DICE Certificate Profiles

Version 1.0
Revision 0.01
July 23, 2020

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.

This document is copyrighted by Trusted Computing Group (TCG), and no license, express or implied, is granted herein other than as follows: You may not copy or reproduce the document or distribute it to others without written permission from TCG, except that you may freely do so for the purposes of (a) examining or implementing TCG specifications or (b) developing, testing, or promoting information technology standards and best practices, so long as you distribute the document with these disclaimers, notices, and license terms.

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

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

DISCLAIMERS, NOTICES, AND LICENSE TERMS ................................................................. 1

1 SCOPE .................................................................................................................................. 3
   1.1 Key Words ......................................................................................................................... 3
   1.2 Statement Type .................................................................................................................. 3

2 REFERENCES .......................................................................................................................... 4

3 TERMS AND DEFINITIONS .................................................................................................... 5
   3.1 Trusted Computing Terms ................................................................................................. 5
   3.2 Glossary ............................................................................................................................. 5
   3.3 DICE Architecture Terminology Conventions ................................................................. 5
   3.4 Abbreviations .................................................................................................................... 6

4 INTRODUCTION ....................................................................................................................... 7

5 DICE Architecture Conventions for X.509 Certificates ....................................................... 8
   5.1.1 Serial Number Generation ............................................................................................. 8
   5.1.2 Certificate Lifetime ........................................................................................................ 8
   5.1.3 Subject Name .................................................................................................................. 8
   5.1.4 Issuer Name .................................................................................................................... 8
   5.1.5 Policy OIDs .................................................................................................................... 8
   5.1.6 Certificate Profiles ......................................................................................................... 10
1 SCOPE

1.1 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, Key words for use in RFCs to Indicate Requirement Levels.

1.2 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 statements.

**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 requires no action).

**End of informative comment**
2 REFERENCES


3 TERMS AND DEFINITIONS

For the purposes of this document, the following terms and definitions apply.

3.1 Trusted Computing Terms

This section contains terminology commonly understood by security, cryptology, and trusted computing practitioners. The reader may be interested in the following terminology references:

- Trusted Computing Group Glossary [1],
- NIST Computer Security Resource Center Glossary [2].

3.2 Glossary

<table>
<thead>
<tr>
<th>TERM</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digest</td>
<td>The result of a cryptographic hash operation.</td>
</tr>
<tr>
<td>Device</td>
<td>A highly integrated platform containing a programmable component with other optional programmable components and peripherals.</td>
</tr>
<tr>
<td>DevID, IDevID, LDevID</td>
<td>These terms are defined by the IEEE 802.1AR [3] standard as information that an entity (a person or device) possesses that allow it to make a verifiable claim of identity, i.e., to be authenticated.</td>
</tr>
<tr>
<td>Measurement</td>
<td>A digest of code and/or configuration data. It is implementation-specific, and out of scope for this document, whether a measurement is over a region of memory, a firmware or software image, or some combination thereof.</td>
</tr>
</tbody>
</table>

3.3 DICE Architecture Terminology Conventions

This section describes conventions for use of the DICE acronym in connection with various concepts found in the architecture where the literal expansion of the DICE acronym may result in confusing or awkward syntax.

<table>
<thead>
<tr>
<th>TERM</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DICE</td>
<td>Device Identifier Composition Engine, a hardware Root of Trust (RoT)</td>
</tr>
<tr>
<td>DICE Architecture</td>
<td>This usage refers to the set of concepts that make up the trusted computing architecture with a Device Identity Composition Engine as its central feature.</td>
</tr>
<tr>
<td>DICE Engine</td>
<td>See DICE. The redundancy in terms is noted and it confers no further meaning.</td>
</tr>
<tr>
<td>DICE Layer</td>
<td>This is a shorthand term used to describe an element of a DICE Architecture.</td>
</tr>
<tr>
<td>DeviceID</td>
<td>An asymmetric key that authenticates a combination of device and firmware. This term refers specifically to the key derived from the Compound Device Identifier value that is produced by the DICE.</td>
</tr>
<tr>
<td>Layered Identity, DICE Layered Identity</td>
<td>An identity that cannot exist without a precise chain of TCB components because it is derived from a Compound Device Identifier.</td>
</tr>
</tbody>
</table>
3.4 Abbreviations
For the purposes of this specification, the following abbreviations apply.

