

 RSP Test Certificates Definition for IoT and Consumer

Version 3.0.1

26 January 2024

Security Classification: Non-confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Copyright © 2024 GSM Association

Disclaimer

The GSM Association (“Association”) makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Compliance Notice

The information contain herein is in full compliance with the GSM Association’s antitrust compliance policy.

This Permanent Reference Document is classified by GSMA as an Industry Specification, as such it has been developed and is maintained by GSMA in accordance with the provisions set out in GSMA AA.35 - Procedures for Industry Specifications.

Table of Contents

[1 Introduction 3](#_Toc157170834)

[1.1 Scope 3](#_Toc157170835)

[1.2 References 3](#_Toc157170836)

[2 Tool chain for generation of the keys and certificates 4](#_Toc157170837)

[2.1 OpenSSL 4](#_Toc157170838)

[2.2 Keys generation 4](#_Toc157170839)

[2.3 CI Certificate Generation 5](#_Toc157170840)

[2.4 Non-Root Certificate generation 5](#_Toc157170841)

[2.5 Certificate display 7](#_Toc157170842)

[3 Test Certificates and keys – Valid test cases 7](#_Toc157170843)

[3.1 Certificate Issuer 7](#_Toc157170844)

[3.1.1 CI Certificate: definition of data to be signed 7](#_Toc157170845)

[3.1.2 CI SubCA Certificate: definition of data to be signed 8](#_Toc157170846)

[3.1 eUICC 10](#_Toc157170847)

[3.1.1 eUICC Certificate: definition of data to be signed 10](#_Toc157170848)

[3.2 EUM 12](#_Toc157170849)

[3.2.1 EUM Certificate: definition of data to be signed 12](#_Toc157170850)

[3.2.2 EUM SubCA Certificate: definition of data to be signed 14](#_Toc157170851)

[3.3 SM-DP+ 16](#_Toc157170852)

[3.3.1 SM-DP+ SubCA 16](#_Toc157170853)

[3.3.2 SM-DP+ Certificate for Authentication 18](#_Toc157170854)

[3.3.3 SM-DP+ Certficate for Profile Biding 20](#_Toc157170855)

[3.3.4 TLS 22](#_Toc157170856)

[3.4 SM-DS 24](#_Toc157170857)

[3.4.1 SM-DS SubCA 24](#_Toc157170858)

[3.4.2 SM-DSauth 25](#_Toc157170859)

[3.4.3 TLS 28](#_Toc157170860)

[3.5 eIM 30](#_Toc157170861)

[3.5.1 eIM Certificate for Signing 30](#_Toc157170862)

[3.5.2 TLS/DTLS 31](#_Toc157170863)

[4 Test Certificates and keys – Invalid test cases 32](#_Toc157170864)

[4.1 SM-DP+ 33](#_Toc157170865)

[4.1.1 SM-DP+ Certificate for Authentication 33](#_Toc157170866)

[4.1.2 SM-DP+ Certificate for Profile Binding 34](#_Toc157170867)

[4.1.3 SM-DP+ TLS Certificate 36](#_Toc157170868)

[4.2 SM-DS 40](#_Toc157170869)

[4.2.1 SM-DS Certificate for Authentication 40](#_Toc157170870)

[4.2.2 SM-DS TLS Certificate 42](#_Toc157170871)

[Annex A RSP Certificates and Keys Files (Normative) 46](#_Toc157170872)

[Annex B Alternative to Certificate Generation 47](#_Toc157170873)

[Annex C Generation of self-signed Test CI Certificates 48](#_Toc157170874)

[Annex D Process to submit support of Test CI Certificates 50](#_Toc157170875)

[Annex E Constants/Variables 52](#_Toc157170876)

[Annex F Templates for generating certificates 54](#_Toc157170877)

[Annex G Document Management 58](#_Toc157170878)

[G.1 Document History 58](#_Toc157170879)

#  Introduction

## Scope

This document's scope is to define the Test Certificates that will be used in the tests specified in SGP.23 [1] based on SGP.22 [2].

These Test Certificates are based on NIST P-256 and/or BrainpoolP256r1 curves.

The Test Certificates MAY chain up to the GSMA CI Certificate defined in this document (see section 3.1.1), or a self-signed CI Certificate (see annex D). In any case, the Test Certificates SHALL NOT be present in any commercial RSP products in their operational lifecycle.

The certificates to be created for nominal test cases, along with the relevant key pairs, are the following:

* One Test CI Certificate (CERT.CI.SIG) per curve
* One Test CI SubCA Certificate (CERT.CISUBCA.SIG) per curve
* One EUM Certificate (CERT.EUM.SIG) per curve
* One EUM SubCA Certificate (CERT.EUMSUBCA.SIG) per curve
* For each SM-DP+, two Certificates (CERT.DPauth.SIG and CERT.DPpb.SIG) per curve
* One SM-DP+ SubCA Certificate (CERT.DPSUBCA.SIG) per curve
* Two SM-DP+ TLS Certificate (CERT.DP.TLS) per curve
* One eUICC Certificate (CERT.EUICC.SIG) per curve
* One SM-DS SubCA Certificate (CERT.DSSUBCA.SIG) per curve
* One SM-DS Certificate (CERT.DSauth.SIG) per curve
* Two SM-DS TLS Certificate (CERT.DS.TLS) per curve

Note: For Variant O Certificates in Annex A, the CERT.XX.ECDSA name remains the same for backward compatibility with previous versions.

## References

|  |  |  |
| --- | --- | --- |
| Ref | DocumentNumber | Title |
| [[1](#TF_Req)] | SGP.22 | GSMA "RSP Technical specification" (latest version in v3.x series)  |
| [2] | SGP.23 | GSMA "RSP Test Specification" (latest version in v3.x series) |
| [3] | RFC5280 | Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile |
| [4] | GSMA PRD AA.35 | Procedures for Industry Specifications |

# Tool chain for generation of the keys and certificates

This section describes the tools and the environment that have been used to generate the keys and the certificates described in this document.

## OpenSSL

OpenSSL is an open source project that also provides a general-purpose cryptography library.

Information and documentation can be found here: <https://www.openssl.org/>.

Binaries can be downloaded here: <https://wiki.openssl.org/index.php/Binaries>.

The next section assumes that the tool has been installed and correctly configured in your environment.

The OpenSSL version used to generate the certificates in this document is 1.1.0e

## Keys generation

The following command lines generate (randomly) a private key

* For NIST P-256 curve:

openssl ecparam -name prime256v1 -genkey -out <sk\_file\_name>

* For brainpoolP256r1 curve:

openssl ecparam -name brainpoolP256r1 -genkey -out <sk\_file\_name>

<sk\_file\_name> specifies the file name that will contain the generated private key (not encrypted) in the PEM form.

NOTE: The PEM form is the default format: it consists of the ASN.1 DER format base64 encoded with additional header and footer lines.

The complete description of the Openssl ecparam command can be found here: <https://www.openssl.org/docs/man1.1.0/apps/ecparam.html>

The following command line generates the related public key.

openssl ec -in <sk\_file\_name> -pubout -out <pk\_file\_name>

<sk\_file\_name> specifies the file name that contains the private key generated with the previous command line.

<pk\_file\_name> specifies the file name that will contain the generated public key in the PEM form.

The complete description of the Openssl ec command can be found here: <https://www.openssl.org/docs/man1.1.0/apps/ec.html>

## CI Certificate Generation

The following command lines generate a root certificate like for the Test CI. The first command line generates the certificate in PEM format (Base64 encoded) and the second command line converts the same certificate from PEM format into DER (i.e. binary DER) encoded format.

openssl req -config <ca\_configuration\_file> -key <ca\_sk\_file\_name> -new -x509 -days <days> -sha256 -set\_serial <serial> -extensions extend -out <cert\_pem\_file\_name>

openssl x509 -in <cert\_pem\_file\_name> -outform DER -out <cert\_der\_file\_name>

<ca\_configuration\_file> is the configuration file that contains the attributes and extensions values of the CI certificate.

<ca\_sk\_file\_name> specifies the file name that contains the CA private key in PEM format.

<serial> specifies the serial number to set in the certificate, the serial number can be decimal or hex (if preceded by 0x).

<days> specifies the number of days of validity to set in the certificate.

<cert\_pem\_file\_name> specifies the file name that will contain the certificate in PEM format.

<cert\_der\_file\_name> specifies the file name that will contain the certificate in DER format

The complete description of the Openssl req command can be found here: <https://www.openssl.org/docs/man1.1.0/apps/req.html>

The complete description of the input data file format for <ca\_configuration\_file> specifying certificate extension can be found here:

<https://www.openssl.org/docs/man1.1.0/apps/x509v3_config.html>

## Non-Root Certificate generation

The generation of a certificate starts with the generation of a Certificate Signing Request (CSR). The following command line generates this CSR.

openssl req -new -nodes -sha256 -config <input\_csr\_file\_name> -key <sk\_file\_name> -out <csr\_file\_name>

<input\_csr\_file\_name> specifies the file name that contains the input data for CSR.

<sk\_file\_name> specifies the file name that contains the private key generated with the command described in section 2.2.

<csr\_file\_name> specifies the file name that will contain the generated CSR.

The complete description of the Openssl req command can be found here: <https://www.openssl.org/docs/man1.1.0/apps/req.html>

The complete description of the input data file format for CSR can be found here: <https://www.openssl.org/docs/man1.1.0/apps/x509v3_config.html>

The following command lines generate the certificate corresponding to a CSR. The first command line generates the certificate in PEM format (Base64 encoded) and the second command line converts the same certificate from PEM format into DER (i.e. binary DER) encoded format.

openssl x509 -req -in <csr\_file\_name> -CA <ca\_cert\_file\_name> -CAkey <ca\_sk\_file\_name> -set\_serial <serial> -days <days> -extfile <cert\_ext\_file\_name> -out <cert\_pem\_file\_name>

openssl x509 -in <cert\_pem\_file\_name> -outform DER -out <cert\_der\_file\_name>

<csr\_file\_name> specifies the file name that contains the CSR generated with the previous command line.

<ca\_cert\_file\_name> specifies the file name that contains the CA Certificate in PEM format.

<ca\_sk\_file\_name> specifies the file name that contains the CA private key in PEM format related to the certificate indicated by <ca\_cert\_file\_name>.

<serial> specifies the serial number to set in the certificate, the serial number can be decimal or hex (if preceded by 0x)

<days> specifies the number of days of validity to set in the certificate.

<cert\_ext\_file\_name> specifies the file name that contains certificate extensions to set in the certificate.

<cert\_pem\_file\_name> specifies the file name that will contain the certificate in PEM format.

<cert\_der\_file\_name> specifies the file name that will contain the certificate in DER format

NOTE: As defined, the input CA certificate to generate the Non-Root Certificates SHALL be in PEM format, the following command will be used to convert from DER format to PEM format (whether the PEM format is not provided)

openssl x509 -inform der -in <cert\_der\_file\_name> -out <cert\_pem\_file\_name>

The complete description of the Openssl x509 command can be found here: <https://www.openssl.org/docs/man1.1.0/apps/x509.html>

The complete description of the file format for specifying certificate extension can be found here: <https://www.openssl.org/docs/man1.1.0/apps/x509v3_config.html>

## Certificate display

A certificate can be displayed with the following command lines.

openssl x509 -in <cert\_pem\_file\_name> -text -noout

openssl x509 -in <cert\_der\_file\_name> -inform der -text -noout

<cert\_pem\_file\_name> specifies the file name that contains the certificate in PEM format.

<cert\_der\_file\_name> specifies the file name that contains the certificate in DER format.

# Test Certificates and keys – Valid test cases

Please note that currently no CRLs are provided. It needs to be confirmed that the value contained in extension crlDistributionPoint will not lead to a problem with LPA/SM-DP+/SM-DS implementations.

