RSP Test Certificates Definition for IoT and Consumer
Version 3.0.1
26 January 2024

Security Classification: Non-confidential

Copyright Notice

Disclaimer

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

1.1 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.

1.2 References

Ref	Document Number	Title
[1]	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

2 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.

2.1 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

2.2 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

2.3 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

2.4 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

2.5 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.

3 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.

3.1 Certificate Issuer

3.1.1 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 CI ou = TESTCERT o = RSPTEST c = 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

Field	Value
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

Table 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>.der CERT_CI_SIG_<curve>.pem	Certificate of the CI

Table 2: 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
```

3.1.2 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

Field	Value
Subject	cn = Test CI SubCA ou = TESTCERT o = RSPTEST c = 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

Table 3: 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>.der CERT_CISubCA_SIG_<curve>.pem	Certificate of the CI SubCA

Table 4: 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
```

3.1 eUICC

3.1.1 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 eUICC serialNumber = for Variant O '89049032123451234512345678901235' (EID) serialNumber = for Variant Ov3, A, B, C - See Annex E.1 o = Same as EUM - See 3.3.1 c = ES
Extension	
authorityKeyIdentifier	keyid, issuer
subjectKeyIdentifier	hash
keyUsage Extension	Critical digitalSignature

certificatePolicies	Critical For 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

Table 5: 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

Table 6: 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
```

3.2 EUM

3.2.1 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 EUM c = ES
Extensions	
authorityKeyIdentifier	issuer, keyid
subjectKeyIdentifier	hash
keyUsage	critical, keyCertSign
certificate Policies	Critical For O: 2.23.146.1.2.1.2 For Ov3, A, B, C: 2.23.146.1.2.1.0.0.0
subjectAltName	See Annex E.2

Field	Value
basicConstraints	For Variants O, Ov3 and B Critical CA = true pathLenConstraint = 0 For Variants A and C Critical CA = true pathLenConstraint = 1
crlDistributionPoints	[1]CRL Distribution Point Distribution Point Name: Full Name: URL=http://ci.test.example.com/CRL-B.crl
nameConstraints	Critical For O: permittedSubtrees: id-at-organizationName: '2.5.4.10' organization name: "RSP Test EUM" UTF8String id-at-serialNumber: '2.5.4.5' iin: "89049032" PrintableString For Ov3, A, B, C permittedSubtrees: id-at-organizationName: '2.5.4.10' organization name: "RSP Test EUM" UTF8String
permittedEIN	For Variants Ov3, A, B, C permitedEins: See Annex E.3

Table 7: 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

Table 8: 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
```

3.2.2 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)

Field	Value
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	Critical CA = true pathLenConstraint = 0
crlDistributionPoints	Same as 3.3.1

Table 9: 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

Table 10: 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
```

3.3 SM-DP+

3.3.1 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_DPSubCA ou = TESTCERT o = RSPTEST c = 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:true pathLenConstraint = 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>

Table 11: 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

File name	Description
CERT_SM_DPSubCA_<variant> _SIG_<curve>.der	Certificate of the SM-DP+ SubCA

Table 12: 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
```

3.3.2 SM-DP+ Certificate for Authentication

3.3.2.1 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 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, 2.23.146.1.2.1.4 For 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>

Table 13: 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+

Table 14: 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
```

3.3.3 SM-DP+ Certficate for Profile Biding

3.3.3.1 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, 2.23.146.1.2.1.5 For 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>

Table 15: 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+

Table 16: 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
```

3.3.4 TLS

3.3.4.1 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	Critical digitalSignature
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>

Table 17: 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

Table 18: 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
```

3.4 SM-DS

3.4.1 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_DSSubCA ou = TESTCERT o = RSPTEST c = ES
Extensions	
subjectKeyIdentifier	hash
authorityKeyIdentifier	keyid, issuer
keyUsage	critical, keyCertSign, cRLSign
certificatePolicies	2.23.146.1.2.1.0.0.2
basicConstraints	critical, cA:true pathLenConstraint = 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>

Table 19: 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

Table 20: 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
```

3.4.2 SM-DSauth

3.4.2.1 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 of CERT.CISubCA.SIG “subject” field For Variant A and C: <Value of CERT.SM_DSSubCA.SIG “subject” field
Validity	1095 days (3 years)
Subject	o = ‘ACME’ cn = ‘TEST SM-DS <SM-DS number>’
Extensions	
authorityKeyIdentifier	27eyed,issuer
subjectKeyIdentifier	hash
keyUsage	critical, digitalSignature
certificatePolicies	For Variant O: critical, 2.23.146.1.2.1.7 For 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>

Table 21: 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

Table 22: 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
```

3.4.3 TLS

3.4.3.1 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	Critical digitalSignature
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>

Table 23: 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

Table 24: 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
```

3.5 eIM

3.5.1 eIM Certificate for Signing

3.5.1.1 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.com C=DE
Extensions	
authorityKeyIdentifier	keyid, issuer
keyUsage	Critical, digitalSignature

Table 25: 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
```

3.5.1.2 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

Table 26: Signing Keys and Certificates of eIM

3.5.1.3 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.

3.5.2 TLS/DTLS

3.5.2.1 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

Field	Value
subject	C = 'DE' CN = 'eim.example.com'
Extensions	
authorityKeyIdentifier	keyId, issuer
keyUsage	critical digitalSignature
extendedKeyUsage	critical, serverAuth, clientAuth
subjectAltName	See Annex E.2
crlDistributionPoints	<Value of CERT.CI.SIG."crlDistributionPoints" field>

Table 27: 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

Table 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
```

4 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.

