SGP.26 RSP Test Certificates defintions

Version 1.4

31 July 2020

This Industry Specification is a Non-binding Permanent Reference Document of the GSMA

Security Classification: Non-confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is confidential to the Association and 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

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.

Antitrust Notice

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

This GSMA Permanent Reference Document (PRD) 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 GSMA AA.35 - Procedures for Industry Specifications..

Table of Contents

1 Introduction 4
1.1 Scope 4
1.2 References 4
2 Tool chain for generation of the keys and certificates 5
2.1 OpenSSL 5
2.2 Keys generation 5
2.3 CI Certificate Generation 6
2.4 Non-Root Certificate generation 6
2.5 Certificate display 8
3 Test Certificates and keys – Valid test cases 8
3.1 Certificate Issuer 8
3.1.1 CI Certificate: definition of data to be signed 8
3.1.2 CI Keys and Certificate 9
3.1.3 Input data for generation 9
3.2 eUICC 10
3.2.1 eUICC Certificate: definition of data to be signed 10
3.2.2 eUICC Keys and Certificate 10
3.2.3 Input data for generation 11
3.3 EUM 11
3.3.1 EUM Certificate: definition of data to be signed 11
3.3.2 EUM Keys and Certificate 12
3.3.3 Input data for generation 13
3.4 SM-DP+ 13
3.4.1 DPauth 13
3.4.2 DPpb 17
3.4.3 TLS 20
3.5 SM-DS 27
3.5.1 DSauth 27
3.5.2 TLS 28
4 Test Certificates and keys – Invalid test cases 31
4.1 eUICC 31
4.2 SM-DP+ 32
4.2.1 DPauth 32
4.2.2 DPpb 34
4.2.3 TLS 37
4.3 SM-DS 46
4.3.1 DSauth 46
4.3.2 TLS 49
Annex A RSP Certificates and Keys Files (Normative) 59
Annex B Alternative to Certificate Generation 60
Annex C Generation of self-signed Test CI Certificates 61
Annex D Process to submit support of Test CI Certificates 63

Annex E Document Management 65
E.1 Document History 65

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.ECDSA) per curve
- One EUM Certificate (CERT.EUM.ECDSA) per curve
- For each SM-DP+, two Certificates (CERT.DPauth.ECDSA and CERT.DPpb.ECDSA) per curve
- Two SM-DP+ TLS Certificate (CERT.DP.TLS) per curve
- One eUICC Certificate (CERT.EUICC.ECDSA) per curve
- One SM-DS Certificate (CERT.DSauth.ECDSA) per curve
- Two SM-DS TLS Certificate (CERT.DS.TLS) per curve

The certificates to be created for error cases are the following:

- Two SM-DP+ Certificates (CERT.DPauth.ECDSA and CERT.DPpb.ECDSA) per curve with invalid signature
- One SM-DS Certificate (CERT.DSauth.ECDSA) per curve with invalid signature
- Two SM-DP+ Certificates (CERT.DPauth.ECDSA and CERT.DPpb.ECDSA) with invalid curve
- One SM-DS Certificate (CERT.DSauth.ECDSA) with invalid curve

1.2 References

Ref	Document Number	Title
[1]	SGP.22	GSMA "RSP Technical specification" V2.2.2
[2]	SGP.23	GSMA "RSP Test Specification" v1.7
[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	2
serialNumber	'00 B8 74 F3 AB FA 6C 44 D3'
signature	sha256ECDSA
Issuer	See 'subject'
Validity	12783 days (35 years)

Subject	cn = Test CI
	ou = TESTCERT
	o = RSPTEST
	c = IT

subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters '1.2.840.10045.3.1.7' (prime256v1) or '1.3.36.3.3.2.8.1.1.7' (brainpoolP256r1) subjectPublicKey=[CI public key value]
Extension	(Sequence)
subjectKeyIdentifier extension	NIST: 'F5 41 72 BD F9 8A 95 D6 5C BE B8 8A 38 A1 C1 1D 80 0A 85 C3' Brainpool: 'C0 BC 70 BA 36 92 9D 43 B4 67 FF 57 57 05 30 E5 7A B8 FC D8'
keyUsage Extension	Certificate Signing, Off-line CRL Signing, CRL Signing (06)
certificatePolicies Extension	'2.23.146.1.2.1.0' (id-rspRole-ci)

Field	Value
basicConstraints Extension	CA = true
subjectAltName Extension	'2.999.1'
crlDistributionPoints Extension	[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.ECDSA

3.1.2 CI Keys and Certificate

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

File name	Description
SK_CI_ECDSA_NIST.pem	NIST P-256 Private Key of the CI
CERT_CI_ECDSA_NIST.der CERT_CI_ECDSA_NIST.pem	Certificate of the CI for its NIST P-256 Public Key in DER and PEM formats
SK_CI_ECDSA_BRP.pem	Brainpool P256r1 Private Key of the CI
CERT_CI_ECDSA_BRP.der CERT_CI_ECDSA_BRP.pem	Certificate of the CI for its Brainpool P256r1 Public Key in DER and PEM formats

Table 2: CI Keys and Certificates

3.1.3 Input data for generation

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

The CERT.CI.ECDSA is generated using the command lines described in section 2.3 with the following input data:

<ca_configuration_file>: `CI-csr.cnf` as defined in Annex A.