## Certificate Issuer

### CI Certificate: definition of data to be signed

| Field | Value  |
| --- | --- |
| version | <automatically set> |
| serialNumber | ‘See Annex E.1 |
| signature | sha256  |
| Issuer | <Value of 'subject' field> |
| Validity | 12783 days (35 years) |
| Subject | cn = Test CIou = TESTCERTo = RSPTESTc = IT |
| Extensions |  |
| subjectKeyIdentifier  | hash |
| keyUsage  | Critical, keyCertSign, cRLSign |
| certificatePolicies  | For Variant O'2.23.146.1.2.1.0' For Variant Ov3, A, B, C'2.23.146.1.2.1.0'  |
| basicConstraints  | Critical, CA = true |
| subjectAltName  | See Annex E.2 |
| crlDistributionPoints (Only present in variant O) | [1]CRL Distribution Point Distribution Point Name: Full Name: URL=http://ci.test.example.com/CRL-A.crl[2]CRL Distribution Point Distribution Point Name: Full Name: URL=http://ci.test.example.com/CRL-B.crl |

1. : CERT.CI.SIG

Hereafter the generated CI keys and certificates as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_CI\_ SIG\_<curve>.pem | Private Key of the CI |
| PK\_CI\_SIG\_<curve>.pem | Public Key of the CI |
| CERT\_CI\_SIG\_<curve>.derCERT\_CI\_SIG\_<curve>.pem | Certificate of the CI  |

1. : CI Keys and Certificates

In order to generate the different files, next commands must be performed using the previous values and following input files:

|  |  |
| --- | --- |
| Common input files | Description |
| SK\_CI\_SIG\_<curve>.pem | Private key of the Certificate Issuer  |
| CI-csr.cnf | CSR configuration file (See Annex F) |

$ openssl req -new -nodes -<signature> -config CI-csr.cnf
 -key SK\_CI\_SIG\_<curve>.pem -out CI\_SIG\_<curve>.csr

$ openssl req -config CI-csr.cnf -key SK\_CI\_SIG\_<curve>.pem -new -x509 -days <validity> -sha256 -set\_serial <serialNumber> -extensions extend -out CERT\_CI\_SIG\_<curve>.pem

$ openssl x509 -in CERT\_CI\_SIG\_<curve>.pem -outform DER -out CERT\_CI\_<SIG>\_<curve>.der

### CI SubCA Certificate: definition of data to be signed

| Field | Value  |
| --- | --- |
| version |  <automatically set> |
| serialNumber | See Annex E.1 |
| signature | sha256 |
| Issuer | <Value of ‘subject’ of Root CI in 3.1.1> |
| Validity | Same as 3.1.1 |
| Subject | cn = Test CI SubCAou = TESTCERTo = RSPTESTc = ES |
| Extensions |  |
| subjectKeyIdentifier  | hash |
| authorityKeyIdentifier  | keyid, issuer |
| keyUsage | Critical, keyCertSign, cRLSign |
| certificatePolicies  | 2.23.146.1.2.1.0.0 |
| basicConstraints  | Critical,CA = true |
| subjectAltName  | See Annex E.2 |
| crlDistributionPoints  | [1]CRL Distribution Point Distribution Point Name: Full Name: URL=http://ci.test.example.com/CRL-A.crl[2]CRL Distribution Point Distribution Point Name: Full Name: URL=http://ci.test.example.com/CRL-B.crl |

1. : CERT.CISubCA.SIG

Hereafter the generated CI keys and certificates as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_CISubCA\_SIG\_<curve>.pem | Private Key of the CI SubCA |
| PK\_CISubCA\_SIG\_<curve>.pem | Public Key of the CISubCA |
| CERT\_CISubCA\_SIG\_<curve>.derCERT\_CISubCA\_ SIG\_<curve>.pem | Certificate of the CI SubCA |

1. : CI Keys and Certificates

**Variant B and C**

In order to generate the different files, next commands must be performed using the previous values and following input files:

|  |  |
| --- | --- |
| Common input files | Description |
| CERT\_CI\_SIG\_<curve>.pem | Certificate of the Certificate Issure |
| SK\_CI\_SIG\_<curve>.pem | Private key of the Certificate Issuer |
| SK\_CISubCA\_SIG\_<curve>.pem | Private key of the CI SubCA  |
| CISubCA-csr.cnf | CSR configuration file (See Annex F) |
| CISubCA-ext.cnf | Extension Configuration file (See Annex F) |

req -new -nodes -sha256 -config CISubCA-csr.cnf -key SK\_CISubCA\_SIG\_<curve>.pem -out CISubCA-csrgenbrp

x509 -req -in CISubCA-csrgenbrp -CA CERT\_CI\_SIG\_<curve>.pem -CAkey SK\_CI\_SIG\_BRP.pem -set\_serial <serialNumber> -days <validity> -extfile CISUBCA-ext.cnf -out CERT\_CISubCA\_SIG\_<curve>.pem

x509 -in CERT\_CISubCA\_SIG\_<curve>.pem -outform DER -out CERT\_CISubCA\_SIG\_<curve>.der

## eUICC

### eUICC Certificate: definition of data to be signed

|  |  |
| --- | --- |
| Field | Value  |
| Version | <automatically set> |
| serialNumber | See Annex E.1 |
| signature | sha256  |
| Issuer | For Variants O, Ov3 and B<Value of CERT.EUM.SIG."subject" field>For Variants A and C <Value of CERT.EUMSUBCA.SIG."subject" field> |
| Validity | 2000000 days |
| Subject | cn = Test eUICCserialNumber = for Variant O '89049032123451234512345678901235' (EID)serialNumber = for Variant Ov3, A, B, C - See Annex E.1o = Same as EUM - See 3.3.1c = ES |
| **Extension**  |  |
| authorityKeyIdentifier  | keyid, issuer |
| subjectKeyIdentifier  | hash |
| keyUsage Extension | CriticaldigitalSignature  |
| certificatePolicies  | CriticalFor Variant O: '2.23.146.1.2.1.1'For Variant Ov3, A, B, C: 2.23.146.1.2.1.0.0.0.0.0 |

1. : CERT.EUICC.SIG

NOTE: OpenSSL tool does not allow the generation of Infinite duration certificates. For this reason, the eUICC certificate generated herein, only intended for test purposes, is not aligned with the SGP.14 specification. An eUICC certificate generated with another tool supporting this capability SHALL have the duration set to Infinite.

Here are the generated eUICC keys and certificates as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_EUICC\_ SIG\_<curve>.pem | Private key of the eUICC for creating signatures |
| PK\_EUICC\_ SIG\_<curve>.pem | Public Key of the eUICC |
| CERT\_EUICC\_<variant>\_ SIG\_<curve>.der | Certificate of the eUICC  |

1. : eUICC Keys and Certificates

In order to generate the different files, next commands must be performed using the previous values and following input files:

**Variant A and C**

|  |  |
| --- | --- |
| Common input files | Description |
| CERT\_EUMSUBCA\_<variant>\_SIG\_<curve>.pem | Certificate of the Certificate Issuer |
| SK\_EUMSUBCA\_SIG\_<curve>.pem | Private key of the Certificate Issuer |
| SK\_EUICC\_SIG\_<curve>.pem | Private key of the CI SubCA  |
| EUICC-csr.cnf | CSR configuration file (See Annex F) |
| EUICC-ext.cnf | Extension Configuration file (See Annex F) |

req -new -nodes -sha256 -config EUICC-csr.cnf -key SK\_EUICC\_SIG\_<curve>.pem -out EUICC-csrgenbrp

x509 -req -in EUICC-csrgenbrp -CA CERT\_EUMSUBCA\_<variant>\_SIG\_<curve>.pem -CAkey SK\_EUMSUBCA\_SIG\_<curve>.pem -set\_serial <serialNumber> -days <validity> -extfile EUICC-ext.cnf -out CERT\_EUICC\_<variant>\_SIG\_<curve>.pem

x509 -in CERT\_EUICC\_<variant>\_SIG\_<curve>.pem -outform DER -out CERT\_EUICC\_<variant>\_SIG\_<curve>.der

**Variant O and B**

|  |  |
| --- | --- |
| Common input files | Description |
| CERT\_EUM\_<variant>\_SIG\_<curve>.pem | Certificate of the Certificate Issuer |
| SK\_EUM\_SIG\_<curve>.pem | Private key of the Certificate Issuer |
| SK\_EUICC\_SIG\_<curve>.pem | Private key of the CI SubCA  |
| EUICC-csr.cnf | CSR configuration file (See Annex F) |
| EUICC-ext.cnf | Extension Configuration file (See Annex F) |

req -new -nodes -sha256 -config EUICC-csr.cnf -key SK\_EUICC\_SIG\_<curve>.pem -out EUICC-csrgenbrp

x509 -req -in EUICC-csrgenbrp -CA CERT\_EUM\_<variant>\_SIG\_<curve>.pem -CAkey SK\_EUM\_SIG\_<curve>.pem -set\_serial <serialNumber> -days <validity> -extfile EUICC-ext.cnf -out CERT\_EUICC\_<variant>\_SIG\_<curve>.pem

x509 -in CERT\_EUICC\_<variant>\_SIG\_<curve>.pem -outform DER -out CERT\_EUICC\_<variant>\_SIG\_<curve>.der

## EUM

### EUM Certificate: definition of data to be signed

| Field | Value  |
| --- | --- |
| version | <automaticallyset> |
| serialNumber | See Annex E.1 |
| signature | sha256 |
| Issuer | For Variants O, Ov3 and A <Value of CERT.CI.SIG."subject" field>For Variants B and C <Value of CERT.CISUBCA.SIG."subject" field> |
| validity | 12410 days (34 years) |
| subject | cn = EUM Test o = RSP Test EUMc = ES |
| Extensions |  |
| authorityKeyIdentifier  | issuer, keyid |
| subjectKeyIdentifier  | hash |
| keyUsage  | critical, keyCertSign |
| certificate Policies | Critical For O: 2.23.146.1.2.1.2For Ov3, A, B, C: 2.23.146.1.2.1.0.0.0 |
| subjectAltName  | See Annex E.2 |
| basicConstraints | For Variants O, Ov3 and BCriticalCA = truepathLenConstraint = 0For Variants A and CCriticalCA = truepathLenConstraint = 1 |
| crlDistributionPoints  | [1]CRL Distribution Point Distribution Point Name: Full Name: URL=http://ci.test.example.com/CRL-B.crl |
| nameConstraints | CriticalFor O:permittedSubtrees: id-at-organizationName: '2.5.4.10' organization name: ''RSP Test EUM'' UTF8String  id-at-serialNumber: '2.5.4.5' iin: ''89049032" PrintableStringFor Ov3, A, B, CpermittedSubtrees: id-at-organizationName: '2.5.4.10' organization name: ''RSP Test EUM'' UTF8String  |
| permittedEIN | For Variants Ov3, A, B, CpermitedEins: See Annex E.3 |

1. : CERT.EUM.SIG

Hereafter the generated EUM keys and certificates as defined in Annex A.

| File name | Description |
| --- | --- |
| SK\_EUM\_SIG\_<curve>.pem | Private key of the EUM for creating signatures |
| PK\_EUM\_SIG\_<curve>.pem | Public Key of the EUM |
| CERT\_EUM\_<variant>\_ SIG\_<curve>.der | Certificate of the EUM  |

1. : EUM Keys and Certificates

In order to generate the different files, next commands must be performed using the previous values and following input files:

**Variant O and A**

|  |  |
| --- | --- |
| Common input files | Description |
| CERT\_CI\_SIG\_<curve>.pem | Certificate of the Certificate Issuer |
| SK\_CI\_SIG\_<curve>.pem | Private key of the Certificate Issuer |
| SK\_EUM\_SIG\_<curve>.pem | Private key of the EUM  |
| EUM-csr.cnf | CSR configuration file (See Annex F) |
| EUM-ext.cnf | Extension Configuration file (See Annex F) |

req -new -nodes -sha256 -config EUM-csr.cnf -key SK\_EUM\_SIG\_<curve>.pem -out EUM-csrgenbrp

x509 -req -in EUM-csrgenbrp -CA CERT\_CI\_SIG\_<curve>.pem -CAkey SK\_CI\_SIG\_<curve>.pem -set\_serial <serialNumber> -days <Validity> -extfile EUM-ext.cnf -out CERT\_EUM\_<variant>\_SIG\_<curve>.pem

x509 -in CERT\_EUM\_<variant>\_SIG\_<curve>.pem -outform DER -out CERT\_EUM\_<variant>\_SIG\_<curve>.der

**Variant B and C**

|  |  |
| --- | --- |
| Common input files | Description |
| CERT\_CISubCA\_SIG\_<curve>.pem | Certificate of the Certificate Issuer |
| SK\_CISubCA\_SIG\_<curve>.pem | Private key of the Certificate Issuer |
| SK\_EUM\_SIG\_<curve>.pem | Private key of the EUM  |
| EUM-csr.cnf | CSR configuration file (See Annex F) |
| EUM-ext.cnf | Extension Configuration file (See Annex F) |

req -new -nodes -sha256 -config EUM-csr.cnf -key SK\_EUM\_SIG\_<curve>.pem -out EUM-csrgenbrp

x509 -req -in EUM-csrgenbrp -CA CERT\_CISubCA\_SIG\_<curve>.pem -CAkey SK\_CISubCA\_SIG\_<curve>.pem -set\_serial <serialNumber> -days <Validity> -extfile EUM-ext.cnf -out CERT\_EUM\_<variant>\_SIG\_<curve>.pem

x509 -in CERT\_EUM\_<variant>\_SIG\_<curve>.pem -outform DER -out CERT\_EUM\_<variant>\_SIG\_<curve>.der

### EUM SubCA Certificate: definition of data to be signed

| Field | Value  |
| --- | --- |
| version | <automatically set> |
| serialNumber | See Annex E.1 |
| signature | sha256 |
| Issuer | <Value of CERT.EUM.SIG."subject" field> |
| validity | 12410 days (34 years) |
| subject | Same as 3.3.1 |
| Extensions |  |
| authorityKeyIdentifier  | keyid, issuer |
| subjectKeyIdentifier  | hash |
| keyUsage  | Same as 3.3.1 |
| Certificate Policies | 2.23.146.1.2.1.0.0.0.0 |
| subjectAltName | See Annex E.2 |
| basicConstraints | CriticalCA = truepathLenConstraint = 0 |
| crlDistributionPoints  | Same as 3.3.1 |

1. : CERT.EUMSubCA.SIG

Hereafter the generated EUM keys and certificates as defined in Annex A.

| File name | Description |
| --- | --- |
| SK\_EUMSubCA\_SIG\_<curve>.pem | Private key of the EUM SubCA for creating signatures |
| PK\_EUMSubCA\_ SIG\_<curve>.pem | Public Key of the EUM SubCA |
| CERT\_EUMSubCA\_<variant>\_ SIG\_<curve>.der | Certificate of the EUM SUBCA  |