4.1 SM-DP+

4.1.1 SM-DP+ Certificate for Authentication

4.1.1.1 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

Table 2: 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)

4.1.1.2 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].

4.1.1.2.1 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

File name	Description
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

Table 3: 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.

4.1.2 SM-DP+ Certificate for Profile Binding

4.1.2.1 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

Table 4: 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)

4.1.2.1.1

4.1.2.2 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.

4.1.2.2.1

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

Table 5: 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.

4.1.3 SM-DP+ TLS Certificate

4.1.3.1 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

Table 6: 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)

4.1.3.2 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.

4.1.3.2.1

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

Table 7: 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.

4.1.3.3 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

4.1.3.3.1

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.

4.1.3.4 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

4.1.3.4.1

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_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.

Table 8: DP_TLS Keys and Certificates with critical extension not present

4.1.3.5 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

4.1.3.5.1

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_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.

Table 9: DP+ TLS Certificates with invalid 'extended key usage'

4.1.3.6 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

4.1.3.6.1

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_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.

Table 10: DP+ TLS Keys and Certificates with invalid 'key usage' extension

4.1.3.7 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

4.1.3.7.1

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_EXPIRED_<Variant>.der	Expired CERT.SM_DP.TLS certificate of the S_SM-DP+ with a valid signature, correctly formatted as X.509 certificate.

Table 11: DP+ TLS Keys and expired Certificates

4.2 SM-DS

4.2.1 SM-DS Certificate for Authentication

4.2.1.1 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

Table 12: 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)

4.2.1.2 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

Table 13: 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

4.2.2 SM-DS TLS Certificate

4.2.2.1 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

Table 14: 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)

4.2.2.2 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.

4.2.2.2.1

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

Table 15: 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

4.2.2.3 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.

Table 16: SM-DS TLS Certificates with invalid 'certificate policies'

4.2.2.4 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.

Table 17: SM-DS TLS Certificate missing critical extension

4.2.2.5 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.

Table 18: SM-DS TLS Certificate with invalid 'extended key usage'

4.2.2.6 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.

Table 19: SM-DS TLS Certificate with invalid 'key usage'

4.2.2.7 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.

Table 20: SM-DS TLS keys and expired Certificate

Annex A 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.

Annex B 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'.

Annex C 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**

Field	Value
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

Annex D 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.

Annex E 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

Table 21: 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>

Table 22: 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.com 3rd Server: OID = RID:2.999.14 DNS = testsmdpplus4example.com 4th 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.com 2nd Server: OID = RID:2.999.15.2 DNS = testsmds1example.com
eIM TLS/DTLS	OID = RID:2.999.20 DNS = eim.example.com

Table 24: 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+ Authentication SM-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

Table 25: Variant A, B, C and Ov3 Values (where applicable)

Table 26

E.3 Variables

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

Table 27: Constant Names

Annex F Templates for generating certificates

Files	Content
CI-csr.cnf	Prompt = no distinguished_name = dn-param extensions = 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 = no distinguished_name = dn-param <*Value of "subject" field*>
CISubCA-ext.cnf	prompt = no extensions = 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 = no distinguished_name = dn-param <*Value of "subject" field*>

EUM-ext.cnf	prompt = no extensions = 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 = no distinguished_name = dn-param <*Value of "subject" field*>
EUMSubCA-ext.cnf	prompt = no extensions = 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 = no distinguished_name = dn-param <*Value of "subject" field*>
EUICC-ext.cnf	Prompt=no Extensions=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 = no distinguished_name = dn-param <*Value of "subject" field*>
SM_DPSubCA-ext.cnf	Prompt = no Extensions=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 = no distinguished_name = dn-param <*Value of "subject" field*>
SM_DSSubCA-ext.cnf	Prompt = no Extensions=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 = no distinguished_name = dn-param <*Value of "subject" field*>
SM_DP-ext.cnf	prompt = no extensions = 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 = no distinguished_name = dn-param <*Value of "subject" field*>
SM_DS-ext.cnf	prompt = no extensions = 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 = no distinguished_name = req_distinguished_name [req_distinguished_name] <*Value of "subject" field*>
EIM_ext.cnf	prompt = no extensions = extend [extend] authorityKeyIdentifier= <Value of "authorityKeyIdentifier" field> keyUsage= <*Value of "keyUsage" field*>
EIMtls-csr.cnf	prompt = no distinguished_name = req_distinguished_name [req_distinguished_name] <*Value of "subject" field*>
EIMtls-ext.cnf	prompt = no extensions = 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*>

Annex G Document Management

G.1 Document History

Version	Date	CR	Brief Description of Change	Approval Authority	Editor / Company
v1.0	9 June 2017		New PRD Publication	PSMC	Yolanda Sanz GSMA
V1.1	28 Sept 2017		The first minor version of SGP.26	RSPPLEN	Yolanda Sanz GSMA
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.