<serial> set with value defined in section 3.1.1 for `serialNumber` data field.

<days> set with value defined in section 3.1.1 for `validity` data field.

3.2 eUICC

3.2.1 eUICC Certificate: definition of data to be signed

Field	Value
Version	2
serialNumber	'02 00 00 00 00 00 00 00 01'
signature	sha256ECDSA

Issuer	cn = EUM Test
	o = RSP Test EUM
	c = ES

Validity	2000000 days

Subject	cn = Test eUICC
	serialNumber = '89049032123451234512345678901235' (EID)
	o = RSP Test EUM
	c = DE

subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters '1.2.840.10045.3.1.7' (prime256v1) or '1.3.36.3.3.2.8.1.1.7' (brainpoolP256r1) subjectPublicKey=[EUICC public key value] (see section 3.2.2)
Extension (Sequence)	
authorityKeyIdentifier Extension	<Value of CERT.EUM.ECDSA."subjectKeyIdentifier" field> for prime256v1 or brainpoolP256r1
subjectKeyIdentifier Extension	NIST: A5 24 76 AF 5D 50 AA 37 64 37 CC B1 DA 21 72 EF 45 F4 84 F0Brainpool: C8 A6 4F 34 3B 85 B7 B0 57 8D C5 7F 8F 13 58 6D C8 04 ED 84
keyUsage Extension	Critical digitalSignature ('80')
certificatePolicies Extension	Critical '2.23.146.1.2.1.1' (id-rspRole-euicc)

Table 3: CERT.EUICC.ECDSA

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.

3.2.2 eUICC Keys and Certificate

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

File name	Description
SK_EUICC_ECDSA_NIST.pem	NIST P-256 Private key of the eUICC for creating signatures
PK_EUICC_ECDSA_NIST.pem	NIST P-256 Public Key of the eUICC (part of the CERT_EUICC_ECDSA_NIST.der)
CERT_EUICC_ECDSA_NIST.der	Certificate of the eUICC for its NIST P-256 Public key
SK_EUICC_ECDSA_BRP.pem	Brainpool P256r1 Private key of the eUICC for creating signatures
PK_EUICC_ECDSA_BRP.pem	Brainpool P256r1 Public Key of the eUICC (part of the CERT_EUICC_ECDSA_BRP.der)
CERT_EUICC_ECDSA_BRP.der	Certificate of the eUICC for its Brainpool P256r1 Public key

Table 4: eUICC Keys and Certificates

3.2.3 Input data for generation

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

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

<input_csr_file_name>: `eUICC-csr.cnf` as defined in Annex A.

<ca_cert_file_name> and <ca_sk_file_name>: files generated in section 3.3.2 (file containing the CERT.EUM.ECDSA and SK.EUM.ECDSA respectively).

<serial> set with value defined in section 3.2.1 for `serialNumber` data field.

<days> set with value defined in section 3.2.1 for `validity` data field.

<cert_ext_file_name>: `eUICC-ext.cnf` as defined in Annex A.

3.3 EUM

3.3.1 EUM Certificate: definition of data to be signed

Field	Value
version	2
serialNumber	'12 34 56 78'
signature	algorithm = '1.2.840.10045.4.3.2' (sha256ECDSA)
Issuer	<Value of CERT.CI.ECDSA."subject" field>
validity	12410 days (34 years)

subject	cn = EUM Test
	o = RSP Test EUM
	c = ES

Field	Value
subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters= '1.2.840.10045.3.1.7' (prime256v1) or '1.3.36.3.3.2.8.1.1.7' (brainpoolP256r1) subjectPublicKey=[EUM public key value] (see section 3.3.2)
authorityKeyIdentifier Extension	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for prime256v1 or brainpoolP256r1

subjectKeyIdentifier Extension	NIST (prime256v1):
	DD:3D:A2:4D:35:0C:1C:C5:D0:AF:09:65:F4:0E:C3:4C:5E:E4:09:F1
	Brainpool (brainpoolP256r1):
	6F A1 E5 21 73 63 A8 22 BD ED 98 8A 1A 0D 0F F5 D7 62 0D B7

keyUsage Extension	Critical Certificate Sign ('04')
Certificate Policies	Critical '2.23.146.1.2.1.2' (id-rspRole-eum)
subjectAltName Extension	'2.999.5'

basicConstraints	Critical
	CA = true
	pathLenConstraint = 0

crlDistributionPoints Extension	[1]CRL Distribution Point Distribution Point Name: Full Name: URL=http://ci.test.example.com/CRL-B.crl
nameConstraints	Critical permittedSubtrees: id-at-organizationName: '2.5.4.10' organization name: "RSP Test EUM" UTF8String id-at-serialNumber: '2.5.4.5' iin: "89049032" PrintableString

