

This extension is applicable to only the PKC encoding of the TCG Platform Certificate.

The extension defines miscellaneous properties and security assertions about the platform's manufacturer.

The following Platform attributes are included:

`tCGCredentialType,` **`tCGCredentialSpecification`**, **`tCGPlatformSpecification`** and **`tBBSecurityAssertions`**.

The following attributes can be included:

`platformConfiguration` and **`platformConfigUri.`**

End of informative comment

This extension SHALL be non-critical.

The following attributes are documented for compatibility with previously published TCG or TCPA specifications but SHOULD NOT be included in Platform Certificates:

- The "TCPA Specification Version" attribute, with field values correctly reflecting the highest version of the TCG specification with which the TPM implementation conforms.
- If the TPM has been successfully evaluated against a Common Criteria protection profile, the TPM protection profile identifier attribute.
- If the TPM has been successfully evaluated against a Common Criteria security target, the TPM security target identifier attribute.
- If the RTM and the means by which the TPM and RTM have been incorporated into the platform have been successfully evaluated against a Common Criteria protection profile, the "TBB protection profile" identifier attribute.
- If the RTM and the means by which the TPM and RTM have been incorporated into the platform have been successfully evaluated against a Common Criteria security target, the "TBB security target" identifier attribute.
- Optionally, the "security qualities" attribute with a text string reflecting the security qualities of the platform.

3.3.27 `basicConstraints` extension

Start of informative comment

This extension is applicable to only the PKC encoding of the TCG Platform Certificate.

This extension indicates whether the subject is a CA.

End of informative comment

The “CA” Constraint SHALL be present and be set to FALSE.

This extension SHALL be critical.

3.3.28 `extKeyUsage` extension

Start of informative comment

This extension is applicable to only the PKC encoding of the TCG Platform Certificate.

This extension defines the intended purpose of the subject public key.

End of informative comment

`extKeyUsage` SHOULD contain the OID **`tcg-kp-PlatformKeyCertificate`** defined in section 3.3.1 of this document. The OID is used to unambiguously identify the certificate as a Platform Public Key Certificate.

This extension SHALL be non-critical.

4 Trait definition and instances

4.1 Trait sequence and TRAIT class

Start of informative comment

This Platform Certificate Profile specification defines the following constants for use with ASN.1 types defined in this section:

URIMAX is a constant used to provide an upper bound on the length of a URI included in the platform certificate. This upper bound is helpful to consumers of the extension and also helps limit the overall size of the certificate. In order to provide a reasonable upper bound for ASN.1 parsers, **URIMAX** is limited to 1024 in the context of this document. This value was selected as it matches the length limit for <A> anchors in HTML as specified by the SGML declaration (LITLEN) for HTML [5].

STRMAX is a constant defining the upper bound on the length of a string type. Like the **URIMAX,** this is to aid ASN.1 parsers and help limit the upper bound on the length of the certificate. Based on the expected sizes of ASN.1-encoded strings in this document, **STRMAX** has an upper bound of 256.

CERTSTRMAX is a constant defining the upper bound on the length of a string containing a PEM-encoded certificate. In the context of this document the recommended maximum size for **CERTSTRMAX** is 100KB.

End of informative comment

URIMAX SHALL NOT exceed a value of 1024.

STRMAX SHALL NOT exceed a value of 256.

CERTSTRMAX SHALL NOT exceed a value of 100 KB.

Start of informative comment

This section defines the **Trait** sequence and the **TRAIT** class. The **TRAIT** class allows the addition of **Trait** sequences to the attributes of a TCG Platform Certificate, without requiring modification to the overall ASN.1 definition. The definition of **Trait** and **TRAIT** is analogous to those of **Extension** and **EXTENSION** defined in [27] and [14].

The **Trait** fields are used as follows:

- **traitId** is an OID that specifies the ASN.1 syntax of the value that is in **traitValue**.
- **traitCategory** is an OID that specifies the category of the information encoded in **traitValue**. For example, **traitCategory** specifies whether **traitValue** contains the manufacturer name of a component, or the FIPS level security assertion of the platform.
- **traitRegistry** is an OID that specifies the registry that a Verifier would use to verify whether the assertion encoded in **traitValue** is valid in the current state of the platform. Each **Trait** of the set representing a component or assertion can have its own **traitRegistry**. For example, the component manufacturer name and model name can use different registries. A **traitRegistry**'s OIDs can be defined by TCG or other entities such as the platform manufacturer. The **tcg-registry-none** OID indicates that no registry is used.
- **description** is an optional string that contains information about the content of the **Trait**.
- **descriptionURI** is an optional string that contains an URI relevant to the content of the **Trait**.
- **traitValue** is a byte array whose content and encoding is specified by **traitId**, **traitCategory** and **traitRegistry**.

End of informative comment

When a **traitRegisty** field uses **tcg-registry-none**, or another value not defined by TCG, the Platform Certificate or Delta Platform Certificate issuer SHALL document in the **description** or **descriptionURI** fields where a relying party can find the definition of the associated **Trait SEQUENCE**.

```
Trait ::= SEQUENCE {
   traitId        TRAIT.&id({TraitSet}), -- Specifies the traitValue encoding
   traitCategory  OBJECT IDENTIFIER,     -- Identifies the information category contained in traitValue
   traitRegistry  OBJECT IDENTIFIER,     -- Identifies the registry used to match against the traitValue
   description    [0] IMPLICIT UTF8String (SIZE (1..STRMAX)) OPTIONAL,
   descriptionURI [1] IMPLICIT IA5String (SIZE (1..URIMAX)) OPTIONAL,
   traitValue     OCTET STRING
                  ( CONTAINING TRAIT.&TraitValueType({TraitSet}{@traitId}) ENCODED BY der)
}

TRAIT ::= CLASS {
    &id OBJECT IDENTIFIER UNIQUE,
    &TraitValueType }
WITH SYNTAX {
    SYNTAX &TraitValueType
    IDENTIFIED BY &id }

TraitSet TRAIT ::= {...}
```

Start of informative comment

While the ITU-T X.509 specification [27] uses the information object class (i.e. **CLASS**) for defining the **EXTENSION** class, the IETF RFC5280 [14][1] simplifies the **Extension** definition by omitting the information object class construction. Using a similar approach, the **Trait** definition can be simplified as follows.

```
Trait ::= SEQUENCE {
    traitId        OBJECT IDENTIFIER, -- Specifies the traitValue encoding
    traitCategory  OBJECT IDENTIFIER, -- Identifies the information category contained in traitValue
    traitRegistry  OBJECT IDENTIFIER, -- Identifies the registry used to match against the traitValue
    description    [0] IMPLICIT UTF8String (SIZE (1..STRMAX)) OPTIONAL,
    descriptionURI [1] IMPLICIT IA5String (SIZE (1..URIMAX)) OPTIONAL,
    traitValue    OCTET STRING
                  -- contains the DER encoding of an ASN.1 value corresponding to the extension type
                  -- identified by traitId
}
```

End of informative comment

4.1.1 Location category

Start of informative comment

This section defines the trait category for a trait that contains a component location inside the platform represented by the Platform Certificate (i.e., in relation to the platform assembly or enclosure). For example, in a platform with multiple memory slots, this trait could specify a particular slot for a specific memory component.

End of informative comment

A **Trait** that contains a component location SHALL have **tcg-tr-cat-componentLocation** in its **traitCategory** field.

4.1.2 Component Part Number category

Start of informative comment

[1] The extension sequence is defined in RFC5280 and it is used in both RFC5280 and RFC5755.