<table>
<thead>
<tr>
<th>ABBREVIATION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Certificate Authority</td>
</tr>
<tr>
<td>CDI</td>
<td>Compound Device Identifier</td>
</tr>
<tr>
<td>DICE</td>
<td>Device Identifier Composition Engine</td>
</tr>
<tr>
<td>ECA</td>
<td>Embedded Certificate Authority</td>
</tr>
<tr>
<td>OID</td>
<td>Object Identifier</td>
</tr>
<tr>
<td>PCR</td>
<td>Platform Configuration Register</td>
</tr>
<tr>
<td>RoT</td>
<td>Root-of-Trust</td>
</tr>
<tr>
<td>TCB</td>
<td>Trusted Computing Base</td>
</tr>
<tr>
<td>TCI</td>
<td>TCB Component Identity</td>
</tr>
<tr>
<td>UDS</td>
<td>Unique Device Secret</td>
</tr>
</tbody>
</table>
4 INTRODUCTION

This document assumes the reader is familiar with the TCG specification Hardware Requirements for a Device Identifier Composition Engine [4], i.e., DICE. The DICE specification describes the composition of an identifier that represents the combination of hardware and software that begins a device boot sequence.

The DICE hardware specification describes the construction of the Compound Device Identifier (CDI). The DICE Layering Architecture specification [5] describes how this construction also applies to transitions between components of a device’s Trusted Computing Base (TCB). A device’s TCB consists of all security relevant components that have been loaded at a given point in the boot sequence. A TCB component comprises hardware, firmware, software and/or configuration. A measurement of a TCB component is a TCB Component Identifier (TCI). An example of a TCI value is a digest of component firmware. In some cases where a component does not consist of measurable firmware or software, a hardware product identifier (or equivalent) may be used.

The DICE layering architecture takes a layered approach to model multi-component systems. DICE hardware is used as the hardware Root of Trust (RoT) that anchors every layered component. The DICE hardware specification [4] describes the hardware layer combining the DICE hardware secret (called a UDS) with a measurement of the next firmware/software component, and providing a CDI secret to that next layer. Each layered TCB component performs an analogous process, combining the current TCB component’s CDI secret with a measurement (TCI) of the next TCB component, and providing the next TCB component with its own CDI.

TCB components also use their DICE Compound Device Identifier (CDI) as the input to a key generation function. For example, this may be an asymmetric key generation function producing a key pair that may be enrolled as an 802.11AR device identity credential, known as IDevID or LDevID [3]. Keys derived from CDI values may be enrolled with an application specific Certificate Authority and used to perform attestation, authentication, and certification.

This specification describes conventions and profiles for Embedded Certificate Authority (ECA) certificates and attestation certificates, including IDevID and LDevID credentials, when used in a layered architecture.
5 DICE Architecture Conventions for X.509 Certificates

The following sections describe conventions for certificates having a DICE-derived key pair. The certificate format is presumed to be X.509.v3 (see RFC5280).

5.1.1 Serial Number Generation
Certificate Serial Numbers distinguish different certificates from one another. As a result, the Serial Number field MUST be unique per CA for each Alias Key certificate. Different embedded CAs may issue certificates with the same Serial Number fields.

Start of informative comment

Be sure to understand the ASN.1 encoding for a serial number that is a non-negative integer, with particular attention to the first two bytes.

End of informative comment

5.1.2 Certificate Lifetime
Devices with a secure local clock can use the clock to set the validity period of certificates. Conventionally, devices without a secure clock use generalized time values in the distant future to ensure the validity of the certificates they create. If a secure clock is unavailable, then devices can set the notAfter portion of the certificate’s Validity Period to the X509-defined GeneralizedTime value of 99991231235959Z. This value is used to indicate that the certificate has no defined expiration date. Devices can set the notBefore portion to a known date and time in the recent past (e.g., TCB build time).

In practice, devices implementing this specification will either re-certify keys on each boot (because the device is also re-creating keypairs on each boot) or keys may be persisted if device firmware remains unchanged. In either scenario, it is assumed that expiration of certificates coincides with update of the device firmware.