1. : EUM SubCA Keys and Certificates

**Variant B and C**

|  |  |
| --- | --- |
| Common input files | Description |
| CERT\_EUM\_SIG\_<curve>.pem | Certificate of the Certificate Issuer |
| SK\_EUM\_SIG\_<curve>.pem | Private key of the Certificate Issuer |
| SK\_EUMSubCA\_SIG\_<curve>.pem | Private key of the EUM  |
| EUMSubCA-csr.cnf | CSR configuration file (See Annex F) |
| EUMSubCA-ext.cnf | Extension Configuration file (See Annex F) |

req -new -nodes -sha256 -config EUMSUBCA-csr.cnf -key SK\_EUMSubBCA\_SIG\_<curve>.pem -out EUMSubCA-csrgenbrp

x509 -req -in EUMSubCA-csrgenbrp -CA CERT\_EUM\_<variant>\_SIG\_<curve>.pem -CAkey SK\_EUM\_SIG\_<curve>.pem -set\_serial <serialNumber> -days <Validity> -extfile EUMSubCA-ext.cnf -out CERT\_EUMSubCA\_<variant>\_SIG\_<curve>.pem

x509 -in CERT\_EUMSubCA\_<variant>\_SIG\_<curve>.pem -outform DER -out CERT\_EUMSubCA\_<variant>\_SIG\_<curve>.der

## SM-DP+

### SM-DP+ SubCA

Note: CERT.SM\_DPSubCA.SIG is only defined for Variant A and Variant C

| Field | Value  |
| --- | --- |
| version | *<Value automatically set>* |
| serialNumber | See Annex E.1 |
| signature | sha256 |
| Issuer | For Variant A:*<Value of CERT.CI.SIG subject field>*For Variant C:*<Value of CERT.CISubCA.SIG subject field>* |
| Validity | 1095 days (3 years) |
| Subject | cn = Test CI SM\_DPSubCAou = TESTCERTo = RSPTESTc = ES |
| Extensions |  |
| subjectKeyIdentifier | hash |
| authorityKeyIdentifier | keyid, issuer |
| keyUsage | critical, keyCertSign, cRLSign |
| certificatePolicies  | critical, 2.23.146.1.2.1.0.0.1 |
| basicConstraints | critical, cA:truepathLenConstraint = 0 |
| subjectAltName  | See Annex E.2 |
| crlDistributionPoints | For Variant A: <Value of CERT.CI.SIG "crlDistributionPoints" field>For Variant C: <Value of CERT.CISubCA.SIG "crlDistributionPoints" field> |

1. : CERT.SM\_DPSubCA.SIG data

Hereafter the generated SM-DP+ SubCA keys and certificates as defined in Annex A.

| File name | Description |
| --- | --- |
| SK\_SM\_DPSubCA\_SIG\_<curve>.pem | Private key of the SM-DP+ SubCA  |
| PK\_SM\_DPSubCA\_SIG\_<curve>.pem | Public Key of the SM-DP+ SubCA |
| CERT\_SM\_DPSubCA\_<variant>\_SIG\_<curve>.der | Certificate of the SM-DP+ SubCA  |

1. : EUM SubCA Keys and Certificates

In order to generate the different files, next commands must be performed using the previous values and following input files

|  |  |
| --- | --- |
| Common input files | Description |
| SK\_SM\_DPSubCA\_<curve>.pem | Private key of the Certificate Issuer SubCA |
| SubCA\_csr.cnf | CSR configuration file (See Annex F) |
| SubCA\_ext.cnf | Certificate configuration file (See Annex F) |

$ openssl req -new -nodes -<signature> -config DPSUBCA-csr.cnf
 -key SK\_S\_SM\_DPSubCA\_SIG\_<curve>.pem -out DP\_SubCA\_SIG\_<curve>.csr
 **Variant A**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_CI\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_CI\_SIG\_<curve>.pem | Private key of the Certificate Issuer |

openssl req -new -nodes -sha256 -config SM\_DPSubCA-csr.cnf -key SK\_SM\_DPSubCA\_SIG\_<curve>.pem -out SM\_DPSubCA-csrgennist

openssl x509 -req -in SM\_DPSubCA-csrgennist -CA CERT\_CI\_SIG\_<curve>.pem -CAkey SK\_CI\_SIG\_<curve>.pem -set\_serial <serialNumber> -days <validity> -extfile SM\_DPSubCA-ext.cnf -out CERT\_SM\_DPSubCA\_<variant>\_SIG\_<curve>.pem

openssl x509 -in CERT\_SM\_DPSubCA\_<variant>\_SIG\_<curve>.pem -outform DER -out CERT\_SM\_DPSubCA\_<variant>\_SIG\_<curve>.der

**Variant C**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_CISubCA\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_CISubCA\_SIG\_<curve>.pem | Private key of the Certificate Issuer |

openssl req -new -nodes -sha256 -config SM\_DPSubCA-csr.cnf -key SK\_SM\_DPSubCA\_SIG\_<curve>.pem -out SM\_DPSubCA-csrgennist

openssl x509 -req -in SM\_DPSubCA-csrgennist -CA CERT\_CISubCA\_SIG\_<curve>.pem -CAkey SK\_CISubCA\_SIG\_<curve>.pem -set\_serial <serialNumber> -days <validity> -extfile SM\_DPSubCA-ext.cnf -out CERT\_SM\_DPSubCA\_VARC\_SIG\_NIST.pem

openssl x509 -in CERT\_SM\_DPSubCA\_<variant>\_SIG\_<curve>.pem -outform DER -out CERT\_SM\_DPSubCA\_<variant>\_SIG\_<curve>.der

### SM-DP+ Certificate for Authentication

#### SM-DP+Certificate for Authentication: definition of data to be signed

| Field | Value  |
| --- | --- |
| Version | *<Value automatically set>* |
| serialNumber | See Annex E.1 |
| signature |  sha256 |
| Issuer | For Variant O: <Value of CERT.CI.SIG "subject" field>For Variant B:<Value ofCERT.CISubCA.SIG “subject” fieldFor Variant A and C: <Value ofCERT.SM\_DPSubCA.SIG “subject” field |
| Validity | 1095 days (3 years) |
| Subject | o = ACME cn = TEST SM-DP+ <SM-DP+ number>  |
| Extensions  |  |
| authorityKeyIdentifier | keyid, issuer |
| subjectKeyIdentifier | hash |
| keyUsage | Critical, digitalSignature  |
| certificatePolicies | For Variant O: critical, 2.23.146.1.2.1.4For Variant A, B and C: critical, 2.23.146.1.2.1.0.0.1.1 |
| subjectAltName | See Annex E.2 |
| crlDistributionPoints | For Variant O: <Value of CERT.CI.SIG "crlDistributionPoints" field>For Variant B:<Value of CERT.CISubCA.SIG "crlDistributionPoints" field>For Variant A and C:<Value of CERT.SM\_DPSubCA.SIG "crlDistributionPoints" field> |

1. : CERT.SM\_DPauth.SIG data

Hereafter the generated keys and certificates of SM-DP+ for Authentication as defined in Annex A.

| File name | Description |
| --- | --- |
| SK\_S\_SM\_DP<Number>auth\_<variant>\_ SIG\_<curve>.pem | Private Key of the SM-DP+ for creating signatures for SM-DP+ authentication |
| PK\_S\_SM\_DP<Number>auth\_<variant>\_SIG\_<curve>.pem | Public Key of the SM-DP+  |
| CERT\_S\_SM\_DP<Number>auth\_<variant>\_SIG\_<curve>.der | Certificate of the SM-DP+  |

1. : SM-DPAuth Keys and Certificates of SM-DP+

In order to generate the different files, next commands must be performed using the previous values and following input files

|  |  |
| --- | --- |
| Common input files | Description |
| SK\_S\_SM\_DP<number>auth\_SIG\_<curve>.pem | Private key of the SM-DP+ for authentication |
| SM\_DP<number>auth-csr.cnf | CSR configuration file (See Annex F) |
| SM\_DP<number>auth-ext.cnf | Certificate configuration file (See Annex F) |

$ openssl req -new -nodes -<signature> -config SM\_DP<number>auth-csr.cnf
 -key SK\_S\_SM\_DP<number>auth\_SIG\_<curve>.pem -out SM\_DPauth\_SIG\_<curve>.csr

**Variant O**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_CI\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_CI\_SIG\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DPauth\_SIG\_<curve>.csr
 -CA CERT\_CI\_SIG\_<curve>.pem -Cakey SK\_CI\_SIG\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DP<number>auth-ext.cnf
 -out CERT\_S\_SM\_DP<number>auth\_SIG\_<curve>.pem

**Variant B**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_CISubCA\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_SM\_DPSubCA\_<variant>\_SIG\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DPauth\_SIG\_<curve>.csr
 -CA CERT\_CISubCA\_SIG\_<curve>.pem -Cakey SK\_CISubCA\_SIG\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DP<number>auth-ext.cnf
 -out CERT\_S\_SM\_DP<number>auth\_<variant>\_SIG\_<curve>.pem

**Variant A and C**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_SM\_DPSubCA\_variant>\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_SM\_DPSubCA\_<variant>\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DPauth\_SIG\_<curve>.csr
 -CA CERT\_SM\_DPSubCA\_<variant>\_SIG\_<curve>.pem -Cakey SK\_SM\_DPSubCA\_<variant>\_SIG\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DP<number>auth-ext.cnf
 -out CERT\_S\_SM\_DP<number>auth\_<variant>\_SIG\_<curve>.pem

### SM-DP+ Certficate for Profile Biding

#### SM-DP+Certificate for Profile Binding: definition of data to be signed

| Field | Value  |
| --- | --- |
| Version | *<Value automatically set>* |
| serialNumber | See Annex E.1 |
| Signature | sha256 |
| Issuer | For Variant O: <Value of CERT.CI.SIG "subject" field>For Variant B: <Value of CERT.CISubCA.SIG “subject” field For Variant A and C: <Value of CERT.SM\_DPSubCA.SIG “subject” field |
| Validity | 1095 days (3 years) |
| Subject | o = ACME cn = TEST SM-DP+ <SM-DP+ number> |
| Extensions  |  |
| authorityKeyIdentifier | keyid, issuer |
| subjectKeyIdentifier | hash |
| keyUsage | critical, digitalSignature |
| certificatePolicies | For Variant O: critical, 2.23.146.1.2.1.5For Variant A, B and C: critical, 2.23.146.1.2.1.0.0.1.2 |
| subjectAltName | See Annex E.2 |
| crlDistributionPoints | For Variant O: <Value of CERT.CI.SIG "crlDistributionPoints" field>For Variant B:<Value of CERT.CISubCA.SIG "crlDistributionPoints" field>For Variant A and C:<Value of CERT.SM\_DPSubCA.SIG "crlDistributionPoints" field> |

1. : CERT.SM\_DPpb.SIG data

Hereafter the generated keys and certificates of the SM-DP+ n°1 for Profile Package Binding as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DP<Number>pb\_SIG\_<curve>.pem | Private Key of the SM-DP+  |
| PK\_S\_SM\_DP<Number>pb\_SIG\_<curve>.pem | Public Key of the SM-DP+  |
| CERT\_S\_SM\_DP<Number>pb\_<variant>\_SIG\_<curve>.der | Certificate of the SM-DP+  |

1. : SM-DPpb Keys and Certificates of SM-DP+

In order to generate the different files, next commands must be performed using the previous values and following input files

|  |  |
| --- | --- |
| Common input files | Description |
| SK\_S\_SM\_DP<number>pb\_SIG\_<curve>.pem | Private key of the SM-DP+ for Profile Binding |
| SM\_DP<number>pb-csr.cnf | CSR configuration file (See Annex F) |
| SM\_DP<number>pb-ext.cnf | Certificate configuration file (See Annex F) |

$ openssl req -new -nodes -<signature> -config SM\_DP<number>pb-csr.cnf
 -key SK\_S\_SM\_DP<number>pb\_SIG\_<curve>.pem -out SM\_DPpb\_SIG\_<curve>.csr

**Variant O**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_CI\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_CI\_SIG\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DPpb\_SIG\_<curve>.csr
 -CA CERT\_CI\_SIG\_<curve>.pem -Cakey SK\_CI\_SIG\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DP<number>pb-ext.cnf
 -out CERT\_S\_SM\_DP<number>pb\_SIG\_<curve>.pem

**Variant B**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_CISubCA\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_CISubCA\_SIG\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DPpb\_SIG\_<curve>.csr
 -CA CERT\_CISubCA\_SIG\_<curve>.pem -Cakey SK\_CISubCA\_SIG\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DP<number>pb-ext.cnf
 -out CERT\_S\_SM\_DP<number>pb\_<variant>\_SIG\_<curve>.pem

**Variant A and C**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_DPSubCA\_<variant>\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_DPSubCA\_<variant>\_SIG\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DPpb\_SIG\_<curve>.csr
 -CA CERT\_SM\_DPSubCA\_<variant>\_SIG\_<curve>.pem -Cakey SK\_SM\_DPSubCA\_SIG\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DP<number>pb-ext.cnf
 -out CERT\_S\_SM\_DP<number>pb\_<variant>\_SIG\_<curve>.pem

### TLS

#### SM-DP+ n°1 TLS Certificate: definition of data to be signed

| Field | Value |
| --- | --- |
| Version | <automatically set> |
| serialNumber | See Annex E.1 |
| signature | sha256 |
| Issuer | For Variant O: <Value of CERT.CI.SIG."subject" field>For Variant A and C: <Value of CERT.DPSubCA.SIG."subject" field>For Variant B:<Value of CERT.CISubCA.SIG."subject" field> |
| validity | 398 days  |
| subject | o = 'ACME'cn = 'testsmdpplus1.example.com' |
| Extensions |  |
| authorityKeyIdentifier | keyId, issuer |
| subjectKeyIdentifier | hash |
| keyUsage | CriticaldigitalSignature |
| certificatePolicies | For Variant O: '2.23.146.1.2.1.3' For Variant A, B, C: critical, 2.23.146.1.2.1.0.0.1.0 |
| extendedKeyUsage | critical, serverAuth, clientAuth |
| subjectAltName | See Annex E.2 |
| crlDistributionPoints | <Value of CERT.CI.SIG."crlDistributionPoints" field> |