This section defines the trait category for a trait that contains a component part number. Components may be associated with multiple part numbers due to characteristics of their supply chain. For example, a component may have both a supplier-internal part number and a customer-facing part number. A trait with this trait category can be used to represent such part numbers.

End of informative comment

A **`Trait`** that contains a component location SHALL have **`tcg-tr-cat-componentPartNumber`** in its **`traitCategory`** field.

4.2 `Trait` instances

4.2.1 `booleanTrait`

Start of informative comment

This section defines a trait structure that contains a Boolean.

End of informative comment

```
booleanTrait TRAIT ::= {
    SYNTAX        BOOLEAN
    IDENTIFIED BY tcg-tr-ID-boolean
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **`booleanTrait`** definition can be simplified as follows.

```
name          tcg-tr-ID-boolean
OID           tcg-tr-ID 1
syntax        BOOLEAN
```

End of informative comment

4.2.2 `certificateIdentifierTrait`

Start of informative comment

This section defines a trait structure that contains a reference to another certificate.

For the informative definition of the **`IssuerSerial`** sequence see Section 3.3.13.

The **`HashedCertificateIdentifier`** sequence consists of the **`hashAlgorithm`** field and the **`hashOverSignatureValue`**. The **`hashAlgorithm`** field is of type **`AlgorithmIdentifier`** as defined in [14]. The **`hashAlgorithm`** field identifies the hashing algorithm used in **`hashOverSignatureValue`** field. The **`hashOverSignatureValue`** is calculated over the certificate's **`BIT STRING signatureValue`** for an attribute certificate or **`signature`** for a public key certificate (excluding the tag, length, and number of unused bits).

The **`AlgorithmIdentifier`** sequence is defined in [14] as follows.

```
AlgorithmIdentifier  ::=  SEQUENCE  {
    algorithm  OBJECT IDENTIFIER,
    parameters ANY DEFINED BY algorithm OPTIONAL
}
```

Since the algorithms used are all hashing algorithms, the **`parameters`** field is not used. The issuer can utilize any of the hashing algorithm OIDs found in RFC3279 [16], RFC4055 [17], SHA-3 Related Algorithms and Identifiers for PKIX [18], and GB/T 33560-2017 [19].

Additional certificate identifier traits may be defined by TCG.

End of informative comment

A **Trait** that contains a reference to a TCG Endorsement Key Certificate, compliant with [7], SHALL use **tcg-tr-cat-EKCertificate** in its **traitCategory** field.

A **Trait** that contains a reference to a TCG Initial Attestation Key Certificate, compliant with [8], SHALL use **tcg-tr-cat-IAKCertificate** in its **traitCategory** field.

A **Trait** that contains a reference to a TCG Initial Device IDentity Certificate, compliant with [8], SHALL use **tcg-tr-cat-IDevIDCertificate** in its **traitCategory** field.

A **Trait** that contains a reference to a TCG DICE Certificate, compliant with [31], SHALL use **tcg-tr-cat-DICECertificate** in its **traitCategory** field.

A **Trait** that contains a reference to a Security Protocol and Data Model Certificate, compliant with [30], SHALL use **tcg-tr-cat-SPDMCertificate** in its **traitCategory** field.

A **Trait** that contains a reference to a TCG Platform Certificate, compliant with this specification, SHALL use **tcg-tr-cat-PlatformCertificate** in its **traitCategory** field.

A **Trait** that contains a reference to a TCG Delta Platform Certificate, compliant with this specification, SHALL use **tcg-tr-cat-DeltaPlatformCertificate** in its **traitCategory** field.

A **Trait** that contains a reference to a TCG Rebase Platform Certificate, compliant with this specification, SHALL use **tcg-tr-cat-RebasePlatformCertificate** in its **traitCategory** field.

A **Trait** that contains a reference to a generic Attribute or Public Key Certificate not described by any of the categories above, compliant with [12] or [14] respectively, SHALL use **tcg-tr-cat-genericCertificate** in its **traitCategory** field.

```
certificateIdentifierTrait TRAIT ::= {
    SYNTAX        CertificateIdentifier
    IDENTIFIED BY tcg-tr-ID-certificateIdentifier
}

CertificateIdentifier ::= SEQUENCE {
    hashedCertIdentifier  [0] IMPLICIT HashedCertificateIdentifier OPTIONAL,
    genericCertIdentifier [1] IMPLICIT IssuerSerial OPTIONAL
}

HashedCertificateIdentifier ::= SEQUENCE {
    hashAlgorithm          AlgorithmIdentifier,
    hashOverSignatureValue OCTET STRING
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **certificateIdentifierTrait** definition can be simplified as follows.

```
name          tcg-tr-ID-certificateIdentifier
OID           tcg-tr-ID 2
syntax        CertificateIdentifier
```

End of informative comment

4.2.3 commonCriteriaTrait

Start of informative comment

This section defines a trait structure that contains a target for a Common Criteria evaluation. The profile and target for the evaluation can be described by either an OID, a URI to a document describing the value, or both. If both are present, they need to represent consistent values. The URI values are included in a **`URIReference,`** which describes the URI to the document and a cryptographic hash value which identifies a specific version of the document. The Common Criteria Certificate also needs to be identified, with at least its number and the name of the entity, e.g., the scheme that issued the certificate.

For the informative definition of the **`AlgorithmIdentifier`** sequence, see Section 4.2.2.

The **`GeneralizedTime`** ASN.1 type is defined in [28].

The **`CommonCriteriaMeasures`** sequence is defined in [7] as follows:

```
CommonCriteriaMeasures ::= SEQUENCE {
    Version            IA5String (SIZE (1..STRMAX)), -- “2.2” or “3.1”; future syntax defined by CC
    Assurancelevel     EvaluationAssuranceLevel,
    evaluationStatus   EvaluationStatus,
    plus               BOOLEAN DEFAULT FALSE,
    strengthOfFunction [0] IMPLICIT StrengthOfFunction OPTIONAL,
    profileOid         [1] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
    profileUri         [2] IMPLICIT URIReference OPTIONAL,
    targetOid          [3] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
    targetUri          [4] IMPLICIT URIReference OPTIONAL
}

EvaluationAssuranceLevel ::= ENUMERATED {
    level1 (1),
    level2 (2),
    level3 (3),
    level4 (4),
    level5 (5),
    level6 (6),
    level7 (7)
}

EvaluationStatus ::= ENUMERATED {
    designedToMeet       (0),
    evaluationInProgress (1),
    evaluationCompleted  (2)
}

StrengthOfFunction ::= ENUMERATED {
    basic  (0),
    medium (1),
    high   (2)
}

-- Reference to external document containing information relevant to this subject.
-- The hashAlgorithm and hashValue MUST both exist in each reference if either
-- is present.
URIReference ::= SEQUENCE {
    uniformResourceIdentifier IA5String (SIZE (1..URIMAX)),
    hashAlgorithm             AlgorithmIdentifier OPTIONAL,
    hashValue                 BIT STRING OPTIONAL
}
```

End of informative comment

A **`Trait`** that indicates a Common Criteria evaluation using the **`commonCriteriaTrait`** SHALL use **`tcg-tr-cat-commonCriteria`** in its **`traitCategory`** field.

```
commonCriteriaTrait TRAIT ::= {
    SYNTAX        CommonCriteriaEvaluation
    IDENTIFIED BY tcg-tr-ID-CommonCriteria
}

CommonCriteriaEvaluation ::= SEQUENCE {
    cCMeasures                CommonCriteriaMeasures,
    cCCertificateNumber       UTF8String (SIZE (1..STRMAX)),
    cCCertificateAuthority    UTF8String (SIZE (1..STRMAX)),
    evaluationScheme          [0] IMPLICIT UTF8String (SIZE (1..STRMAX)) OPTIONAL,
            cCCertificateIssuanceDate [1] IMPLICIT GeneralizedTime OPTIONAL,
            cCCertificateExpiryDate   [2] IMPLICIT GeneralizedTime OPTIONAL
        }
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14] , the **`commonCriteriaTrait`** definition can be simplified as follows.