5.1.3 Subject Name
Subject names SHOULD identify the component environment where the private key resides. If the component identity (e.g., TCI Ln) is a device identifier then the device vendor name may also be needed to make Subject unique. Subject may contain representations of the device serial number, firmware identifiers, or DICE layering coordinates.

The attestation Verifier SHOULD NOT obtain attestation claims from Subject or SubjectAltName fields.

When a DICE layer functions as an ECA, the root ECA Subject name MUST be device unique.

Attestation Verifiers MUST be able to perform byte array comparison of Subject names. For example, parsing and comparing sub-fields may be possible but not mandatory.

Refer to RFC5280 for rules and guidance on the interaction of the Subject and SubjectAltName fields when both are present in a certificate.

5.1.4 Issuer Name
The use of Issuer Name fields in all certificate types at layer n+1 MUST match the Subject Name of the issuing certificate at layer n. Issuer may contain representations of the device serial number, firmware identifiers, or DICE layering coordinates. Refer to RFC5280 for rules and guidance on the interaction of the Issuer and IssuerAltName fields when both are present in a certificate.

The attestation Verifier SHOULD NOT obtain attestation claims from Issuer or IssuerAltName fields.

5.1.5 Policy OIDs
The policy OID certificate extension may include the following policy OIDs. They are used by certificate issuers to identify the purpose of the TCB layer. Certificate verification processes use policy OIDs to correctly validate the usage or to improve verification processing efficiency.
The certificate profiles section (5.1.6) describes expected policy OID usage.

The following extensions use an OIDs arc from the TCG namespace.

**TCG Branch:** \( \text{tcg OBJECT IDENTIFIER ::= \{2 23 133\}} \)

**DICE Branch:** \( \text{tcg-dice OBJECT IDENTIFIER ::= \{ tcg platformClass(5) 4 \}} \)
  \( \text{tcg-dice-kp OBJECT IDENTIFIER ::= \{ tcg-dice 100 \}} \)

### 5.1.5.1 Initial Identity Policy OID

The identityInit policy OID authorizes a key to authenticate an initial identity (e.g., IDevID) for a device, DICE layer or DICE component:

\( \text{tcg-dice-kp-identityInit OBJECT IDENTIFIER ::= \{tcg-dice-kp 6\}} \)

The key MUST be bound to the device during manufacturing.

The key MUST be certified by the manufacturer. If the key is certified using an embedded CA:

- The embedded CA key MUST be certified by the manufacturer.
- The embedded CA key MUST be bound to the device during manufacturing.

No policyQualifier is defined for this OID.

### 5.1.5.2 Local Identity Policy OID

The identityLoc policy OID authorizes a key to authenticate a local identity (e.g., LDevID) for a device, DICE layer or DICE component:

\( \text{tcg-dice-kp-identityLoc OBJECT IDENTIFIER ::= \{tcg-dice-kp 7\}} \)

The key SHOULD be bound to the device post manufacturing.

The key MUST be certified by a local issuer.

If the key is certified using an embedded CA, the embedded CA MUST be certified by the local issuer.

No policyQualifier is defined for this OID.

### 5.1.5.3 Initial Attestation Policy OID

The attestInit policy OID authorizes a key to attest (sign) evidence that describes a device (e.g., PCR, SW hash, product name), DICE layer or DICE component:

\( \text{tcg-dice-kp-attestInit OBJECT IDENTIFIER ::= \{tcg-dice-kp 8\}} \)

The key MUST be bound to the device during manufacturing.

The key MUST be certified by the manufacturer. If the key is certified using an embedded CA:

- The embedded CA key MUST be certified by the manufacturer.
- The embedded CA key MUST be bound to the device during manufacturing.

No policyQualifier is defined for this OID.

### 5.1.5.4 Local Attestation Policy OID

The attestLoc policy OID authorizes a key to attest (sign) evidence that describes a device (e.g., PCR, SW hash, product name), DICE layer or DICE component:

\( \text{tcg-dice-kp-attestLoc OBJECT IDENTIFIER ::= \{tcg-dice-kp 9\}} \)
The key SHOULD be bound to the device post manufacturing.
The key MUST be certified by a local issuer.
If the key is certified using an embedded CA, the embedded CA MUST be certified by the local issuer.
No policyQualifier is defined for this OID.