1. : CERT.DP.TLS for SM-DP+

Hereafter the generated keys and certificates of the SM-DP+ n°1 for Profile Package Binding as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DP<Number>\_TLS\_<curve>.pem | Private Key of the SM-DP+ for TLS |
| PK\_S\_SM\_DP<Number>\_TLS\_<curve>.pem | Public Key of the SM-DP+ for TLS |
| CERT\_S\_SM\_DP<Number> \_<variant>\_TLS\_<curve>.der | Certificate of the SM-DP+ for TLS |

1. : TLS Keys and Certificates of SM-DP+

In order to generate the different files, next commands must be performed using the previous values and following input files

|  |  |
| --- | --- |
| **Common input files** | **Description** |
| SK\_S\_SM\_DP<number>\_TLS\_<curve>.pem | Private key of the SM-DP+ TLS |
| SM\_DP<number>tls-csr.cnf | CSR configuration file (See Annex F) |
| SM\_DP<number>tls-ext.cnf | Certificate configuration file (See Annex F) |

$ openssl req -new -nodes -<signature> -config SM\_DP<number>tls-csr.cnf
 -key SK\_S\_SM\_DP<number>\_TLS\_<curve>.pem -out SM\_DP\_TLS\_<curve>.csr

**Variant O**

|  |  |
| --- | --- |
| **Specific variant input files** | **Description** |
| CERT\_CI\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_CI\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DPtls\_<curve>.csr
 -CA CERT\_CI\_<curve>.pem -Cakey SK\_CI\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DP<number>tls-ext.cnf
 -out CERT\_S\_SM\_DP<number>\_TLS\_<curve>.pem

**Variant B**

|  |  |
| --- | --- |
| **Specific variant input files** | **Description** |
| CERT\_CISubCA\_<curve>.pem | Certificate Issuer |
| SK\_CISubCA\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DPtls\_<curve>.csr
 -CA CERT\_CISubCA\_<curve>.pem -Cakey SK\_CISubCA\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DP<number>tls-ext.cnf
 -out CERT\_S\_SM\_DP<number>\_<variant>\_TLS\_<curve>.pem

**Variant A and C**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_SM\_DPSubCA\_<variant>\_<curve>.pem | Certificate Issuer |
| SK\_SM\_DPSubCA\_<variant>\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SMDPpb\_<curve>.csr
 -CA CERT\_SM\_DPSubCA\_<variant>\_<curve>.pem -Cakey SK\_SM\_DPSubCA\_<variant>\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DP<number>tls-ext.cnf
 -out CERT\_S\_SM\_DP<number>\_<variant>\_TLS\_<curve>.pem

## SM-DS

### SM-DS SubCA

Note: CERT.SM\_DSSubCA.SIG is only defined for Variant A and Variant C

| Field | Value  |
| --- | --- |
| version | *<Value automatically set>* |
| serialNumber | See Annex E.1 |
| signature | sha256 |
| Issuer | For Variant A:*<Value of CERT.CI.SIG subject field>*For Variant C:*<Value of CERT.CISubCA.SIG subject field>* |
| Validity | 1095 days (3 years) |
| Subject | cn = Test CI SM\_DSSubCAou = TESTCERTo = RSPTESTc = ES |
| Extensions |  |
| subjectKeyIdentifier | hash |
| authorityKeyIdentifier | keyid, issuer |
| keyUsage | critical, keyCertSign, cRLSign |
| certificatePolicies  | 2.23.146.1.2.1.0.0.2 |
| basicConstraints | critical, cA:truepathLenConstraint = 0 |
| subjectAltName  | See Annex E.2 |
| crlDistributionPoints | For Variant A: <Value of CERT.CI.SIG "crlDistributionPoints" field>For Variant C: <Value of CERT.CISubCA.SIG "crlDistributionPoints" field> |

1. : CERT.SM\_DSSubCA.SIG data

Hereafter the generated SM-DP+ SubCA keys and certificates as defined in Annex A.

| File name | Description |
| --- | --- |
| SK\_SM\_DSSubCA\_SIG\_<curve>.pem | Private key of the SM-DS SubCA  |
| PK\_SM\_DSSubCA\_SIG\_<curve>.pem | Public Key of the SM-DS SubCA |
| CERT\_SM\_DSSubCA\_<variant>\_SIG\_<curve>.der | Certificate of the SM-DS SUBCA  |

1. : EUM SubCA Keys and Certificates

In order to generate the different files, next commands must be performed using the previous values and following input files

|  |  |
| --- | --- |
| Common input files | Description |
| SK\_SM\_DSSubCA\_<curve>.pem | Private key of the Certificate Issuer SubCA |
| SubCA\_csr.cnf | CSR configuration file (See Annex F) |
| SubCA\_ext.cnf | Certificate configuration file (See Annex F) |

$ openssl req -new -nodes -<signature> -config SM\_DSSubCA-csr.cnf
 -key SK\_SM\_DSSubCA\_<curve>.pem -out SM\_DS\_SubCA\_<curve>.csr
 **Variant A**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_CI\_<curve>.pem | Certificate Issuer |
| SK\_CI\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DS\_SubCA\_<curve>.csr
 -CA CERT\_CI\_<curve>.pem -Cakey SK\_CI\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DSSubCA-ext.cnf
 -out CERT\_SM\_DSSubCA\_<variant>\_<curve>.pem

**Variant C**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_CISubCA\_<curve>.pem | Certificate Issuer |
| SK\_CISubCA\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DS\_SubCA\_<curve>.csr
 -CA CERT\_CISubCA\_<curve>.pem -Cakey SK\_CISubCA\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DSSubCA-ext.cnf
 -out CERT\_SM\_DSSubCA\_<variant>\_<curve>.pem

### SM-DSauth

#### SM-DS Certificate for Authentication: definition of data to be signed

|  |  |
| --- | --- |
| Field | Value  |
| Version | <Value automatically set> |
| serialNumber | See Annex E.1 |
| Signature | sha256 |
| Issuer | For Variant O: <Value of CERT.CI.SIG "subject" field>For Variant B:<Value ofCERT.CISubCA.SIG “subject” fieldFor Variant A and C: <Value ofCERT.SM\_DSSubCA.SIG “subject” field |
| Validity | 1095 days (3 years) |
| Subject | o = ‘ACME’cn = ‘TEST SM-DS <SM-DS number>’ |
| Extensions |  |
| authorityKeyIdentifier | eyed,issuer |
| subjectKeyIdentifier | hash |
| keyUsage | critical, digitalSignature |
| certificatePolicies | For Variant O: critical, 2.23.146.1.2.1.7For Variant A, B and C: critical, 2.23.146.1.2.1.0.0.2.1 |
| subjectAltName | See Annex E.2 |
| crlDistributionPoints | For Variant O:<Value of CERT.CI.SIG. ”crlDistributionPoints” field>For Variant B:<Value of CERT.CISubCA.SIG “crlDistributionPoints” field>For Variant A and C:<Value of CERT.SM\_DSSubCA.SIG "crlDistributionPoints" field> |

1. : CERT.SM\_DSauth.SIG

Hereafter the generated keys and certificates of SM-DS for Authentication as defined in Annex A.

| File name | Description |
| --- | --- |
| SK\_S\_SM\_DS<Number>auth\_<variant>\_ SIG\_<curve>.pem | Private Key of the SM-DS for creating signatures for SM-DS authentication |
| PK\_S\_SM\_DS<Number>auth\_<variant>\_SIG\_<curve>.pem | Public Key of the SM-DS |
| CERT\_S\_SM\_DS<Number>auth\_<variant>\_SIG\_<curve>.der | Certificate of the SM-DS |

1. : DSAuth Keys and Certificates of SM-DS

In order to generate the different files, next commands must be performed using the previous values and following input files:

|  |  |
| --- | --- |
| Common input files | Description |
| SK\_S\_SM\_DS<number>auth\_SIG\_<curve>.pem | Private key of the SM-DS for Authentication |
| SM\_DS<number>aut-csr.cnf | CSR configuration file (See Annex F) |
| SM\_DS<number>auth-ext.cnf | Certificate configuration file (See Annex F) |

$ openssl req -new -nodes -<signature> -config DS<number>auth-csr.cnf
 -key SK\_S\_SM\_DS<number>auth\_<curve>.pem -out SMDSauth\_<curve>.csr

**Variant O**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_CI\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_CI\_SIG\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DSauth\_<curve>.csr
 -CA CERT\_CI\_SIG\_<curve>.pem -Cakey SK\_CI\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DS<number>auth-ext.cnf
 -out CERT\_S\_SM\_DS<number>auth\_SIG\_<curve>.pem

**Variant B**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_CISubCA\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_CISubCA\_SIG\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DSauth\_<curve>.csr
 -CA CERT\_CISubCA\_SIG\_<curve>.pem -Cakey SK\_CISubCA\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DS<number>auth-ext.cnf
 -out CERT\_S\_SM\_DS<number>auth\_<variant>\_SIG\_<curve>.pem

**Variant A and C**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_SM\_DSSubCA\_<variant>\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_SM\_DSSubCA\_<variant>\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DSauth\_<curve>.csr
 -CA CERT\_SM\_DSSubCA\_<variant>\_SIG\_<curve>.pem -Cakey SK\_SM\_DSSubCA\_<variant>\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DS<number>auth-ext.cnf
 -out CERT\_S\_SM\_DS<number>auth\_<variant>\_SIG\_<curve>.pem

### TLS

#### SM-DS n°1 TLS Certificate: definition of data to be signed

| Field | Value  |
| --- | --- |
| Version | <automatically set> |
| serialNumber | See Annex E.1 |
| Signature | SHA256 |
| Issuer | For Variant O: <Value of CERT.CI.SIG."subject" field>For Variant A and C:<Value of CERT.SM\_DSSubCA.SIG."subject" field> For Variant B:<Value of CERT.CISubCA.SIG."subject" field>  |
| Validity | 398 days  |
| Subject | o = 'RSPTEST' cn = 'testrootsmds.example.com' |
| Extensions |   |
| authority Key Identifier | keyId, issuer |
| subject Key Identifier | hash |
| key usage | CriticaldigitalSignature |
| Extension for Certificate Policies | For Variant O: '2.23.146.1.2.1.6' For Variant Ov3, A, B, C: critical, 2.23.146.1.2.1.0.0.2.0 |
| Extension for Extended Key usage | critical, serverAuth, clientAuth |
| subjectAltName | See Annex E.2 |
| Extension for CRL Distribution Points | <Value of CERT.CI.SIG."crlDistributionPoints" field> |

1. : CERT.DS.TLS for SM-DS

Hereafter the generated SM-DS keys and certificates for TLS as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DS<number>\_TLS\_<curve>.pem | Private key of the SM-DS for securing TLS connection |
| PK\_S\_SM\_DS<number>\_<variant>\_TLS\_<curve>.pem | Public Key of the SM-DS |
| CERT\_S\_SM\_DS<number>\_<variant>\_TLS\_<curve>.der | Certificate of the SM-DS  |

1. : DS\_TLS Keys and Certificates for SM-DS

In order to generate the different files, next commands must be performed using the previous values and following input files

|  |  |
| --- | --- |
| **Common input files** | **Description** |
| SK\_S\_SM\_DS<number>\_TLS\_<curve>.pem | Private key of the SM-DS for TLS |
| SM\_DS<number>tls-csr.cnf | CSR configuration file (See Annex F) |
| SM\_DS<number>tls-ext.cnf | Certificate configuration file (See Annex F) |

$ openssl req -new -nodes -<signature> -config SM\_DS<number>tls-csr.cnf
 -key SK\_S\_SM\_DS<number>\_TLS\_<curve>.pem -out SM\_DS\_TLS\_<curve>.csr

**Variant O**

|  |  |
| --- | --- |
| **Specific variant input files** | **Description** |
| CERT\_CI\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_CI\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DStls\_<curve>.csr
 -CA CERT\_CI\_<curve>.pem -Cakey SK\_CI\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DS<number>tls-ext.cnf
 -out CERT\_S\_SM\_DS<number>\_TLS\_<curve>.pem

**Variant B**

|  |  |
| --- | --- |
| **Specific variant input files** | **Description** |
| CERT\_CISubCA\_<curve>.pem | Certificate Issuer |
| SK\_CISubCA\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DStls\_<curve>.csr
 -CA CERT\_CISubCA\_<curve>.pem -Cakey SK\_CISubCA\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DS<number>tls-ext.cnf
 -out CERT\_S\_SM\_DS<number>\_<variant>\_TLS\_<curve>.pem

**Variant A and C**

|  |  |
| --- | --- |
| Specific variant input files | Description |
| CERT\_SM\_DSSubCA\_<variant>\_<curve>.pem | Certificate Issuer |
| SK\_SM\_DSSubCA\_<variant>\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in SM\_DSpb\_<curve>.csr
 -CA CERT\_SM\_DSSubCA\_<variant>\_<curve>.pem -Cakey SK\_DSSubCA\_<variant>\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile SM\_DS<number>tls-ext.cnf
 -out CERT\_S\_SM\_DS<number>\_<variant>\_TLS\_<curve>.pem

## eIM

### eIM Certificate for Signing

#### eIM Certificate for Signing definition of data to be signed

| Field | Value  |
| --- | --- |
| Version | *<Value automatically set>* |
| serialNumber | See Annex E.1 |
| signature |  sha256 |
| Issuer | <Value of CERT.CI.SIG "subject" field> |
| Validity | 2555 days (7 years) |
| Subject | CN = eim.example.comC=DE |
| Extensions  |  |
| authorityKeyIdentifier | keyid, issuer |
| keyUsage | Critical, digitalSignature  |