```
        name          tcg-tr-ID-commonCriteria
        OID           tcg-tr-ID 3
        syntax        CommonCriteriaMeasures
```

End of informative comment

4.2.4 componentClassTrait

Start of informative comment

This section defines a trait structure that identifies the type of component. The 4-byte value in a **`componentClassTrait`** is defined by the registry identified in the **`traitRegistry`** field.

End of informative comment

```
        ComponentClassTrait TRAIT ::= {
            SYNTAX        OCTET STRING (SIZE(4))
            IDENTIFIED BY tcg-tr-ID-componentClass
        }
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **`componentClassTrait`** definition can be simplified as follows.

```
        name          tcg-tr-ID-componentClass
        OID           tcg-tr-ID 4
        syntax        OCTET STRING (SIZE(4))
```

End of informative comment

4.2.5 componentIdentifierV11Trait

Start of informative comment

This section defines a trait structure that contains a **`ComponentIdentifier`** sequence as defined in version 1.1 of this specification [32].

This trait structure can be used to report a component's information in the same sequence as with version 1.1 of this specification.

For the Component Class registries, **`tcg-registry-componentClass-tcg`** refers to the registry specified in [34], **`tcg-registry-componentClass-dmtf`** to the registry specified in [24], **`tcg-registry-componentClass-pcie`** to the registry specified in [25], and **`tcg-registry-componentClass-storage`** to the registry specified in [36]

The **`AttributeStatus`** enumeration is defined in Section 3.3.19.

The **`CertificateIdentifier`** sequence is defined in Section4.2.2.

The **`URIReference`** sequence is defined in Section4.2.3.

End of informative comment

A **Trait** that identifies a component using the **componentIdentifierV11Trait** SHALL use **tcg-tr-cat-componentIdentifierV11** in its **traitCategory** field.

A **Trait** that identifies a component using the **componentIdentifierV11Trait** SHALL populate the **componentClass**, **componentManufacturer** and **componentSerial** fields in its **ComponentIdentifierV11** sequence.When listing a MAC address in the **addressValue** field, the UTF8 characters encoding the big-endian hexadecimal presentation of the MAC address, with no delimiter, SHALL be used and uppercase letters SHALL be used.

```
componentIdentifierV11Trait TRAIT ::= {
    SYNTAX        ComponentIdentifierV11
    IDENTIFIED BY tcg-tr-ID-componentIdentifierV11
}

ComponentIdentifierV11 ::= SEQUENCE {
    componentClass           ComponentClass,
    componentManufacturer    UTF8String (SIZE (1..STRMAX)),
    componentModel           UTF8String (SIZE (1..STRMAX)),
    componentSerial          [0] IMPLICIT UTF8String (SIZE (1..STRMAX)) OPTIONAL,
    componentRevision        [1] IMPLICIT UTF8String (SIZE (1..STRMAX)) OPTIONAL,
    componentManufacturerId  [2] IMPLICIT PrivateEnterpriseNumber OPTIONAL,
    fieldReplaceable         [3] IMPLICIT BOOLEAN OPTIONAL,
    componentAddresses       [4] IMPLICIT SEQUENCE(SIZE(1.. MAX)) OF ComponentAddress OPTIONAL,
    componentPlatformCert    [5] IMPLICIT CertificateIdentifier OPTIONAL,
    componentPlatformCertUri [6] IMPLICIT URIReference OPTIONAL,
    status                   [7] IMPLICIT AttributeStatus OPTIONAL
}

ComponentClass ::= SEQUENCE {
    componentClassRegistry ComponentClassRegistry,
    componentClassValue    OCTET STRING (SIZE(4))
}

ComponentClassRegistry ::= OBJECT IDENTIFIER ( tcg-registry-componentClass-tcg | tcg-registry-componentClass-dmtf | tcg-registry-componentClass-pcie | tcg-registry-componentClass-storage )

PrivateEnterpriseNumber ::= OBJECT IDENTIFIER

ComponentAddress ::= SEQUENCE {
    addressType AddressType,
    addressValue UTF8String (SIZE (1..STRMAX))
}

AddressType ::= OBJECT IDENTIFIER ( tcg-address-ethernetmac | tcg-address-wlanmac | tcg-address-bluetoothmac )
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **componentIdentifierV11Trait** definition can be simplified as follows.

```
name         tcg-tr-ID-componentIdentifierV11
OID          tcg-tr-ID 5
syntax       ComponentIdentifierV11
```

End of informative comment

4.2.6 fIPSLevelTrait

Start of informative comment

This section defines a trait structure that contains a FIPS 140 series publication classification assertion.

End of informative comment

A **Trait** that indicates a FIPS classification level using the **fIPSLevelTrait** SHALL use **tcg-tr-cat-FIPSLevel** in its **traitCategory** field.

```
fIPSLevelTrait TRAIT ::= {
    SYNTAX        FIPSLevel
    IDENTIFIED BY tcg-tr-ID-FIPSLevel
}

FIPSLevel ::= SEQUENCE {
    version IA5String (SIZE (1..STRMAX)), -- “140-1”, “140-2”, or “140-3”
    level SecurityLevel,
    plus BOOLEAN DEFAULT FALSE
}

SecurityLevel ::= ENUMERATED {
    level1 (1),
    level2 (2),
    level3 (3),
    level4 (4)
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14] , the **fIPSLevelTrait** definition can be simplified as follows.

```
name         tcg-tr-ID-FIPSLevel
OID          tcg-tr-ID 6
syntax       FIPSLevel
```

End of informative comment

4.2.7 iSO9000Trait

Start of informative comment

This section defines a trait structure that contains an ISO9000 certification assertion.

End of informative comment

A **Trait** that contains an ISO9000 certification assertion using the **iSO9000Trait** SHALL use **tcg-tr-cat-ISO9000** in its **traitCategory** field.

```
ISO9000Trait TRAIT ::= {
    SYNTAX        ISO9000Certification
    IDENTIFIED BY tcg-tr-ID-ISO9000
}

ISO9000Certification ::= SEQUENCE {
    iso9000Certified BOOLEAN DEFAULT FALSE,
    iso9000Uri       IA5String (SIZE (1..URIMAX)) OPTIONAL
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **iSO9000Trait** definition can be simplified as follow.

```
name         tcg-tr-ID-ISO9000
OID          tcg-tr-ID 7
syntax       ISO9000Certification
```

End of informative comment

4.2.8 networkMACTrait

Start of informative comment

This section defines a trait structure that contains the MAC address of a network interface.

The **ComponentAddress** sequence is defined in 4.2.5.

End of informative comment

A **Trait** that contains a MAC address using the **networkMACTrait** SHALL use **tcg-tr-cat-networkMAC** in its **traitCategory** field.