Table 5: CERT.EUM.ECDSA

3.3.2 EUM Keys and Certificate

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

File name	Description
SK_EUM_ECDSA_NIST.pem	NIST P-256 Private key of the EUM for creating signatures
PK_EUM_ECDSA_NIST.pem	NIST P-256Public Key of the EUM (part of the CERT_EUM_ECDSA_NIST.der)
CERT_EUM_ECDSA_NIST.der	Certificate of the EUM for its Public NIST P-256 key

File name	Description
SK_EUM_ECDSA_BRP.pem	Brainpool P256r1 Private key of the EUM for creating signatures
PK_EUM_ECDSA_BRP.pem	Brainpool P256r1 Public Key of the EUM (part of the CERT_EUM_ECDSA_BRP.der)
CERT_EUM_ECDSA_BRP.der	Certificate of the EUM for its Public Brainpool P256r1 key

Table 6: EUM Keys and Certificates

3.3.3 Input data for generation

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

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

<input_csr_file_name>: `EUM-csr.cnf` as defined in Annex A.

<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 3.3.2 for `serialNumber` data field.

<days> set with value defined in section 3.3.2 for `validity` data field.

<cert_ext_file_name>: `EUM-ext.cnf` as defined in Annex A.

3.4 SM-DP+

3.4.1 DPauth

3.4.1.1 SM-DP+ n°1 Certificate for Authentication: definition of data to be signed

Field	Value
Version	'2'
serialNumber	'100'
signature	algorithm = '1.2.840.10045.4.3.2' (sha256ECDSA)
Issuer	<Value of CERT.CI.ECDSA."subject" field>
Validity	1095 days (3 years)

Subject	o = 'ACME'
	cn = 'TEST SM-DP+'

Field	Value
subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters= '1.2.840.10045.3.1.7' (prime256v1) or '1.3.36.3.3.2.8.1.1.7' (brainpoolP256r1) subjectPublicKey= corresponding <PK.DPauth.ECDSA value> (see 3.4.1.2)
Extensions	(Sequence)
Extension for authorityKeyIdentifier	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for prime256v1 or brainpoolP256r1
Extension for subjectKeyIdentifier	NIST: 'BD 5A 82 CC 1A 96 60 21 18 BA 75 60 A1 FF 83 A7 8B 21 0B E5' Brainpool: '79 A4 BD 4D 78 FF 47 34 BC 60 45 CF 91 96 24 4A 1F B8 4B EB'
Extension for keyUsage	Digital Signature ('80')
Extension for certificatePolicies	'2.23.146.1.2.1.4' (id-rspRole-dp-auth)
Extension for subjectAltName	'2.999.10'
Extension for crlDistributionPoints	<Value of CERT.CI.ECDSA."crlDistributionPoints" field>

Table 7: CERT.DPauth.ECDSA of SM-DP+ n°1

3.4.1.2 SM-DP+ n°1 Keys and Certificate

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

File name	Description
SK_S_SM_DPauth_ECDSA_NIST.pem	NIST P-256 Private Key of the SM-DP+ n°1 for creating signatures for SM-DP+ authentication
PK_S_SM_DPauth_ECDSA_NIST.pem	NIST P-256 Public Key of the SM-DP+ n°1 (part of the CERT_S_SM_DPauth_ECDSA_NIST.der)
CERT_S_SM_DPauth_ECDSA_NIST.der	Certificate of the SM-DP+ n°1for its Public NIST P-256 key used for SM-DP+ authentication
SK_S_SM_DPauth_ECDSA_BRP.pem	Brainpool P256r1 Private Key of the SM-DP+ n°1for creating signatures for SM-DP+ authentication
PK_S_SM_DPauth_ECDSA_BRP.pem	Brainpool P256r1 Public Key of the SM-DP+ n°1 (part of the CERT_S_SM_DPauth_ECDSA_BRP.der)

File name	Description
CERT_S_SM_DPauth_ECDSA_BRP.der	Certificate of the SM-DP+ n°1for its Public Brainpool P256r1 key used for SM-DP+ authentication

Table 8: DPAuth Keys and Certificates of SM-DP+ n°1

3.4.1.3 Input data for generation

The SK.DPauth.ECDSA and PK.DPauth.ECDSA of the SM-DP+ n°1 are generated using the command lines as described in section 2.2.