5.1.5.5 Initial Assertion Policy OID
The assertInit policy OID authorizes an attestation key to assert (sign) reference measurements about the device, DICE layer or DICE component:

tcg-dice-kp-assertInit OBJECT IDENTIFIER ::= {tcg-dice-kp 10}

The key MUST be bound to the device during manufacturing.
The key MUST be certified by the manufacturer.
No policyQualifier is defined for this OID.

5.1.5.6 Local Assertion Policy OID
The assertLoc policy OID authorizes an attestation key to assert (sign) reference measurements about the device, DICE layer or DICE component:

tcg-dice-kp-assertLoc OBJECT IDENTIFIER ::= {tcg-dice-kp 11}
The key SHOULD be bound to the device post manufacturing.
The key MUST be certified by a local issuer.
No policyQualifier is defined for this OID.

5.1.5.7 Embedded Certificate Authority Policy OID
The ECA policy OID authorizes use of an embedded CA. It authorizes certificate issuance for keys residing on the current device, DICE layer or DICE component:

tcg-dice-kp-eca OBJECT IDENTIFIER ::= {tcg-dice-kp 12}
The Issuer signing key MUST be certified by the manufacturer or local issuer.
No policyQualifier is defined for this OID.

5.1.6 Certificate Profiles
DICE layers may have specialized functionality and behavior. Certificate profiles allow creation of certificates that are specialized for the expected TCB use. A TCB may combine functionality and behavior that is specific to multiple certificate profiles. The certificate profile is the union of the profiles defined below.

In certificates where the subject key was generated using a CDI value, the corresponding TCI value MUST be included in attestation evidence (e.g., as a certificate extension) if present.
5.1.6.1 Initial Device Identifier (IDevID) Certificates
Device manufacturers, OEMs or other entities in a supply chain may issue IEEE802.1AR IDevID certificates using an external CA. If so, the constraints of Table 1 SHALL apply.

<table>
<thead>
<tr>
<th>FIELD NAME</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer</td>
<td>MUST identify or chain to the device manufacturer / supply chain entity that issues the certificate. If the Issuer is an embedded CA then the ECA issuer MUST chain to the manufacturer CA.</td>
</tr>
<tr>
<td>Subject</td>
<td>MUST identify the TCB owning the IDevID private key. The Subject name may be a class identifier implying there may be other device instances sharing the same name.</td>
</tr>
<tr>
<td>Subject Public Key Info</td>
<td>Contains the public key and algorithm identifier that is protected by an immutable TCB layer or a TCB layer that SHALL be modifiable only by the Issuer (as per [4]).</td>
</tr>
<tr>
<td>Key Usage</td>
<td>If Subject is an ECA then this field MUST contain keyCertSign and MUST NOT contain cRLSign. Otherwise MUST NOT contain keyCertSign.</td>
</tr>
<tr>
<td>Extended Key Usage</td>
<td>This field may contain any appropriate values for the usage model, e.g., id-kp-clientAuth for clients.</td>
</tr>
<tr>
<td>Basic Constraints</td>
<td>If Subject is an ECA then this field MUST contain cA:TRUE and pathLengthConstraint as appropriate. Otherwise the certificate SHOULD NOT contain BasicConstraints.</td>
</tr>
<tr>
<td>Policy OIDs</td>
<td>MUST contain tcg-dice-kp-identityInit, MAY contain tcg-dice-kp-eca, tcg-dice-kp-attestInit.</td>
</tr>
<tr>
<td>Attestation Extensions</td>
<td>A future TCG specification may address attestation extensions.</td>
</tr>
</tbody>
</table>

Table 1: IDevID certificate profile

5.1.6.2 Local Device ID (LDevID) Certificates
The device owner may issue an IEEE802.1AR LDevID certificate using an external CA. If so, the constraints in Table 2 SHALL apply.