```
NetworkMACTrait TRAIT ::= {
    SYNTAX        ComponentAddress
    IDENTIFIED BY tcg-tr-ID-networkMAC
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14] , the **networkMACTrait** definition can be simplified as follows.

```
name         tcg-tr-ID-etworkMAC
OID          tcg-tr-ID 8
syntax       ComponentAddress
```

End of informative comment

4.2.9 oIDTrait

Start of informative comment

This section defines a trait structure that contains an OID.

End of informative comment

A **Trait** that contains an OID using the **oIDTrait** SHALL use **tcg-tr-cat-OID** in its **traitCategory** field.

```
OIDTrait TRAIT ::= {
    SYNTAX        OBJECT IDENTIFIER
    IDENTIFIED BY tcg-tr-ID-OID
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **oIDTrait** definition can be simplified as follow.

```
Name         tcg-tr-ID-OID
OID          tcg-tr-ID 9
syntax       OBJECT IDENTIFIER
```

End of informative comment

4.2.10 pENTrait

Start of informative comment

This section defines a trait structure that contains an OID that is assigned to the platform manufacturer in the IANA Private Enterprise Number web site [9].

The **PrivateEnterpriseNumber** type is defined in Section 4.2.5.

End of informative comment

A **Trait** that contains an IANA Private Enterprise Number using the **pENTrait** SHALL use **tcg-tr-cat-PEN** in its **traitCategory** field.

```
PENTrait TRAIT ::= {
    SYNTAX        PrivateEnterpriseNumber
    IDENTIFIED BY tcg-tr-ID-PEN
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **pENTrait** definition can be simplified as follows.

```
        name          tcg-tr-ID-PEN
        OID           tcg-tr-ID 10
        syntax        PrivateEnterpriseNumber
```

End of informative comment

4.2.11 platformFirmwareCapabilitiesTrait

Start of informative comment

This section defines a trait structure that indicates the security capabilities provided by the firmware.

The assertions are defined as follows.

- **fwSetupAuthLocal**: The platform supports authentication by a physically present user for platform firmware setup.
- **fwSetupAuthRemote**: The platform supports authentication by a remote entity for platform firmware setup.
- **sMMProtection**: The platform supports Management Mode memory protection, for example SMM protections, provided by the chipset and supported by the platform firmware.
- **fwKernelDMAProtection**: The platform supports firmware-based protection against DMA attacks via DMA-capable peripheral devices such as PCIe, USB-C, and USB 3.0. This capability indicates that firmware supports UEFI settings to allow kernel drivers utilize firmware protections (e.g., disable bus mastering) and hardware protections (e.g., IOMMU) against DMA attacks. NOTE: as this capability requires IOMMU support, the **iOMMUSupport** hardware capability in **platformHardwareCapabilitiesTrait** will also be set if this capability is set.

End of informative comment

A **Trait** that indicates security capabilities provided by the firmware using the **platformFirmwareCapabilitiesTrait** SHALL use **tcg-tr-cat-platformFirmwareCapabilities** in its **traitCategory** field.

```
        platformFirmwareCapabilitiesTrait TRAIT ::= {
            SYNTAX          PlatformFirmwareCapabilities
            IDENTIFIED BY tcg-tr-ID-platformFirmwareCapabilities
        }

        PlatformFirmwareCapabilities ::= BIT STRING {
            fwSetupAuthLocal        (0),
            fwSetupAuthRemote       (1),
            sMMProtection           (2),
            fwKernelDMAProtection (3)
        }
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **platformFirmwareCapabilitiesTrait** definition can be simplified as follows.

```
        name          tcg-tr-ID-platformFirmwareCapabilities
        OID           tcg-tr-ID 11
        syntax        PlatformFirmwareCapabilities
```

End of informative comment

4.2.12 platformFirmwareSignatureVerificationTrait

Start of informative comment

This section defines a trait structure that indicates the method by which platform firmware signatures can be verified by the platform during boot.

The assertions are defined as follows.

- **hardwareSRTM**: An H-CRTM is present and verifies the signature of the next stage of the initial boot block (IBB).
- **secureBoot**: UEFI Secure Boot is present.

End of informative comment

A **Trait** that identifies the platform firmware signature verification mechanism using the **platformFirmwareSignatureVerificationTrait** SHALL use **tcg-tr-cat-platformFirmwareSignatureVerification** in its **traitCategory** field.

```
platformFirmwareSignatureVerificationTrait TRAIT ::= {
    SYNTAX        PlatformFirmwareSignatureVerification
    IDENTIFIED BY tcg-tr-ID-platformFirmwareSignatureVerification
}

PlatformFirmwareSignatureVerification ::= BIT STRING {
    hardwareSRTM (0),
    secureBoot   (1)
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **platformFirmwareSignatureVerificationTrait** definition can be simplified as follows.

```
name         tcg-tr-ID-platformFirmwareSignatureVerification
OID          tcg-tr-ID 12
syntax       PlatformFirmwareSignatureVerification
```

End of informative comment

4.2.13 platformFirmwareUpdateComplianceTrait

Start of informative comment

This section defines a trait structure that indicates that the type of platform firmware update mechanism employed by the vendor verifies a signature in accordance with NIST SP800-147, SP800-147B or SP800-193.

The assertions are defined as follows.

- **sp800-147**: Platform firmware update complies with SP800-147.
- **sp800-147B**: Platform firmware update complies with SP800-147B; this option only applies to Server platforms.
- **sp800-193**: Platform firmware update complies with SP800-193.

End of informative comment

A **Trait** that identifies the platform firmware update mechanism using the **platformFirmwareUpdateComplianceTrait** SHALL use **tcg-tr-cat-platformFirmwareUpdateCompliance** in its **traitCategory** field.

```
platformFirmwareUpdateComplianceTrait TRAIT ::= {
    SYNTAX        PlatformFirmwareUpdateCompliance
    IDENTIFIED BY tcg-tr-ID-platformFirmwareUpdateCompliance
}

PlatformFirmwareUpdateCompliance ::= BIT STRING {
    sp800-147  (0),
            sp800-147B (1),
            sp800-193  (2)
        }
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **`platformFirmwareUpdateComplianceTrait`** definition can be simplified as follow.

```
        name         tcg-tr-ID-platformFirmwareUpdateCompliance
        OID          tcg-tr-ID 13
        syntax       PlatformFirmwareUpdateCompliance
```

End of informative comment

4.2.14 platformHardwareCapabilitiesTrait

Start of informative comment

This section defines a trait structure that indicates the security capabilities provided by the platform motherboard or components physically attached to the motherboard.

The assertions are defined as follows.

- **`iOMMUSupport`**: The platform provides an IOMMU to protect the platform from DMA-based attacks.
- **`trustedExecutionEnvironment`**: The platform contains a Trusted Execution Environment.
- **`physicalTamperProtection`**: The platform supports a method of physical tamper protection, e.g., a chassis lock.
- **`physicalTamperDetection`**: The platform supports a method of physical tamper detection, e.g., a chassis intrusion switch.
- **`firmwareFlashWP`**: The platform supports firmware flash write protection, for example provided by the chipset or flash part.
- **`externalDMASupport`**: The platform includes external ports capable of DMA, e.g., USB-C or USB 3.0.

End of informative comment

A **`Trait`** that indicates the security capabilities provided by the platform motherboard using the **`platformHardwareCapabilitiesTrait`** SHALL use **`tcg-tr-cat-platformHardwareCapabilities`** in its **`traitCategory`** field.