1. : CERT.EIM.ECDSA data

In order to generate the different files, next commands must be performed using the previous values and following input files

|  |  |
| --- | --- |
| Input files | Description |
| SK\_EIMsign\_<curve>.pem | Private key of the eIM |
| EIM\_csr.cnf | CSR configuration file (See Annex F) |
| EIM\_ext.cnf | Certificate configuration file (See Annex F) |

$ openssl req -new -nodes -<signature> -config EIM\_csr.cnf
 -key SK\_EIMsign\_<curve>.pem -out EIMsign\_<curve>.csr

|  |  |
| --- | --- |
| Input files | Description |
| CERT\_CI\_<curve>.pem | Certificate Issuer |
| SK\_CI\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl x509 -req -in EIMsign\_<curve>.csr
 -CA CERT\_CI\_<curve>.pem -Cakey SK\_CI\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile EIM\_ext.cnf
 -out CERT\_S\_EIMsign\_<curve>.pem

#### eIM Keys and Certificate

Hereafter the generated keys and certificates of eIM for Signing as defined in Annex A.

| File name | Description |
| --- | --- |
| SK\_S\_EIMsign\_ECDSA\_NIST.pem | NIST P-256 Private Key of the eIM for signing eUICC package requests |
| PK\_S\_EIMsign\_ECDSA\_NIST.pem | NIST P-256 Public Key of the eIM(part of the CERT\_S\_EIMsign\_ECDSA\_NIST.der) |
| CERT\_S\_EIMsign\_ECDSA\_NIST.der | Certificate of the eIM for its Public NIST P-256 key used for signature verification |
|  |
| SK\_S\_EIMsign\_ECDSA\_BRP.pem | Brainpool P256r1 Private Key of the eIM for signing eUICC package requests |
| PK\_S\_EIMsign\_ECDSA\_BRP.pem | Brainpool P256r1 Public Key of the eIM(part of the CERT\_S\_EIMsign\_ECDSA\_BRP.der) |
| CERT\_S\_EIMsign\_ECDSA\_BRP.der | Certificate of the eIM for its Public Brainpool P256r1 key used for signature verification |

1. : Signing Keys and Certificates of eIM

#### Input data for generation

The SK.EIMsign.ECDSA and PK.EIMsign.ECDSA of the eIM are generated using the command lines as described in section 2.2.

The related CERT.EIMsign.ECDSA is generated using the command lines described in section 2.4 with the following input data:

<input\_csr\_file\_name>: EIM\_csr.cnf as defined in Annex F.

<ca\_cert\_file\_name> and <ca\_sk\_file\_name>: files generated in section 3.1.2 (file containing the CERT.CI.ECDSA and SK.CI.ECDSA respectively).

<serial> set with value defined in section 3.6.1.1 for serialNumber data field.

<days> set with value defined in section 3.6.1.1 for validity data field.

<cert\_ext\_file\_name>: EIM\_ext.cnf as defined in Annex F.

### TLS/DTLS

#### eIM TLS/DTLS Certificate: definition of data to be signed

| Field | Value |
| --- | --- |
| Version | <automatically set> |
| serialNumber | See Annex E.1 |
| signature | sha256 |
| Issuer | <Value of CERT.CI.SIG."subject" field> |
| validity | 398 days  |
| subject | C = 'DE'CN = 'eim.example.com' |
| Extensions |  |
| authorityKeyIdentifier | keyId, issuer |
| keyUsage | criticaldigitalSignature  |
| extendedKeyUsage | critical, serverAuth, clientAuth |
| subjectAltName | See Annex E.2 |
| crlDistributionPoints | <Value of CERT.CI.SIG."crlDistributionPoints" field> |

1. : CERT.EIM.TLS

Hereafter the generated keys and certificates for TLS/DTLS as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_EIM\_TLS\_<curve>.pem | Private Key of the EIM for TLS/DTLS |
| PK\_S\_EIM\_TLS\_<curve>.pem | Public Key of the EIM for TLS/DTLS |
| CERT\_S\_EIM\_TLS\_<curve>.der | Certificate of the EIM for TLS/DTLS |

1. : EIM TLS/DTLS Keys and Certificates of SM-DP+

In order to generate the different files, next commands must be performed using the previous values and following input files

|  |  |
| --- | --- |
| **Input files** | **Description** |
| SK\_S\_EIM\_TLS\_<curve>.pem | Private key of the EIM TLS/DTLS |
| EIMtls-csr.cnf | CSR configuration file (See Annex F) |
| EIMtls-ext.cnf | Certificate configuration file (See Annex F) |
| CERT\_CI\_SIG\_<curve>.pem | Certificate Issuer |
| SK\_CI\_<curve>.pem | Private key of the Certificate Issuer |

$ openssl req -new -nodes -<signature> -config EIMtls-csr.cnf
 -key SK\_S\_EIM\_TLS\_<curve>.pem -out EIM\_TLS\_<curve>.csr

$ openssl x509 -req -in EIM\_TLS\_<curve>.csr
 -CA CERT\_CI\_<curve>.pem -Cakey SK\_CI\_<curve>.pem
 -set\_serial <serialNumber> -days <validity> -extfile EIMtls-ext.cnf
 -out CERT\_S\_EIM>\_TLS\_<curve>.pem

# Test Certificates and keys – Invalid test cases

The sections below describe

* The data structure and content of the certificates used for running the invalid test cases in SGP.23;
* how such certificates are derived: both the toolchain and the input data are described.

## SM-DP+

### SM-DP+ Certificate for Authentication

#### SM-DP+ Certificate for Authentication – Invalid Signature

All the data to be signed are the same as the ones defined in section 3.3.2 for all the variants..

The openSSL script for generation can be found in Section 3.3.2.

Hereafter the SM-DP+ certificates for Authentication with invalid signature as defined in Annex A.

|  |  |
| --- | --- |
| **File name** | **Description** |
| CERT\_S\_SM\_DPauth\_INV\_SIGN\_NIST\_<Variant>.der | Certificate of the SM-DP+ with invalid signature for its Public NIST P-256 key used for SM‑DP+ authentication |
|  |
| CERT\_S\_SM\_DPauth\_INV\_SIGN\_BRP\_<Variant>.der | Certificate of the SM-DP+ with invalid signature for its Public Brainpool P256r1 key used for SM‑DP+ authentication |

1. : SM\_DPauth\_INV\_SIGN Certificates

Few bytes of the generated signatures contained in the DER files have been manually changed as follow:

* NIST signature: The 10 last bytes of the DER file are replaced by “00000000000000000000” (hexadecimal representation)
* Brainpool signature: The 8 last bytes of the DER file are replaced by “111111111111” (hexadecimal representation)

#### SM-DP+ Certificate for Authentication – Invalid Curve

The Elliptic Curves NIST P-192 and Brainpool P192r1 are chosen for triggering the Authenticate and Download Error Code unsupportedCurve(3) as defined in SGP.22 [1].

##### All the data to be signed are the same as the ones defined in section 3.3.2 for all the variants.

Hereafter the SM-DP+ certificates and keys for Authentication with invalid curve as defined in Annex A.

| File name | Description |
| --- | --- |
| SK\_S\_SM\_DPauth\_SIG\_NIST192.pem | NIST P-192 Private Key of the SM-DP+ for creating signatures for SM-DP+ authentication |
| PK\_S\_SM\_DPauth\_SIG\_NIST192.pem | NIST P-192 Public Key of the SM-DP+ (part of the CERT\_S\_SM\_DPauth\_INV\_CURVE\_NIST192.der) |
| CERT\_S\_SM\_DPauth\_INV\_CURVE\_NIST192\_<Variant>.der | Certificate of the SM-DP+ for its Public NIST P-192 key used for SM-DP+ authentication |
|  |  |
| SK\_S\_SM\_DPauth\_SIG\_BRP192.pem | Brainpool P-192 Private Key of the SM-DP+ for creating signatures for SM-DP+ authentication |
| PK\_S\_SM\_DPauth\_SIG\_BRP192.pem | Brainpool P-192 Public Key of the SM-DP+ (part of the CERT\_S\_SM\_DPauth\_INV\_CURVE\_BRP192.der) |
| CERT\_S\_SM\_DPauth\_INV\_CURVE\_BRP192\_<Variant>.der | Certificate of the SM-DP+ for its Public Brainpool P-192 key used for SM-DP+ authentication |

1. : SM-DP+ forAuthentication Keys and Certificates with invalid curve

In order to generate the different files, next commands must be performed using the previous values and following input files:

Command lines for the generation of the SK.DPauth.SIG and the corresponding PK.DPauth.SIG for NIST P-192 curve:

openssl ecparam –name prime192v1 –genkey –out SK\_S\_SM\_DPauth\_SIG\_NIST192.pem

openssl ec –in SK\_S\_SM\_DPauth\_SIG\_NIST192.pem –pubout –out PK\_S\_SM\_DPauth\_SIG\_NIST192.pem

Command lines for the generation of the SK.SM\_DPauth.SIG and the corresponding PK.SM\_DPauth.SIG for Brainpool P192r1 curve:

openssl ecparam –name brainpoolP192r1 –genkey –out SK\_S\_SM\_DPauth\_SIG\_BRP192.pem

openssl ec –in SK\_S\_SM\_DPauth\_SIG\_BRP192.pem –pubout –out PK\_S\_SM\_DPauth\_SIG\_BRP192.pem

The CERT.SM\_DPauth.SIG are generated using the command lines described in section 3.3.2. for each variant.

### SM-DP+ Certificate for Profile Binding

#### SM-DP+ Certificate for Profile Biding – Invalid Signature

All the data to be signed are the same as the ones defined in 3.3.3 for all the variants.

The openSSL script for generation can be found in Section 3.3.3.

Hereafter the SM-DP+ certificates for Profile Package Binding with invalid signature as defined in Annex A.

|  |  |
| --- | --- |
| **File name** | **Description** |
| CERT\_S\_SM\_DPpb\_INV\_SIGN\_NIST\_<Variant>.der | Certificate of the SM-DP+ with invalid signature for its Public NIST P-256 key used for Profile Package Binding |
|  |
| CERT\_S\_SM\_DPpb\_INV\_SIGN\_BRP\_<Variant>.der | Certificate of the SM-DP+ with invalid signature for its Public Brainpool P256r1 key used for Profile Package Binding |

1. : DPpb Certificates with invalid signature

Few bytes of the generated signatures contained in the DER files have been manually changed as follow for each variant:

NIST signature: The 10 last bytes of the DER file are replaced by “00000000000000000000” (hexadecimal representation)

Brainpool signature: The 8 last bytes of the DER file are replaced by “111111111111” (hexadecimal representation)

#### SM-DP+ Certificate for Profile Binding – Invalid Curve

The Elliptic Curves NIST P-192 and Brainpool P192r1 are chosen for triggering the Authenticate and Download Error Code unsupportedCurve(3) as defined in SGP.22 [1].

All the data to be signed are the same as the ones defined in 3.3.3 for all the variants.

Hereafter the SM-DP+ certificates and keys for Profile Binding with invalid curve as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DPpb\_SIG\_NIST192.pem | NIST P-192 Private Key of the SM-DP+ for creating signatures for Profile Package Binding |
| PK\_S\_SM\_DPpb\_SIG\_NIST192.pem | NIST P-192 Public Key of the SM-DP+ (part of the CERT\_S\_SM\_DPpb\_INV\_CURVE\_NIST192.der) |
| CERT\_S\_SM\_DPpb\_INV\_CURVE\_NIST192\_<Variant>.der | Certificate of the SM-DP+ for its Public NIST P-192 key used for Profile Package Binding |
|  |  |
| SK\_S\_SM\_DPpb\_SIG\_BRP192.pem | Brainpool P-192 Private Key of the SM-DP+ for creating signatures for Profile Package Binding |
| PK\_S\_SM\_DPpb\_SIG\_BRP192.pem | Brainpool P-192 Public Key of the SM-DP+ (part of the CERT\_S\_SM\_DPpb\_INV\_CURVE\_BRP192.der) |
| CERT\_S\_SM\_DPpb\_INV\_CURVE\_BRP192\_<Variant>.der | Certificate of the SM-DP+ for its Public Brainpool P-192 key used for Profile Package Binding |

1. : SM\_DPpb Keys and Certificates with invalid curve

In order to generate the different files, next commands must be performed using the previous values and following input files:

Command lines for the generation of the SK.SM\_DPpb.SIG and the corresponding PK.SM\_DPpb.SIG for NIST P-192 curve:

openssl ecparam –name prime192v1 –genkey –out SK\_S\_SM\_DPpb\_SIG\_NIST192.pem

openssl ec –in SK\_S\_SM\_DPpb\_SIG\_NIST192.pem –pubout –out PK\_S\_SM\_DPpb\_SIG\_NIST192.pem

Command lines for the generation of the SK.DPpb.SIG and the corresponding PK.DPpb.SIG for Brainpool P192r1 curve:

openssl ecparam –name brainpoolP192r1 –genkey –out SK\_S\_SM\_DPpb\_SIG\_BRP192.pem

openssl ec –in SK\_S\_SM\_DPpb\_SIG\_BRP192.pem –pubout –out PK\_S\_SM\_DPpb\_SIG\_BRP192.pem

The CERT.SM\_DPpb.SIG are generated using the command lines described in section 3.3.3 for each variant.