<table>
<thead>
<tr>
<th>FIELD NAME</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer</td>
<td>MUST identify or chain to the owner CA. If the Issuer is an embedded CA then the ECA issuer MUST chain to the owner CA.</td>
</tr>
<tr>
<td>Subject</td>
<td>See Section 5.1.6.1 - Subject</td>
</tr>
<tr>
<td>Subject Public Key Info</td>
<td>See Section 5.1.6.1 – Subject Public Key Info</td>
</tr>
<tr>
<td>Key Usage</td>
<td>See Section 5.1.6.1 – Key Usage</td>
</tr>
<tr>
<td>Extended Key Usage</td>
<td>See Section 5.1.6.1 – Extended Key Usage</td>
</tr>
<tr>
<td>Basic Constraints</td>
<td>See Section 5.1.6.1 – Basic Constraints</td>
</tr>
<tr>
<td>Assertions Extensions</td>
<td>See Section 5.1.6.1 – Attestation Extensions</td>
</tr>
</tbody>
</table>

Table 2: LDevID certificate profile
5.1.6.3 ECA Certificates
ECA certificates can be issued by an external CA or by an embedded CA (e.g. the Subject is an Embedded Sub-CA). If so, the constraints in Table 3 SHALL apply.

<table>
<thead>
<tr>
<th>FIELD NAME</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer</td>
<td>MUST identify the CA or embedded CA that issues the certificate. The Issuer MUST ensure that the private portion of the Subject Public Key is protected by a TCB. If Issuer is an embedded CA, then Issuer MUST identify the TCB instance that issues this certificate.</td>
</tr>
<tr>
<td>Subject</td>
<td>MUST identify the TCB containing ECA functionality.</td>
</tr>
<tr>
<td>Subject Public Key Info</td>
<td>MUST contain the current TCB Layer ECA public key and algorithm identifier.</td>
</tr>
<tr>
<td>Key Usage</td>
<td>MUST contain keyCertSign. MUST NOT contain cRLSign, may contain other KeyUsage attributes as appropriate</td>
</tr>
<tr>
<td>Basic Constraints</td>
<td>MUST contain cA:TRUE and pathLengthConstraint as appropriate</td>
</tr>
<tr>
<td>Policy OIDs</td>
<td>MUST contain tcg-dice-kp-eca, may contain tcg-dice-kp-attestInit, tcg-dice-kp-attestLoc, tcg-dice-kp-identityInit, and/or tcg-dice-kp-identityLoc</td>
</tr>
<tr>
<td>Attestation Extensions</td>
<td>See Section 5.1.6.1 – Attestation Extensions</td>
</tr>
<tr>
<td>CRLDistributionPoints Extension</td>
<td>MUST be present.</td>
</tr>
</tbody>
</table>

Table 3: ECA certificate profile

5.1.6.4 Attestation Certificates
Attestation certificates can be issued either by an ECA or an external CA. The constraints in Table 4 SHALL apply.

<table>
<thead>
<tr>
<th>FIELD NAME</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer</td>
<td>MUST contain the name of the embedded CA that issues the Subject Public Key certificate. The Issuer may be an ECA (i.e., the previous TCB layer) or an external CA. If the Issuer is an ECA, the Issuer MUST identify the TCB that issues this certificate.</td>
</tr>
<tr>
<td>Subject</td>
<td>MUST identify a TCB class or instance.</td>
</tr>
<tr>
<td>Subject Public Key Info</td>
<td>MUST contain a current TCB attestation public key and algorithm identifier.</td>
</tr>
<tr>
<td>Key Usage</td>
<td>If Subject is an ECA then this field MUST contain keyCertSign and MUST NOT contain cRLSign. Otherwise MUST NOT contain keyCertSign.</td>
</tr>
<tr>
<td>Extended Key Usage</td>
<td>May contain any appropriate values for the usage model, e.g., id-kp-clientAuth for clients.</td>
</tr>
<tr>
<td>Basic Constraints</td>
<td>If Subject is an ECA then this field MUST contain cA:TRUE and pathLengthConstraint as appropriate. Otherwise the certificate SHOULD NOT contain BasicConstraints.</td>
</tr>
<tr>
<td>Policy OIDs</td>
<td>MUST contain either tcg-dice-kp-attestInit or tcg-dice-kp-attestLoc.</td>
</tr>
<tr>
<td>Attestation Extensions</td>
<td>See Section 5.1.6.1 – Attestation Extensions</td>
</tr>
</tbody>
</table>

Table 4: Attestation Identity certificate profile
5.1.6.5 Other DICE Certificates

This specification does not preclude the use of other types of certificates or credentials. However, the certificate profiles from the previous sections should serve as a conceptual guide for working with a layered DICE architecture.