```
        platformHardwareCapabilitiesTrait TRAIT ::= {
            SYNTAX        PlatformHardwareCapabilities
            IDENTIFIED BY tcg-tr-ID-platformHardwareCapabilities
        }

        PlatformHardwareCapabilities ::= BIT STRING {
            iOMMUSupport                (0),
            trustedExecutionEnvironment (1),
            physicalTamperProtection    (2),
            physicalTamperDetection     (3),
            firmwareFlashWP             (4),
            externalDMASupport          (5)
        }
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **`platformHardwareCapabilitiesTrait`** definition can be simplified as follows.

```
        name         tcg-tr-ID-platformHardwareCapabilities
        OID          tcg-tr-ID 14
        syntax       PlatformHardwareCapabilities
```

End of informative comment

4.2.15 rTMTrait

Start of informative comment

This section defines a trait structure that indicates which types of RTM capabilities are available in the platform. **rTMTrait** does not define which RTM is in use when the platform boots.

A platform profile of this specification defines the platform-specific meaning of each bit of **RTMTypes**. This specification provides only a general description of the types:

- **static**: the platform implements the RTM as part of the early platform firmware; this is also called a Static Root of Trust for Measurement (SRTM).
- **dynamic**: the platform implements the RTM after the platform firmware executed; this is also called a" Dynamic Root of Trust for Measurement (DRTM).
- **nonHost**: the platform implements the RTM outside of the CPU or SoC, such as in an independent platform controller.
- **virtual**: the platform implements a virtualized RTM, for example in a Virtual Machine.
- **hardwareStatic**: the platform implements a hardware-based RTM.
- **bMC**: the platform implements the RTM in the Baseboard Management Controller.

End of informative comment

A **Trait** that identifies the Root of Trust for Measurement using the **rTMTrait** SHALL use **tcg-tr-cat-RTM** in its **traitCategory** field.

```
RTMTrait TRAIT ::= {
    SYNTAX        RTMTypes
    IDENTIFIED BY tcg-tr-ID-RTM
}

RTMTypes ::= BIT STRING {
    static         (0),
    dynamic        (1),
    nonHost        (2),
    virtual        (3),
    hardwareStatic (4),
    bMC            (5)
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **rTMTrait** definition can be simplified as follows.

```
name         tcg-tr-ID-RTM
OID          tcg-tr-ID 15
syntax       RTMTypes
```

End of informative comment

4.2.16 statusTrait

Start of informative comment

This section defines a trait structure that indicates whether a component was added, modified, or removed from the base certificate.

The **AttributeStatus** sequence is defined in 3.3.19.

End of informative comment

```
statusTrait TRAIT ::= {
    SYNTAX        AttributeStatus
    IDENTIFIED BY tcg-tr-ID-status
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **statusTrait** definition can be simplified as follows.

```
name         tcg-tr-ID-status
OID          tcg-tr-ID 16
syntax       AttributeStatus
```

End of informative comment

4.2.17 uRITrait

Start of informative comment

This section defines a trait structure that contains a URI that references an external document containing information about the subject of this **Trait**.

The **URIReference** sequence is defined in 4.2.3.

End of informative comment

If either one of the **hashAlgorithm** or **hashValue** fields is present in the **URIReference** sequence, then both fields SHALL be populated.

```
uRITrait TRAIT ::= {
    SYNTAX        URIReference
    IDENTIFIED BY tcg-tr-ID-URI
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **uRITrait** definition can be simplified as follows.

```
name         tcg-tr-ID-URI
OID          tcg-tr-ID 17
syntax       URIReference
```

End of informative comment

4.2.18 uTF8StringTrait

Start of informative comment

This section defines a trait structure that contains a UTF8String.

End of informative comment

```
uTF8StringTrait TRAIT ::= {
    SYNTAX        UTF8String(SIZE (1..STRMAX))
    IDENTIFIED BY tcg-tr-ID-UTF8String
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **uTF8StringTrait** definition can be simplified as follows.

```
name         tcg-tr-ID-UTF8String
OID          tcg-tr-ID 18
syntax       UTF8String(SIZE (1..STRMAX))
```

End of informative comment

4.2.19 iA5StringTrait

Start of informative comment

This section defines a trait structure that contains an IA5String.

End of informative comment

```
iA5StringTrait TRAIT ::= {
    SYNTAX        IA5String(SIZE (1..STRMAX))
    IDENTIFIED BY tcg-tr-ID-IA5String
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **iA5StringTrait** definition can be simplified as follows.

```
name          tcg-tr-ID-UTF8String
OID           tcg-tr-ID 19
syntax        IA5String(SIZE (1..STRMAX))
```

End of informative comment

4.2.20 pEMCertStringTrait

Start of informative comment

This section defines a trait structure that contains a certificate in the PEM format [29] encoded as a UTF8 String.

End of informative comment

A **Trait** that contains a PEM-encoded certificate using the **pEMCertStringTrait** SHALL use **tcg-tr-cat-PEMCertificate** in its **traitCategory** field.

```
pEMCertStringTrait TRAIT ::= {
    SYNTAX        UTF8String (SIZE (1..CERTSTRMAX))
    IDENTIFIED BY tcg-tr-ID-PEMCertString
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **pEMCertStringTrait** definition can be simplified as follows.

```
name          tcg-tr-ID-PEMCertString
OID           tcg-tr-ID 20
syntax        UTF8String(SIZE (1..CERTSTRMAX))
```

End of informative comment

4.2.21 publicKeyTrait

Start of informative comment

This section defines a trait structure that contains a Public Key using a SubjectPublicKeyInfo sequence as defined in RFC 5280 [14].

End of informative comment

A **Trait** that contains a Public Key using the **publicKeyTrait** SHALL use **tcg-tr-cat-publicKey** in its **traitCategory** field.

```
publicKeyTrait TRAIT ::= {
    SYNTAX        SubjectPublicKeyInfo
```

```
        IDENTIFIED BY tcg-tr-ID-publicKey
    }
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **publicKeyTrait** definition can be simplified as follows.

```
    name         tcg-tr-ID-publicKey
    OID          tcg-tr-ID 21
    syntax       SubjectPublicKeyInfo
```

4.2.22 End of informative commententityGeoLocationTrait

Start of informative comment

This section defines a trait structure that specifies the geographic location of a supply chain entity in a variety of formats.

End of informative comment

A **Trait** that specifies a geographic location of an entity using the **entityGeoLocationTrait** SHALL use **tcg-tr-cat-entGeoLocation** in its **traitCategory** field.

```
    entityGeoLocationTrait TRAIT ::= {
        SYNTAX EntityGeoLocation
        IDENTIFIED BY tcg-tr-ID-entGeoLocation
    }

    ISO3166CountryCode ::= PrintableString (SIZE(2..3))
    ISO3166AdminSubdivisionCode ::= PrintableString (SIZE (3..6))
    OpenLocationCode ::= PrintableString

    EntityGeoLocation ::= SEQUENCE {
        countryCode ISO3166CountryCode,
        stateOrProvince [0] IMPLICIT ISO3166AdminSubdivisionCode OPTIONAL,
        localityName [1] IMPLICIT UTF8String (SIZE (1..STRMAX)) OPTIONAL,
        streetAddress [2] IMPLICIT UTF8String (SIZE (1..STRMAX)) OPTIONAL,
        locationCoords [3] IMPLICIT OpenLocationCode OPTIONAL,
        postalCode [5] IMPLICIT UTF8String (SIZE (1..STRMAX)) OPTIONAL
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **entityGeoLocationTrait** definition can be simplified as follows.

```
    name         tcg-tr-ID-entGeoLocation
    OID          tcg-tr-ID 22
    syntax       EntityGeoLocation
```

End of informative comment

4.2.23 countryOfOriginTrait

Start of informative comment

Many countries require the labeling of products to identify their country of origin. The Platform Certificate offers an automated means of digitally capturing this information with the use of the **countryOfOriginTrait**. For example, the “Made in USA” labelling rule by the US FTC requires products to be “all or virtually all” made in the United States is one such rule that could be satisfied with this trait.