### SM-DP+ TLS Certificate

#### SM-DP+ TLS Certificate – Invalid Signature

All the data to be signed are the same as the ones defined in section 3.3.

Hereafter the SM-DP+ TLS certificates with invalid signature as defined in Annex A.

|  |  |
| --- | --- |
| **File name** | **Description** |
| CERT\_S\_SM\_DP\_TLS\_INV\_SIGN\_NIST\_<Variant>.der | Certificate of the SM-DP+ with invalid signature for its Public NIST P-256 key  |
|  |
| CERT\_S\_SM\_DP\_TLS\_INV\_SIGN\_BRP\_<Variant>.der | Certificate of the SM-DP+ with invalid signature for its Public Brainpool P256r1 key |

1. : SM\_DP\_TLS Certificates with invalid signature

Few bytes of the generated signatures contained in the DER files have been manually changed as follow, for each variant:

* NIST signature: The 10 last bytes of the DER file are replaced by “00000000000000000000” (hexadecimal representation)
* Brainpool signature: The 8 last bytes of the DER file are replaced by “111111111111” (hexadecimal representation)

#### SM-DP+ TLS Certificate – Invalid Curve

The Elliptic Curves NIST P-384 is chosen for triggering the Authenticate and Download Error Code unsupportedCurve(3) as defined in SGP.22 [1].

All the data to be signed are the same as the ones defined in Section 3.3.4 for all the variants.

Hereafter the generated SM-DP+ keys and certificates for TLS.

| File name | Description |
| --- | --- |
| SK\_CERT\_CI\_S\_SM\_DP\_NIST\_P384.pem | NIST P-384 Private CI key of the SM-DP+ for securing TLS connection with |
| PK\_CERT\_CI\_S\_SM\_DP\_NIST\_P384.pem | NIST P-384 Public CI Key of the SM-DP+  |
| SK\_ S\_SM\_DP\_TLS\_ NIST\_P384.pem | NIST P-384 Private key of the SM-DP+ for TLS (invalid curve) |
| PK\_ S\_SM\_DP\_TLS\_ NIST\_P384.pem | NIST P-384 Public Key of the SM-DP+ for TLS(invalid curve) |
| CERT\_S\_SM\_DP\_TLS\_INV\_CURVE\_<Variant>.der | CERT.DP.TLS certificate of the S\_SM-DP+, based on NIST P-384 curve  |

1. : SM-DP+ TLS Keys and Certificates with invalid curve

In order to generate the different files, next commands must be performed using the previous values and following input files:

Command lines for the generation of the SK.DP.TLS and the corresponding PK.DP.TLS for NIST P-384 curve:

openssl ecparam –name secp384r1 –genkey –out SK\_S\_SM\_DP\_TLS\_NIST\_P384.pem

openssl ec –in SK\_S\_SM\_DP\_TLS\_NIST\_P384.pem –pubout –out PK\_S\_SM\_DP\_TLS\_NIST\_P384.pem

The CERT.DP.TLS are generated using the command lines described in section Section 3.3.4 for each variant.

#### SM-DP+ TLS Certificate – Invalid Certificate Policy

In order to generate the different files, the commands in section 3.3.4 shall be used.

The input data and files for generation applies with no changes except for the value of <certificatePolicies> extension that shall be set as follows:

* For Variant O: critical, 2.23.146.1.2.1.4
* For Variant A, B and C: critical, 2.23.146.1.2.1.0.0.1.1

Hereafter the generated SM-DP+ keys and certificates for TLS as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DP\_TLS\_NIST.pem | NIST P-256 Private key of the SM-DP+ for securing TLS connection |
| PK\_S\_SM\_DP\_TLS\_NIST.pem | NIST P-256 Public Key of the SM-DP+ (part of the CERT\_S\_SM\_DP\_TLS\_NIST.der) |
| CERT\_S\_SM\_DP\_TLS\_INV\_CERT\_POL\_<Variant>.der | CERT.SM\_DP.TLS certificate of the S\_SM-DP+ with invalid 'Certificate Policies' extension (OID set to 'id-rspRole-dp-auth'), formatted as X.509 certificate. |

#### SM-DP+ TLS Certificate – Missing Critical Extension

In order to generate the different files, the commands in section 3.3.4 shall be used.

The input data and files for generation applies with no changes except for the value of <extendedKeyUsage> extension that shall be set as follows:

* Absent

Hereafter the generated SM-DP+ keys and certificates for TLS as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DP\_TLS\_NIST.pem | NIST P-256 Private key of the SM-DP+ for securing TLS connection |
| PK\_S\_SM\_DP\_TLS\_NIST.pem | NIST P-256 Public Key of the SM-DP+ (part of the CERT\_S\_SM\_DP\_TLS\_NIST.der) |
| CERT\_S\_SM\_DP\_TLS\_INV\_CRITICAL\_EXT\_<Variant>.der | CERT.SM\_DP.TLS certificate of the S\_SM-DP+ with one of the critical extensions not present, formatted as X.509 certificate. |

1. : DP\_TLS Keys and Certificates with critical extension not present

#### SM-DP+ TLS Certificate – Invalid Extended Key Usage

In order to generate the different files, the commands in section 3.3.4 shall be used.

The input data and files for generation applies with no changes except for the value of <extendedKeyUsage> extension that shall be set as follows:

* critical, clientAuth

Hereafter the generated SM-DP+ keys and certificates for TLS as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DP\_TLS\_NIST.pem | NIST P-256 Private key of the SM-DP+ for securing TLS connection |
| PK\_S\_SM\_DP\_TLS\_NIST.pem | NIST P-256 Public Key of the SM-DP+ (part of the CERT\_S\_SM\_DP\_TLS\_NIST.der) |
| CERT\_S\_SM\_DP\_TLS\_INV\_EXT\_KEY\_USAGE\_<Variant>.der | CERT.SM\_DP.TLS certificate of the S\_SM-DP+ with invalid 'extended key usage' extension (not set to 'id-kp-serverAuth'), formatted as X.509 certificate. |

1. : DP+ TLS Certificates with invalid 'extended key usage'

#### SM-DP+ TLS Certificate – Invalid Key Usage

In order to generate the different files, the commands in section 3.3.4 shall be used.

The input data and files for generation applies with no changes except for the value of <keyUsage> extension that shall be set as follows:

* critical, keyAgreement

Hereafter the generated SM-DP+ keys and certificates for TLS as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DP\_TLS\_NIST.pem | NIST P-256 Private key of the SM-DP+ for securing TLS connection |
| PK\_S\_SM\_DP\_TLS\_NIST.pem | NIST P-256 Public Key of the SM-DP+ (part of the CERT\_S\_SM\_DP\_TLS\_NIST.der) |
| CERT\_S\_SM\_DP\_TLS\_INV\_KEY\_USAGE\_<Variant>.der | CERT.SM\_DP.TLS certificate of the S\_SM-DP+ with invalid 'key usage' extension (not set to 'digitalSignature'), formatted as X.509 certificate. |

1. : DP+ TLS Keys and Certificates with invalid 'key usage' extension

#### SM-DP+ TLS Certificate – Expired Certificate

In order to generate the different files, the commands in section X.Y.Z shall be used.

The input data and files for generation applies with no changes except for the value of <Validity> field that shall be set as follows:

* 1 day

Hereafter the generated SM-DP+ keys and certificates for TLS as defined in Annex A.

|  |  |
| --- | --- |
| **File name** | **Description** |
| SK\_S\_SM\_DP\_TLS\_NIST.pem | NIST P-256 Private key of the SM-DP+ for securing TLS connection |
| PK\_S\_SM\_DP\_TLS\_NIST.pem | NIST P-256 Public Key of the SM-DP+ (part of the CERT\_S\_SM\_DP\_TLS\_NIST.der) |
| CERT\_S\_SM\_DP\_TLS\_EXPIRED\_<Variant>.der | Expired CERT.SM\_DP.TLS certificate of the S\_SM-DP+ with a valid signature, correctly formatted as X.509 certificate. |

1. : DP+ TLS Keys and expired Certificates

## SM-DS

### SM-DS Certificate for Authentication

#### SM-DS Certificate for Authentication – Invalid Signature

All the data to be signed are the same as the ones defined in X.Y for all the variants

Hereafter the SM-DS certificates for Authentication with invalid signature as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| CERT\_S\_SM\_DSauth\_INV\_SIGN\_NIST\_<Variant>.der | Certificate of the SM-DS with invalid signature for its Public NIST P-256 key used for SM‑DP+ authentication |
|  |
| CERT\_S\_SM\_DSauth\_INV\_SIGN\_BRP\_<Variant>.der | Certificate of the SM-DS with invalid signature for its Public Brainpool P256r1 key used for SM‑DP+ authentication |

1. : SM\_DS TLS Certificates with invalid signature

Few bytes of the generated signatures contained in the DER files have been manually changed as follow:

* NIST signature: The 10 last bytes of the DER file are replaced by “00000000000000000000” (hexadecimal representation)
* Brainpool signature: The 8 last bytes of the DER file are replaced by “111111111111” (hexadecimal representation)

#### SM-DS Certificate for Authentication - Invalid curve

The Elliptic Curve NIST P-192 and Brainpool P192r1 are chosen for triggering the Authenticate Error Code unsupportedCurve(3) as defined in SGP.22 [1].

All the data to be signed are the same as the ones defined in 3.x for all the variants.

Hereafter the SM-DS certificates and keys for Authentication with invalid curve as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DSauth\_SIG\_NIST192.pem | NIST P-192 Private Key of the SM-DS for creating signatures for SM-DS authentication |
| PK\_S\_SM\_DSauth\_SIG\_NIST192.pem | NIST P-192 Public Key of the SM-DS (part of the CERT\_S\_SM\_DSauth\_INV\_CURVE\_NIST192.der) |
| CERT\_S\_SM\_DSauth\_INV\_CURVE\_NIST192\_<Variant>.der | Certificate of the SM-DS for its Public NIST P-192 key used for SM-DS authentication |
|  |  |
| SK\_S\_SM\_DSauth\_SIG\_BRP192.pem | Brainpool P-192 Private Key of the SM-DS for creating signatures for SM-DS authentication |
| PK\_S\_SM\_DSauth\_SIG\_BRP192.pem | Brainpool P-192 Public Key of the SM-DS (part of the CERT\_S\_SM\_DSauth\_INV\_CURVE\_BRP192.der) |
| CERT\_S\_SM\_DSauth\_INV\_CURVE\_BRP192\_<Variant>.der | Certificate of the SM-DS for its Public Brainpool P-192 key used for SM-DS authentication |

1. : SM-DS Certificates with invalid curve

In order to generate the different files, next commands must be performed using the previous values and following input files:

Command lines for the generation of the SK.DSauth.SIG and the corresponding PK.DSauth.SIG for NIST P-192 curve:

openssl ecparam –name prime192v1 –genkey –out SK\_S\_SM\_DSauth\_SIG\_NIST192.pem

openssl ec –in SK\_S\_SM\_DSauth\_SIG\_NIST192.pem –pubout –out PK\_S\_SM\_DSauth\_SIG\_NIST192.pem

Command lines for the generation of the SK.DSauth.SIG and the corresponding PK.DSauth.SIG for Brainpool P-192 curve:

openssl ecparam –name brainpoolP192r1 –genkey –out SK\_S\_SM\_DSauth\_SIG\_BRP192.pem

openssl ec –in SK\_S\_SM\_DSauth\_SIG\_BRP192.pem –pubout –out PK\_S\_SM\_DSauth\_SIG\_BRP192.pem

The CERT.DSauth.SIG are generated using the command lines described in section 3.4.2

### SM-DS TLS Certificate

#### SM-DS TLS Certificate – Invalid Signature

All the data to be signed are the same as the ones defined in 3.4.3.

Hereafter the SM-DS TLS certificates with invalid signature as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| CERT\_S\_SM\_DS\_TLS\_INV\_SIGN\_NIST\_<Variant>.der | Certificate of the SM-DS with invalid signature for its Public NIST P-256 key  |
|  |
| CERT\_S\_SM\_DS\_TLS\_INV\_SIGN\_BRP\_<Variant>.der | Certificate of the SM-DS with invalid signature for its Public Brainpool P256r1 key |

1. : SM-DS TLS Certificates with invalid signature

Few bytes of the generated signatures contained in the DER files have been manually changed as follow, for each variant:

* NIST signature: The 10 last bytes of the DER file are replaced by “00000000000000000000” (hexadecimal representation)
* Brainpool signature: The 8 last bytes of the DER file are replaced by “111111111111” (hexadecimal representation)

#### SM-DS TLS Certificate – Invalid Curve

The Elliptic Curves NIST P-384 is chosen for triggering the Authenticate and Download Error Code unsupportedCurve(3) as defined in SGP.22 [1].

All the data to be signed are the same as the ones defined in 3.x for all the variants.