While **location** is the only required field when using this trait, the **hasComponents** field can optionally indicate whether this assertion covers additional components that are part of the assembly this trait applies to. For example, a computer graphics card may contain additional sub-components that would be identified as part of the graphics card assembly, while the graphics card would be identified as part of the system assembly.

End of informative comment

A **`Trait`** that specifies a country of origin using the **`countryOfOriginTrait`** SHALL use **`tcg-tr-cat-countryOfOrigin`** in its **`traitCategory`** field.

```
countryOfOriginTrait TRAIT ::= {
    WITH SYNTAX        OriginComposition
    IDENTIFIED BY tcg-tr-ID-countryOfOrigin
}

OriginComposition SEQUENCE ::= Sequence {
     location EntityGeoLocation,
     hasComponents [0] BOOLEAN DEFAULT TRUE OPTIONAL,
}
```

Start of informative comment

Similarly to certificate extensions defined in [12] and [14], the **`countryOfOriginTrait`** definition can be simplified as follows.

```
name         tcg-tr-ID-countryOfOrigin
OID          tcg-tr-ID 23
syntax       OriginComposition
```

End of informative comment

4.3 Example of a `ComponentIdentifier` sequence

Start of informative comment

Table 7: Example of a ComponentIdentifier sequence's contents (informative).

Example name	Field name	Field type	Field content
Component class trait	**`traitId`**	OID	tcg-tr-ID-componentClass
	`traitCategory`	OID	tcg-tr-cat-componentClass
	`traitRegistry`	OID	tcg-registry-componentClass-tcg
	`traitValue`	OCTET STRING	Registry-defined 4-byte value encoded as UTF8String
Component manufacturer trait	**`traitId`**	OID	tcg-tr-ID-UTF8String
	`traitCategory`	OID	tcg-tr-cat-componentManufacturer
	`traitRegistry`	OID	tcg-tr-reg-none
	`traitValue`	OCTET STRING	OEM name encoded as UTF8String
Component model trait	**`traitId`**	OID	tcg-tr-ID-UTF8String
	`traitCategory`	OID	tcg-tr-cat-componentModel
	`traitRegistry`	OID	Trait registry OID
	`description`	UTF8String	"Contains the commercial model name of the component."
	`traitValue`	OCTET STRING	Component model name encoded as UTF8String
Component serial trait	**`traitId`**	OID	tcg-tr-ID-UTF8String
	`traitCategory`	OID	tcg-tr-cat-componentSerial
	`traitRegistry`	OID	Trait registry OID
	`traitValue`	OCTET STRING	Example component serial number encoded as UTF8String
Component field-replaceable trait	**`traitId`**	OID	tcg-tr-ID-boolean
	`traitCategory`	OID	tcg-tr-cat-componentFieldReplaceable
	`traitRegistry`	OID	tcg-tr-reg-none
	`description`	UTF8String	"Indicates if component is field-replaceable."
	`traitValue`	OCTET STRING	BOOLEAN "true" or "false" value

End of informative comment

5 X.509 ASN.1 Structures and OIDs

Start of informative comment

TCG has registered an object identifier (OID) namespace as an “international body” in the ISO registration hierarchy. This leads to shorter OIDs and gives TCG the ability to manage its own namespace. The OID namespace is inherited from TCPA specifications. These definitions are intended to be used within the context of an X.509 certificate specifically leveraging the profile described in RFC 5755 [12] or RFC 5280 [14].

```
        -- TCG specific OIDs
        tcg OBJECT IDENTIFIER ::= {
            joint-iso-itu-t(2) international-organizations(23) tcg(133) }
```

End of informative comment

Start of informative comment

This section specifies the different OIDs that are defined by this specification.

For the convenience of the Reader, the OIDs used in this specification but defined by TCG in other specifications [6, 7, 8, 15, 22, 32] are listed below.

```
        -- TCG specific OIDs
        tcg OBJECT IDENTIFIER ::= {
            joint-iso-itu-t(2) international-organizations(23) tcg(133) }

        tcg-attribute OBJECT IDENTIFIER       ::= {tcg 2}
        tcg-platformClass OBJECT IDENTIFIER   ::= {tcg 5}
        tcg-kp OBJECT IDENTIFIER              ::= {tcg 8}
        tcg-ca OBJECT IDENTIFIER              ::= {tcg 11}
        tcg-address OBJECT IDENTIFIER         ::= {tcg 17}
        tcg-registry OBJECT IDENTIFIER        ::= {tcg 18}


        -- TCG Platform Class Common OIDs
        tcg-common OBJECT IDENTIFIER          ::= {tcg-platformClass 1}

        -- TCG Common Attribute OIDs
       tcg-at-platformConfiguration OBJECT IDENTIFIER                  ::= {tcg-common 7}

        -- TCG Attribute OIDs
       tcg-at-tcgPlatformSpecification OBJECT IDENTIFIER               ::= {tcg-attribute 17}
       tcg-at-tcgCredentialSpecification OBJECT IDENTIFIER             ::= {tcg-attribute 23}
       tcg-at-tcgCredentialType OBJECT IDENTIFIER                      ::= {tcg-attribute 25}

        -- TCG Key Purposes OIDs
        tcg-kp-PlatformAttributeCertificate OBJECT IDENTIFIER          ::= {tcg-kp 2}
        tcg-kp-PlatformKeyCertificate OBJECT IDENTIFIER                ::= {tcg-kp 4}
        tcg-kp-DeltaPlatformAttributeCertificate OBJECT IDENTIFIER     ::= {tcg-kp 5}

        -- TCG Address OIDs
        tcg-address-ethernetmac OBJECT IDENTIFIER            ::= {tcg-address 1}
        tcg-address-wlanmac OBJECT IDENTIFIER                ::= {tcg-address 2}
        tcg-address-bluetoothmac OBJECT IDENTIFIER           ::= {tcg-address 3}

        -- TCG Registry OIDs
        tcg-registry-componentClass OBJECT IDENTIFIER   ::= {tcg-registry 3}
        tcg-registry-componentClass-pcie OBJECT IDENTIFIER   ::= {tcg-registry-componentClass 4}
        tcg-registry-componentClass-disk OBJECT IDENTIFIER   ::= {tcg-registry-componentClass 5}
```