Hereafter the generated SM-DS keys and certificates for TLS.

|  |  |
| --- | --- |
| File name | Description |
| SK\_CERT\_CI\_S\_SM\_DS\_NIST\_P384.pem | NIST P-384 Private CI key of the SM-DS for securing TLS connection with |
| PK\_CERT\_CI\_S\_SM\_DS\_NIST\_P384.pem | NIST P-384 Public CI Key of the SM-DS  |
| SK\_ S\_SM\_DS\_TLS\_ NIST\_P384.pem | NIST P-384 Private key of the SM-DS for TLS (invalid curve) |
| PK\_ S\_SM\_DS\_TLS\_ NIST\_P384.pem | NIST P-384 Public Key of the SM-DS for TLS(invalid curve) |
| CERT\_S\_SM\_DS\_TLS\_INV\_CURVE\_<Variant>.der | CERT.SM\_DS.TLS certificate of the S\_SM-DS, based on NIST P-384 curve  |

1. : SM-DS TLS Certificates with invalid curve

In order to generate the different files, next commands must be performed using the previous values and following input files:

Command lines for the generation of the SK.SM\_DS.TLS and the corresponding PK.SM\_DS.TLS for NIST P-384 curve:

openssl ecparam –name secp384r1 –genkey –out SK\_S\_SM\_DS\_TLS\_NIST\_P384.pem

openssl ec –in SK\_S\_SM\_DS\_TLS\_NIST\_P384.pem –pubout –out PK\_S\_SM\_DS\_TLS\_NIST\_P384.pem

The CERT.SM\_DS.TLS are generated using the command lines described in section 3.4.x. for each variant

#### SM-DS TLS Certificate – Invalid Certificate Policy

In order to generate the different files, the commands in section 3.4.3 shall be used.

The input data and files for generation applies with no changes except for the value of <certificatePolicies> extension that shall be set as follows:

* For Variant O: critical, 2.23.146.1.2.1.4
* For Variant A, B and C: critical, 2.23.146.1.2.1.0.0.2.1

Hereafter the generated SM-DS keys and certificates for TLS as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DS\_TLS\_NIST.pem | NIST P-256 Private key of the SM-DS for securing TLS connection |
| PK\_S\_SM\_DS\_TLS\_NIST.pem | NIST P-256 Public Key of the SM-DS (part of the CERT\_S\_SM\_DS\_TLS\_NIST.der) |
| CERT\_S\_SM\_DS\_TLS\_INV\_CERT\_POL\_<Variant>.der | CERT.DS.TLS certificate of the S\_SM-DS with invalid 'Certificate Policies' extension (OID set to 'id-rspRole-dp-auth'), formatted as X.509 certificate. |

1. : SM-DS TLS Certificates with invalid 'certificate policies'

#### SM-DS TLS Certificate – Missing Critical Extension

In order to generate the different files, the commands in section 3.4.3 shall be used.

The input data and files for generation applies with no changes except for the value of <extendedKeyUsage> extension that shall be set as follows:

* Absent

Hereafter the generated SM-DS keys and certificates for TLS as defined in Annex A.

| File name | Description |
| --- | --- |
| SK\_S\_SM\_DS\_TLS\_NIST.pem | NIST P-256 Private key of the SM-DS for securing TLS connection |
| PK\_S\_SM\_DS\_TLS\_NIST.pem | NIST P-256 Public Key of the SM-DS (part of the CERT\_S\_SM\_DS\_TLS\_NIST.der) |
| CERT\_S\_SM\_DS\_TLS\_INV\_CRITICAL\_EXT\_<Variant>.der | CERT.DS.TLS certificate of the S\_SM-DS with one of the critical extensions not present, formatted as X.509 certificate. |

1. : SM-DS TLS Certificate missing critical extension

#### SM-DS TLS Certificate – Invalid Extended Key Usage

In order to generate the different files, the commands in section 3.4.3 shall be used.

The input data and files for generation applies with no changes except for the value of <extendedKeyUsage> extension that shall be set as follows:

* critical, clientAuth

Hereafter the generated SM-DS keys and certificates for TLS as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DS\_TLS\_NIST.pem | NIST P-256 Private key of the SM-DS for securing TLS connection |
| PK\_S\_SM\_DS\_TLS\_NIST.pem | NIST P-256 Public Key of the SM-DS (part of the CERT\_S\_SM\_DS\_TLS\_NIST.der) |
| CERT\_S\_SM\_DS\_TLS\_INV\_EXT\_KEY\_USAGE\_<Variant>.der | CERT.SM\_DS.TLS certificate of the S\_SM-DS with invalid 'extended key usage' extension (not set to 'id-kp-serverAuth'), formatted as X.509 certificate. |

1. : SM-DS TLS Certificate with invalid 'extended key usage'

#### SM-DS TLS Certificate – Invalid Key Usage

In order to generate the different files, the commands in section 3.4.3 shall be used.

The input data and files for generation applies with no changes except for the value of <keyUsage> extension that shall be set as follows:

* critical, keyAgreement

Hereafter the generated SM-DS keys and certificates for TLS as defined in Annex A.

|  |  |
| --- | --- |
| File name | Description |
| SK\_S\_SM\_DS\_TLS\_NIST.pem | NIST P-256 Private key of the SM-DS for securing TLS connection |
| PK\_S\_SM\_DS\_TLS\_NIST.pem | NIST P-256 Public Key of the SM-DS (part of the CERT\_S\_SM\_DS\_TLS\_NIST.der) |
| CERT\_S\_SM\_DS\_TLS\_INV\_KEY\_USAGE\_<Variant>.der | CERT.DS.TLS certificate of the S\_SM-DS with invalid 'key usage' extension (not set to 'digitalSignature'), formatted as X.509 certificate. |

1. : SM-DS TLS Certificate with invalid 'key usage'

#### SM-DS TLS Certificate – Expired Certificate

In order to generate the different files, the commands in section 3.4.3 shall be used.

The input data and files for generation applies with no changes except for the value of <Validity> field that shall be set as follows:

* 1 day

Hereafter the generated SM-DS keys and certificates for TLS as defined in Annex A.

|  |  |
| --- | --- |
| **File name** | **Description** |
| SK\_S\_SM\_DS\_TLS\_NIST.pem | NIST P-256 Private key of the SM-DS for securing TLS connection |
| PK\_S\_SM\_DS\_TLS\_NIST.pem | NIST P-256 Public Key of the SM-DS (part of the CERT\_S\_SM\_DS\_TLS\_NIST.der) |
| CERT\_S\_SM\_DS\_TLS\_EXPIRED\_<Variant>.der | Expired CERT.DS.TLS certificate of the S\_SM-DS with a valid signature, correctly formatted as X.509 certificate. |

1. : SM-DS TLS keys and expired Certificate

1. RSP Certificates and Keys Files (Normative)

All certificates, keys and configuration files are provided within the SGP.26\_v3.x-YYYY\_Files.ZIP package which accompanies the present document. The latest published version of the ZIP package SHALL be used.

NOTE:

* “x” means the minor version of the present document.
* “YYYY” means the year when the file is updated.
1. Alternative to Certificate Generation

Additionally to the command described in section 2.4, the certificates can be generated using the next command:

openssl ca -batch -config <config\_file> -in <csr\_file\_name> -extensions <ext\_section\_name> -cert <ca\_cert\_file\_name> -keyfile <ca\_sk\_file\_name> -notext -out <cert\_pem\_file\_name> -startdate <validity\_start\_date> -enddate <validity\_end\_date>

Preconditions:

* Following entries are present in the indicated <config\_file> under the default CA section:

       ...

       database      = $ENV::OPENSSL\_HOME/indexXXCert.txt

       serial            = $ENV::OPENSSL\_HOME/serialXXCert

       ...

* Following files are present in OpenSSL home folder and are empty:
	+ - * indexXXCert.txt
			* indexXXCert.txt.attr
* The text file 'serialTlsCert' is present  in OpenSSL home folder and contains the desired serial number as hex string.
* Following extension to be referenced by <ext\_section\_name> sections are present in the indicated <config\_file> forthe appropriate:

               [ extensions]

               keyUsage

               extendedKeyUsage

               certificatePolicies

               subjectKeyIdentifier

               authorityKeyIdentifier

               subjectAltName

              crlDistributionPoints

* <validity\_start\_date> and <validity\_end\_date> are formatted YYMMDDHHMMSSZ, e.g. '170301154500Z' for 'Mar 1 15:45:00 2017 GMT'.
1. Generation of self-signed Test CI Certificates

This section describes the mechanism whereby RSP actors (e.g. SM-DP+ providers, eUICC Manufacturers) can generate and share their own self-signed Root Test CI Certificate (CERT.CI.SIG) with eSIM Device testers and SM-DP+ providers to enable the easy and repeatable download of the Test Profile described in [TS.48 reference] or any other non-operational test profile from a Test SM-DP+ (in other word a Staging SM-DP+ Platform) onto a Test eUICC.

The RSP actor generates the key pair and the self-signed Test CI Certificate (using the relevant SK.CI.SIG) as described in clause 3.1 of the present document.

Alternately, the RSP actor may use a key pair whose private key value is one of the private keys values specified in section 3.1.2.

The private key would be used to sign:

* The Test CERT.CISubCA.SIG for Variants B and C,
* The Test CERT.SM\_DPauth.SIG and Test CERT.SM\_DPpb.SIG to be provisioned onto a Test SM-DP+ platform,
* The Test CERT.SM\_DPSubCA.SIG to be provisioned onto a Test SM-DP+ platform for Variants A and C,
* The Test CERT.SM\_DP.TLS to be provisioned onto a Test SM-DP+ platform,
* The Test CERT.EUM.SIG and CERT.EUICC.SIG certificates to be provisioned onto the Test eUICCs,
* The CERT.EUMSubCA.SIG for Variants A and C.

The below table comprises the recommended minimum certificate definitions for a self-signed certificate. The cells marked “vendor-specific” in the “Value” column can be personalised by the RSP Actor:

| **Field** | **Value** |
| --- | --- |
| version | V3(2) as defined in RFC5280   |
| serialNumber | **Vendor-specific** |
| signature | sha256ECDSA  |
| Issuer | See 'subject' |
| Validity | **Vendor-specific** |
| Subject | **Vendor-specific** |
| Extension |  |
| subjectKeyIdentifier extension | NIST:**Vendor-specific**Brainpool:**Vendor-specific** |
| keyUsage Extension | Certificate Signing, CRL Signing (06) |
| certificatePolicies Extension | '2.23.146.1.2.1.0' (id-rspRole-ci) |
| basicConstraints Extension  | CA = true |
| subjectAltName Extension | **Vendor-specific** |
| crlDistributionPoints Extension | **Vendor-specific** |

**Table 64: Self-Signed CERT.CI.SIG**

The RSP actor may then publish the self-signed test CI as described in Annex D

1. Process to submit support of Test CI Certificates

GSMA maintains a page <https://www.gsma.com/esim/gsma-root-ci/> which publishes:

* A list of providers which support the test root certificate operated by GSMA CI, along with a list of the services they support using the test root certificate issuer
* A list of alternate self-signed root test certificate issuers, along with SM-DP+ servers that support them.

To enable public access of their test SM-DP+ to the broader eSIM test community, the RSP actor provider may submit the following items defined in D.1 and/or D.2 (using the Test Certificate Submission Form) to the e-mail testCICertificates@gsma.com.

Once submitted, the information will be published on <https://www.gsma.com/esim/gsma-root-ci/>

**D.1 List of RSP actors supporting test certificates signed by a test root certificate operated by GSMA CI**

A GSMA CI, in addition to GSMA CI RootCA certificates, may operate test root certificates and key pairs, used to sign test certificates which allow to perform interoperability testing (see Note 1).

NOTE 1 The test certificates defined above will not be recognized and accepted by a production system that trusts only live GSMA CI Root CAs

* Company name
* Confirmation of support of Test Profile as defined in SGP.22 [1]
* List (see Note 2) of test root certificates operated by any GSMA CI(s) that the provider uses as an EUM
* List (see Note 2) of the test root certificate(s) operated by any GSMA CI(s) that the provider uses as an SM-DP+ provider
* List (see Note 2) of the test root certificate(s) operated by any GSMA CI(s) that the provider uses as an SM-DS provider
* The URL to an application that enables the tester to trigger the release of a profile by the SM-DP+, to allow the download of the test profile using at least one of the options defined by SGP.22 [1].

NOTE 2 Each test root certificate in the list is uniquely identified by its Subject Key Identifier as defined in RFC 5280 [3]

**D.2 List of RSP Actor-specific self-signed root test certificate issuers**

* Company Name
* Confirmation of support of Test Profile as defined in SGP.22 [1]
* Confirmation of support of the self-signed root test CI(s) by the Test SM-DP+,
* The URL(see Note) hosting their test root CI Certificate (.pem file format) generated by following the instructions defined in clause 2.3 and 3.1 of the present document,
* Optionally, the URL (see Note) of the associated test CI private key generated by following the instructions defined in clause 2.3 and 3.1 of the present document,
* Optionally, the URL (see Note) of the signed client test EUM certificate and signed Test SM-DP+ server certificates,
* The URL to an application that enables the tester to trigger the release of a profile by the SM-DP+, to allow the download of the test profile using at least one of the options defined by SGP.22 [1].
* Once submitted, the information will be published <https://www.gsma.com/esim/gsma-root-ci/> with a date of publication and a date of expiry of the certificate.  Any renewal or change needs to be submitted using the process above.

NOTE: The test RSP Actor shall publicly host the files and the application necessary for testing.

1. Constants/Variables

This section contains the constant used to generate the certificates.

E.1 “Serial number” field

| **Certificate Type** | **Value** |
| --- | --- |
| CI RootCA | 0x00B874F3ABFA6C44D3 |
| EUM CA | 0x12345678 |
| eUICC  | 0x020000000000000001 |
| SM-DP+ Authentication | 0x100 (1st), 0x200 (2nd) |
| SM-DP+ Profile Binding | 0x101 (1st), 0x201 (2nd) |
| SM-DP+ TLS | 0x9 (1st), 0x99 (2nd), 0x994 (3rd), 0x998 (4th) |
| SM-DS Authentication | 0x7495 |
| SM-DS TLS | 0x1223334444 (1st), 0x122333444455555 (2nd) |
| eIM Signing | 0x03FF0AFF0009990101FF01 |
| eIM TLS/DTLS | 0x03FF0AFF0009990100FF00FF01 |

1. : Variant O Values

| **Certificate Type** | **Value** |
| --- | --- |
| CI RootCA | 0x000 |
| CI SubCA | 0x020 + <certificate number>i.e. 0x021, 0x022… |
| EUM CA | 0x100 + <EUM number> |
| EUM SubCA | 0x120 + <EUM SubCA number> |
| eUICC  | EID |
| SM-DP+ SubCA | 0x220 + <SM-DP+ SubCA number> |
| SM-DP+ Authentication | 0x230 + <SM-DP+ Auth server number> |
| SM-DP+ Profile Binding | 0x240 + <SM-DP+ Pb server number> |
| SM-DP+ TLS | 0x250 + <SM-DP+ TLS server number> |
| SM-DS SubCA | 0x320 + <SM-DP+ SubCA number> |
| SM-DS Authentication | 0x330 + <SM-DS Auth server number> |
| SM-DS TLS | 0x350 + <SM-DP+ TLS server number> |

1. : Variant A, B, C and Ov3 Values (where applicable)

E.2 “SubjectAltName” field

| **Certificate type** | **Value** |
| --- | --- |
| CI RootCA | RID:2.999.1 |
| EUM CA | RID:2.999.5 |
| SM-DP+ Authentication,SM-DP+ Profile Binding | RID:2.999.15 (1st), RID:2.999.12 (2nd)… |
| SM-DP+ TLS | 1st Server: OID = RID:2.999.10 DNS = testsmdpplus1example.com 2nd Server: OID = RID:2.999.12 DNS = testsmdpplus2example.com3rd Server: OID = RID:2.999.14 DNS = testsmdpplus4example.com4th Server: OID = RID:2.999.18 DNS = testsmdpplus8example.com |
| SM-DS Authentication | OID = RID:2.999.15 |
| SM-DS TLS | 1st Server: OID = RID:2.999.15 DNS = testrootsmds.example.com2nd Server: OID = RID:2.999.15.2 DNS = testsmds1example.com |
| eIM TLS/DTLS | OID = RID:2.999.20DNS = eim.example.com |

1. : Variant O Values

| **Certificate Type** | **Value** |
| --- | --- |
| CI RootCA / CI SubCA | RID:2.999.1 |
| EUM CA | RID:2.999.101, RID:2.999.102… |
| SM-DP+ SubCA | RID:2.999.221, RID:2.999.222… |
| SM-DP+ AuthenticationSM-DP+ Profile Binding | RID:2.999.231, RID:2.999.232… |
| SM-DP+ TLS | OID = RID:2.999.251, RID:2.999.252…DNS = testsmdpplus<Server number>example.com |
| SM-DS SubCA | RID:2.999.321, RID:2.999.322… |
| SM-DS Authentication | RID:2.999.331, RID:2.999.332… |
| SM-DS TLS | OID = RID:2.999.251, RID:2.999.252…DNS = testsmds<Server number>example.com |

1. : Variant A, B, C and Ov3 Values (where applicable)

E.3 Variables

| **Constant Name** | **Value** |
| --- | --- |
| <curve> | NIST, BRP |
| <variant> | Var{O, Ov3,A,B,C} |
| <number> | Int{null, 2, 4, 8, ...} |
| <permitedEins> | 89049032 |

1. : Constant Names
2. Templates for generating certificates

|  |  |
| --- | --- |
| Files | Content |
| CI-csr.cnf | Prompt = nodistinguished\_name = dn-paramextensions = extend[extend]subjectKeyIdentifier= *<Value of “*subjectKeyIdentifier*” field>*keyUsage= *<Value of “keyUsage” field>*certificatePolicies= *<Value of “certificatePolicies” field>*basicConstraints= *<Value of “*basicConstraints*” field>*subjectAltName= *<Value of “subjectAltName” field>*crlDistributionPoints= *<Value of “crlDistributionPoints” field>* |
| CISubCA-csr.cnf | prompt = nodistinguished\_name = dn-param*<Value of “subject” field>* |
| CISubCA-ext.cnf | prompt = noextensions = extend[extend]subjectKeyIdentifier= *<Value of “*subjectKeyIdentifier*” field>*keyUsage= *<Value of “keyUsage” field>*certificatePolicies= *<Value of “certificatePolicies” field>*basicConstraints= *<Value of “*basicConstraints*” field>*subjectAltName= *<Value of “subjectAltName” field>*crlDistributionPoints= *<Value of “crlDistributionPoints” field>* |
| EUM-csr.cnf | prompt = nodistinguished\_name = dn-param*<Value of “subject” field>* |
| EUM-ext.cnf | prompt = noextensions = extend[extend]subjectKeyIdentifier= *<Value of “*subjectKeyIdentifier*” field>*authorityKeyIdentifier= *<Value of “*authorityKeyIdentifier*” field>*basicConstraints= *<Value of “*basicConstraints*” field>*certificatePolicies= *<Value of “certificatePolicies” field>*keyUsage= *<Value of “keyUsage” field>*subjectAltName= *<Value of “subjectAltName” field>*crlDistributionPoints= *<Value of “crlDistributionPoints” field>**nameConstraints=<Value of “nameConstraints” field>**# GSMA-specific extension for permittedEins*2.23.146.1.2.2.0 = ASN1:SEQUENCE*:permittedEins*[permittedEins]ein1 = <Value of “permittedEins”> |
| EUMSubCA-csr.cnf | prompt = nodistinguished\_name = dn-param*<Value of “subject” field>* |
| EUMSubCA-ext.cnf | prompt = noextensions = extend[extend]subjectKeyIdentifier= <Value of “subjectKeyIdentifier” field>authorityKeyIdentifier= *<Value of “*authorityKeyIdentifier*” field>*basicConstraints= *<Value of “*basicConstraints*” field>*certificatePolicies= *<Value of “certificatePolicies” field>*keyUsage= *<Value of “keyUsage” field>*subjectAltName= *<Value of “subjectAltName” field>*crlDistributionPoints= *<Value of “crlDistributionPoints” field>* |
| EUICC-csr.cnf | prompt = nodistinguished\_name = dn-param*<Value of “subject” field>* |
| EUICC-ext.cnf | Prompt=noExtensions=extend[extend]subjectKeyIdentifier= <Value of “subjectKeyIdentifier” field>authorityKeyIdentifier= <Value of “authorityKeyIdentifier” field>certificatePolicies= <Value of “certificatePolicies” field>keyUsage= <Value of “keyUsage” field> |
| SM\_DPSubCA-csr.cnf | prompt = nodistinguished\_name = dn-param*<Value of “subject” field>* |
| SM\_DPSubCA-ext.cnf | Prompt = noExtensions=extend[extend]subjectKeyIdentifier= <Value of “subjectKeyIdentifier” field>basicConstraints= <Value of “basicConstraints” field>certificatePolicies= <Value of “certificatePolicies” field>keyUsage= <Value of “keyUsage” field>subjectAltName= <Value of “subjectAltName” field>crlDistributionPoints= <Value of “crlDistributionPoints” field> |
| SM\_DSSubCA-csr.cnf | prompt = nodistinguished\_name = dn-param*<Value of “subject” field>* |
| SM\_DSSubCA-ext.cnf | Prompt = noExtensions=extend[extend]subjectKeyIdentifier= <Value of “subjectKeyIdentifier” field>basicConstraints= <Value of “basicConstraints” field>certificatePolicies= <Value of “certificatePolicies” field>keyUsage= <Value of “keyUsage” field>subjectAltName= <Value of “subjectAltName” field>crlDistributionPoints= <Value of “crlDistributionPoints” field> |
| SM\_DP-csr.cnf | prompt = nodistinguished\_name = dn-param*<Value of “subject” field>* |
| SM\_DP-ext.cnf | prompt = noextensions = extend[extend]subjectKeyIdentifier= *<Value of “*subjectKeyIdentifier*” field>*authorityKeyIdentifier= *<Value of “*authorityKeyIdentifier*” field>*basicConstraints= *<Value of “*basicConstraints*” field>*certificatePolicies= *<Value of “certificatePolicies” field>*keyUsage= *<Value of “keyUsage” field>*subjectAltName= *<Value of “subjectAltName” field>*crlDistributionPoints= *<Value of “crlDistributionPoints” field>* |
| SM\_DS-csr.cnf | prompt = nodistinguished\_name = dn-param*<Value of “subject” field>* |
| SM\_DS-ext.cnf | prompt = noextensions = extend[extend]subjectKeyIdentifier= *<Value of “*subjectKeyIdentifier*” field>*authorityKeyIdentifier= *<Value of “*authorityKeyIdentifier*” field>*basicConstraints= *<Value of “*basicConstraints*” field>*certificatePolicies= *<Value of “certificatePolicies” field>*keyUsage= *<Value of “keyUsage” field>*subjectAltName= *<Value of “subjectAltName” field>*crlDistributionPoints= *<Value of “crlDistributionPoints” field>* |
| EIM\_csr.cnf | prompt = nodistinguished\_name = req\_distinguished\_name[ req\_distinguished\_name ] *<Value of “subject” field>* |
| EIM\_ext.cnf | prompt = noextensions = extend[extend]authorityKeyIdentifier= <Value of “authorityKeyIdentifier” field>keyUsage= *<Value of “keyUsage” field>* |
| EIMtls-csr.cnf | prompt = nodistinguished\_name = req\_distinguished\_name[ req\_distinguished\_name ] *<Value of “subject” field>* |
| EIMtls-ext.cnf | prompt = noextensions = extend[extend]subjectKeyIdentifier= *<Value of “*subjectKeyIdentifier*” field>*authorityKeyIdentifier= *<Value of “*authorityKeyIdentifier*” field>*keyUsage= *<Value of “keyUsage” field>*extendedKeyUsage*= <Value of “extendedKeyUsage” field>*subjectAltName= *<Value of “subjectAltName” field>*crlDistributionPoints= *<Value of “crlDistributionPoints” field>* |

1. Document Management
	1. Document History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Date | CR | Brief Description of Change | Approval Authority | Editor / Company |
| v1.0 | 9 June 2017 |  | New PRD Publication | PSMC | Yolanda SanzGSMA |
| V1.1 | 28 Sept 2017 |  | The first minor version of SGP.26  | RSPPLEN | Yolanda SanzGSMA |
| V1.2 | 3th January |  | The second minor version of SGP.26 | RSPPLEN | Yolanda Sanz GSMA |
| V1.3 | 07 July 2020 |  | The third version of SGP.26 | eSIMG | Yolanda, Sanz GSMA |
| V1.4 | 31 July 2020 |  | The fourth version of SGP.26 | ISAG | Yolanda Sanz, GSMA |
| V1.5 | 30 June 2021 | CR0021R01 | Validity period of TLS Certificates | ISAG | Alejandro Pulido, VALID |
| V3.0 | 12 June 2023 | CR3001R01 | CI SubCA introduction for Variants B and C | eSIMWG3 | Alejandro Pulido, VALID |
| V3.0 | 12 June 2023 | CR3002R01 | EUM SubCA introduction for Variants A and C | eSIMWG3 | Alejandro Pulido, VALID |
| V3.0 | 12 June 2023 | CR3003R01 | Section 1 fixes | eSIMWG3 | Alejandro Pulido, VALID |
| V3.0 | 12 June 2023 | CR3004R01 | Annexes fixes | eSIMWG3 | Alejandro Pulido, VALID |
| V3.0 | 12 June 2023 | CR3005R02 | Annexes E – Contant | eSIMWG3 | Alejandro Pulido, VALID |
| V3.0 | 12 June 2023 | CR3007R01 | eUICC Certificate | eSIMWG3 | Alejandro Pulido, VALID |
| V3.0 | 12 June 2023 | CR3008R02 | DP TLS | eSIMWG3 | Alejandro Pulido, VALID |
| V3.0 | 12 June 2023 | CR3009R01 | DS TLS | eSIMWG3 | Alejandro Pulido, VALID |
| V3.0 | 12 June 2023 | CR3010R01 | SM-DP updates | eSIMWG3 | Alejandro Pulido, VALID |
| V3.0 | 12 June 2023 | CR3011R01 | DS Auth | eSIMWG3 | Javier Hernández, VALID |
| V3.0 | 27 September 2023 | CR3012R01 | Invalid Test Cases DP | eSIMWG3 | Javier Hernández, VALID |
| V3.0 | 27 September 2023 | CR3013R02 | Invalid Test Cases DS | eSIMWG3 | Javier Hernández, VALID |
| V3.0 | 27 September 2023 | CR3014R01 | Full alignment in subsections | eSIMWG3 | Javier Hernández, VALID |
| V3.0 | 27 September 2023 | CR3015R02 | eIM Keys and Certificates | eSIMWG3 | Javier Hernández, VALID |
| V3.0 | 11 December 2023 | CR3016R00 | Invalid Signatures | eSIMWG3 | Javier Hernández, VALID |
| V3.0 | 11 December 2023 | CR3017R00 | NIST P384 | eSIMWG3 | Javier Hernández, VALID |
| V3.0 | 11 December 2023 | CR3018R02 | eIM TLS DTLS Keys and Certificates | eSIMWG3 | Javier Hernández, VALID |
| V3.0.1 | 26 January 2024 | CR301000R00 | Filename Aligment | ISAG | Javier Hernández, VALID |
| CR301001R00 | Fix PermittedEINs |
| CR301002R00 | eIM TLS DTLS Certificate format |

**Other Information**

|  |  |
| --- | --- |
| **Type** | **Description** |
| Document Owner | eSIMG |
| Editor / Company | Javier Hernández / VALID |

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com

Your comments or suggestions & questions are always welcome.