End of informative comment

```
-- TCG specific OIDs
tcg-traits OBJECT IDENTIFIER ::= {tcg 19}


-- TCG Common Attribute OIDs
tcg-at-platformIdentifier OBJECT IDENTIFIER                          ::= {tcg-common 8}


-- TCG Platform Configuration OIDs
tcg-at-platformConfiguration-v3 OBJECT IDENTIFIER                    ::= {tcg-at-platformConfiguration 3}
tcg-at-platformConfigUri-v3 OBJECT IDENTIFIER                        ::= {tcg-at-platformConfiguration 4}


-- TCG Attribute OIDs
tcg-at-previousPlatformCertificates OBJECT IDENTIFIER                ::= {tcg-attribute 26}
tcg-at-tbbSecurityAssertions-v3 OBJECT IDENTIFIER                    ::= {tcg-attribute 27}
tcg-at-cryptographicAnchors OBJECT IDENTIFIER                        ::= {tcg-attribute 28}
tcg-at-platformOwnership OBJECT IDENTIFIER                           ::= {tcg-attribute 29}
tcg-at-manufacturingAssertions OBJECT IDENTIFIER                     ::= {tcg-attribute 30}



-- TCG Key Purposes OIDs
tcg-kp-DeltaPlatformKeyCertificate OBJECT IDENTIFIER                 ::= {tcg-kp 6}
tcg-kp-AdditionalPlatformAttributeCertificate OBJECT IDENTIFIER      ::= {tcg-kp 7}
tcg-kp-AdditionalPlatformKeyCertificate OBJECT IDENTIFIER            ::= {tcg-kp 8}


-- TCG Certificate Policy OIDs
tcg-cap-verifiedPlatformCertificate OBJECT IDENTIFIER                ::= {tcg-ca-policy 4}

-- TCG Registry OIDs
tcg-registry-componentClass-tcg OBJECT IDENTIFIER     ::= {tcg-registry-componentClass 1}
tcg-registry-componentClass-ietf OBJECT IDENTIFIER    ::= {tcg-registry-componentClass 2}
tcg-registry-componentClass-dmtf OBJECT IDENTIFIER    ::= {tcg-registry-componentClass 3}


-- TCG Trait OIDs
tcg-tr-ID OBJECT IDENTIFIER                    ::= {tcg-traits 1}
tcg-tr-ID-boolean OBJECT IDENTIFIER                                  ::= {tcg-tr-ID 1}
tcg-tr-ID-certificateIdentifier OBJECT IDENTIFIER                    ::= {tcg-tr-ID 2}
tcg-tr-ID-commonCriteria OBJECT IDENTIFIER                           ::= {tcg-tr-ID 3}
tcg-tr-ID-componentClass OBJECT IDENTIFIER                           ::= {tcg-tr-ID 4}
tcg-tr-ID-componentIdentifierV11 OBJECT IDENTIFIER                   ::= {tcg-tr-ID 5}
tcg-tr-ID-FIPSLevel OBJECT IDENTIFIER                                ::= {tcg-tr-ID 6}
tcg-tr-ID-ISO9000Level OBJECT IDENTIFIER                             ::= {tcg-tr-ID 7}
tcg-tr-ID-networkMAC OBJECT IDENTIFIER                               ::= {tcg-tr-ID 8}
tcg-tr-ID-OID OBJECT IDENTIFIER                                      ::= {tcg-tr-ID 9}
tcg-tr-ID-PEN OBJECT IDENTIFIER                                      ::= {tcg-tr-ID 10}
tcg-tr-ID-platformFirmwareCapabilities OBJECT IDENTIFIER             ::= {tcg-tr-ID 11}
tcg-tr-ID-platformFirmwareSignatureVerification OBJECT IDENTIFIER    ::= {tcg-tr-ID 12}
tcg-tr-ID-platformFirmwareUpdateCompliance OBJECT IDENTIFIER         ::= {tcg-tr-ID 13}
tcg-tr-ID-platformHardwareCapabilities OBJECT IDENTIFIER             ::= {tcg-tr-ID 14}
tcg-tr-ID-RTM OBJECT IDENTIFIER                                      ::= {tcg-tr-ID 15}
tcg-tr-ID-status OBJECT IDENTIFIER                                   ::= {tcg-tr-ID 16}
tcg-tr-ID-URI OBJECT IDENTIFIER                                      ::= {tcg-tr-ID 17}
tcg-tr-ID-UTF8String OBJECT IDENTIFIER                               ::= {tcg-tr-ID 18}
tcg-tr-ID-IA5String OBJECT IDENTIFIER                                ::= {tcg-tr-ID 19}
tcg-tr-ID-PEMCertString OBJECT IDENTIFIER                            ::= {tcg-tr-ID 20}
tcg-tr-ID-publicKey OBJECT IDENTIFIER                                ::= {tcg-tr-ID 21}
tcg-tr-ID-entGeoLocation OBJECT IDENTIFIER                           ::= {tcg-tr-ID 22}
tcg-tr-ID-countryOfOrigin OBJECT IDENTIFIER                          ::= {tcg-tr-ID 23}


-- TCG Trait Category OIDs
tcg-tr-category OBJECT IDENTIFIER              ::= {tcg-traits 2}
tcg-tr-cat-platformManufacturer OBJECT IDENTIFIER                    ::= {tcg-tr-category 1}
tcg-tr-cat-platformModel OBJECT IDENTIFIER                           ::= {tcg-tr-category 2}
```

```
tcg-tr-cat-platformVersion OBJECT IDENTIFIER                         ::= {tcg-tr-category 3}
tcg-tr-cat-platformSerial OBJECT IDENTIFIER                          ::= {tcg-tr-category 4}
tcg-tr-cat-platformManufacturerIdentifier OBJECT IDENTIFIER          ::= {tcg-tr-category 5}
tcg-tr-cat-platformOwnership OBJECT IDENTIFIER                       ::= {tcg-tr-category 6}
tcg-tr-cat-componentClass OBJECT IDENTIFIER                          ::= {tcg-tr-category 7}
tcg-tr-cat-componentManufacturer OBJECT IDENTIFIER                   ::= {tcg-tr-category 8}
tcg-tr-cat-componentModel OBJECT IDENTIFIER                          ::= {tcg-tr-category 9}
tcg-tr-cat-componentSerial OBJECT IDENTIFIER                         ::= {tcg-tr-category 10}
tcg-tr-cat-componentStatus OBJECT IDENTIFIER                         ::= {tcg-tr-category 11}
tcg-tr-cat-componentLocation OBJECT IDENTIFIER                       ::= {tcg-tr-category 12}
tcg-tr-cat-componentRevision OBJECT IDENTIFIER                       ::= {tcg-tr-category 13}
tcg-tr-cat-componentFieldReplaceable OBJECT IDENTIFIER               ::= {tcg-tr-category 14}
tcg-tr-cat-EKCertificate OBJECT IDENTIFIER                           ::= {tcg-tr-category 15}
tcg-tr-cat-IAKCertificate OBJECT IDENTIFIER                          ::= {tcg-tr-category 16}
tcg-tr-cat-IDevIDCertificate OBJECT IDENTIFIER                       ::= {tcg-tr-category 17}
tcg-tr-cat-DICECertificate OBJECT IDENTIFIER                         ::= {tcg-tr-category 18}
tcg-tr-cat-SPDMCertificate OBJECT IDENTIFIER                         ::= {tcg-tr-category 19}
tcg-tr-cat-PEMCertificate OBJECT IDENTIFIER                          ::= {tcg-tr-category 20}
tcg-tr-cat-PlatformCertificate OBJECT IDENTIFIER                     ::= {tcg-tr-category 21}
tcg-tr-cat-DeltaPlatformCertificate OBJECT IDENTIFIER                ::= {tcg-tr-category 22}
tcg-tr-cat-RebasePlatformCertificate OBJECT IDENTIFIER               ::= {tcg-tr-category 23}
tcg-tr-cat-genericCertificate OBJECT IDENTIFIER                      ::= {tcg-tr-category 24}
tcg-tr-cat-commonCriteria OBJECT IDENTIFIER                          ::= {tcg-tr-category 25}
tcg-tr-cat-componentIdentifierV11 OBJECT IDENTIFIER                  ::= {tcg-tr-category 26}
tcg-tr-cat-FIPSLevel OBJECT IDENTIFIER                               ::= {tcg-tr-category 27}
tcg-tr-cat-ISO9000 OBJECT IDENTIFIER                                 ::= {tcg-tr-category 28}
tcg-tr-cat-networkMAC OBJECT IDENTIFIER                              ::= {tcg-tr-category 29}
tcg-tr-cat-OID OBJECT IDENTIFIER                      ::= {tcg-tr-category 30}
tcg-tr-cat-PEN OBJECT IDENTIFIER                                     ::= {tcg-tr-category 31}
tcg-tr-cat-platformFirmwareCapabilities OBJECT IDENTIFIER            ::= {tcg-tr-category 32}
tcg-tr-cat-platformHardwareCapabilities OBJECT IDENTIFIER            ::= {tcg-tr-category 33}
tcg-tr-cat-platformFirmwareSignatureVerification OBJECT IDENTIFIER   ::= {tcg-tr-category 34}
tcg-tr-cat-platformFirmwareUpdateCompliance OBJECT IDENTIFIER        ::= {tcg-tr-category 35}
tcg-tr-cat-RTM OBJECT IDENTIFIER                                     ::= {tcg-tr-category 36}
tcg-tr-cat-publicKey OBJECT IDENTIFIER                               ::= {tcg-tr-category 37}
tcg-tr-cat-componentPartNumber OBJECT IDENTIFIER                     ::= {tcg-tr-category 38}
tcg-tr-cat-entGeoLocation OBJECT IDENTIFIER                          ::= {tcg-tr-category 39}tcg-tr-cat-
countryOfOrigin OBJECT IDENTIFIER                             ::= {tcg-tr-category 40}



-- TCG Trait Registry OIDs
tcg-tr-registry OBJECT IDENTIFIER      ::= {tcg-traits 3}
tcg-tr-reg-none OBJECT IDENTIFIER                                    ::= {tcg-tr-registry 1}
```

6 References

[1] TCG Glossary, https://trustedcomputinggroup.org/glossary

[2] TCG Infrastructure Working Group Reference Architecture for Interoperability (Part 1), Specification Version 1.0, https://trustedcomputinggroup.org/resource/infrastructure-work-group-reference-architecture-for-interoperability-specification-part-1-version-1-0/

[3] TCPA Main Specification, Version 1.1b, http://www.trustedcomputinggroup.org/tcpa-main-specification-version-1-1b/

[4] Key words for use in RFCs to Indicate Requirement Levels, RFC 2119, https://datatracker.ietf.org/doc/html/rfc2119

[5] Hypertext Markup Language – 2.0, RFC 1866, https://datatracker.ietf.org/doc/html/rfc1866

[6] TCG Credential Profiles For TPM Family 1.2 Specification Version 1.2, http://www.trustedcomputinggroup.org/infrastructure-work-group-tcg-credential-profiles-specification/

[7] TCG EK Credential Profile for TPM Family 2.0, Specification Version 2.0, http://www.trustedcomputinggroup.org/tcg-ek-credential-profile-tpm-family-2-0/

[8] TCG TPM 2.0 Keys for Device Identity and Attestation, https://trustedcomputinggroup.org/resource/tpm-2-0-keys-for-device-identity-and-attestation/

[9] IANA Private Enterprise Numbers, http://www.iana.org/assignments/enterprise-numbers/enterprise-numbers

[10] TCG Glossary, Version 1.1, https://trustedcomputinggroup.org/resource/tcg-glossary/

[11] TCG PC Client Specific Platform Firmware Profile Specification, https://trustedcomputinggroup.org/resource/pc-client-specific-platform-firmware-profile-specification/

[12] An Internet Attribute Certificate Profile for Authorization, www.ietf.org/rfc/rfc5755.txt

[13] TCG Algorithm Registry, http://www.trustedcomputinggroup.org/tcg-algorithm-registry/

[14] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, https://datatracker.ietf.org/doc/html/rfc5280

[15] TCG Platform Attribute Credential Profile Version 1.0, https://trustedcomputinggroup.org/tcg-platform-attribute-credential-profile/

[16] Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, https://datatracker.ietf.org/doc/html/rfc3279

[17] Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, https://datatracker.ietf.org/doc/html/rfc4055

[18] SHA-3 Related Algorithms and Identifiers for PKIX, https://tools.ietf.org/html/draft-turner-lamps-adding-sha3-to-pkix-00

[19] GB/T 33560-2017. Information security technology—Cryptographic application identifier criterion specification. http://www.spc.org.cn/gb168/online/GB%252FT%252033560-2017/

[20] A YANG Data Model for Hardware Management. https://datatracker.ietf.org/doc/html/rfc8348

[21] ITU-T X.520 Information Technology – Open Systems Interconnection – The Directory: Selected Attributed Types. https://www.itu.int/rec/T-REC-X.520-201610-I

[22] TCG PC Client Platform TPM Profile (PTP) Specification. https://trustedcomputinggroup.org/wp-content/uploads/TCG_PC_Client_Platform_TPM_Profile_PTP_2.0_r1.03_v22.pdf

[23] TCG Registry of Reserved TPM 2.0 Handles and Localities. https://trustedcomputinggroup.org/resource/registry/

[24] TCG SMBIOS-Based Component Class Registry. https://trustedcomputinggroup.org/resource/smbios-based-component-class-registry/

[25] TCG PCIe-based Component Class Registry. https://trustedcomputinggroup.org/resource/pcie-based-component-class-registry/

[26] PKCS #10: Certification Request Syntax Specification Version 1.7. https://datatracker.ietf.org/doc/html/rfc2986

[27] ITU-T X.509 Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks. https://www.itu.int/rec/T-REC-X.509-201610-S/

[28] ITU-T X.680 : Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation. https://www.itu.int/rec/T-REC-X.680/

[29] Textual Encodings of PKIX, PKCS, and CMS Structures. https://datatracker.ietf.org/doc/html/rfc7468

[30] Security Protocol and Data Model (SPDM) Specification, version 1.2.0 or later. https://www.dmtf.org/dsp/DSP0274

[31] TCG DICE certificate profiles. https://trustedcomputinggroup.org/resource/dice-certificate-profiles/

[32] TCG Platform Certificate Profile Version 1.1, Revision 19. https://trustedcomputinggroup.org/wp-content/uploads/IWG_Platform_Certificate_Profile_v1p1_r19_pub_fixed.pdf

[33] Errata for TCG Platform Certificate Profile Version 1.1 Revision 19, Errata Version 3.0. https://trustedcomputinggroup.org/resource/tcg-platform-certificate-profile/

[34] TCG Component Class Registry Version 1.0 Revision 14, https://trustedcomputinggroup.org/resource/tcg-component-class-registry/

[35] TCG DICE Endorsement Architecture for Devices Version 1.0 Revision 0.38, https://trustedcomputinggroup.org/wp-content/uploads/TCG-Endorsement-Architecture-for-Devices-V1-R38_pub.pdf

[36] TCG Measurement and Attestation RootS (MARS) Library Specification Version 1 Revision 14, https://trustedcomputinggroup.org/wp-content/uploads/TCG_MARS_Library_Spec_v1r14_pub.pdf

[37] TCG Storage Component Class Registry Version 1.0 Revision 22, https://trustedcomputinggroup.org/resource/storage-component-class-registry/