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

<input_csr_file_name>: `DP-csr.cnf` as defined in Annex A.

<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 3.4.1.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.1.1 for `validity` data field.

<cert_ext_file_name>: `DPauth-ext.cnf` as defined in Annex A.

3.4.1.4 SM-DP+ n°2 Certificate for Authentication: definition of data to be signed

Field	Value
Version	Same as in section 3.4.1.1
serialNumber	'200'
signature	Same as in section 3.4.1.1
Issuer	Same as in section 3.4.1.1
Validity	Same as in section 3.4.1.1

Subject	o = 'ACME'
	cn = 'TEST SM-DP+2'

subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters= '1.2.840.10045.3.1.7' (prime256v1) or '1.3.36.3.3.2.8.1.1.7' (brainpoolP256r1) subjectPublicKey= corresponding <PK.DPauth.ECDSA value> (see 3.4.1.5)
Extensions	Same as in section 3.4.1.1
Extension for authorityKeyIdentifier	Same as in section 3.4.1.1

Field	Value
Extension for subjectKeyIdentifier	NIST: '95 9E F7 E6 50 C1 BE 21 6A 39 19 74 27 6D 26 B8 A9 35 61 71' Brainpool: 'D7 0E FD 05 7B AC 1F 7C 55 EA 5D 8C 26 BE 16 02 92 84 5B AF'
Extension for keyUsage	Same as in section 3.4.1.1
Extension for certificatePolicies	Same as in section 3.4.1.1
Extension for subjectAltName	'2.999.12'
Extension for crlDistributionPoints	Same as in section 3.4.1.1

Table 9: CERT.DPauth.ECDSA of SM-DP+ n°2

3.4.1.5 SM-DP+ n°2 Keys and Certificate

Hereafter the generated keys and certificates of SM-DP+ n°2 for Authentication as defined in Annex A.

File name	Description
SK_S_SM_DP2auth_ECDSA_NIST.pem	NIST P-256 Private Key of the SM-DP+ n°2 for creating signatures for SM-DP+ authentication
PK_S_SM_DP2auth_ECDSA_NIST.pem	NIST P-256 Public Key of the SM-DP+ n°2 (part of the CERT_S_SM_DP2auth_ECDSA_NIST.der)
CERT_S_SM_DP2auth_ECDSA_NIST.der	Certificate of the SM-DP+ n°2 for its Public NIST P-256 key used for SM-DP+ authentication
SK_S_SM_DP2auth_ECDSA_BRP.pem	Brainpool P256r1 Private Key of the SM-DP+ n°2 for creating signatures for SM-DP+ authentication
PK_S_SM_DP2auth_ECDSA_BRP.pem	Brainpool P256r1 Public Key of the SM-DP+ n°2 (part of the CERT_S_SM_DP2auth_ECDSA_BRP.der)
CERT_S_SM_DP2auth_ECDSA_BRP.der	Certificate of the SM-DP+ n°2 for its Public Brainpool P256r1 key used for SM-DP+ authentication

Table 10: DPAuth Keys and Certificates of SM-DP+ n°2

3.4.1.6 Input data for generation

The SK.DPauth.ECDSA and PK.DPauth.ECDSA of the SM-DP+ n°2 are generated using the command lines as described in section 2.2.

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

<input_csr_file_name>: `DP2-csr.cnf` as defined in Annex A.

<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 3.4.1.4 for `serialNumber` data field.

<days> set with value defined in section 3.4.1.4 for `validity` data field.

<cert_ext_file_name>: `DPauth2-ext.cnf` as defined in Annex A.

3.4.2 DPpb

3.4.2.1 SM-DP+ n°1 Certificate for Profile Binding: definition of data to be signed

Field	Value
Version	'2'
serialNumber	'101'
Signature	algorithm = '1.2.840.10045.4.3.2' (sha256ECDSA)
Issuer	<Value of CERT.CI.ECDSA."subject" field>
Validity	1095 days (3 years)

Subject	o = 'ACME'
	cn = 'TEST SM-DP+'

subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters= '1.2.840.10045.3.1.7' (prime256v1) or '1.3.36.3.3.2.8.1.1.7' (brainpoolP256r1) subjectPublicKey= corresponding <PK.DPpb.ECDSA value> (see 3.4.2.2)
Extensions	(Sequence)
Extension for authorityKeyIdentifier	< Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for prime256v1 or brainpoolP256r1

Extension for subjectKeyIdentifier	NIST (prime256v1):
	'E6 EA F7 1E E0 FB 94 30 EC CD 1E BB 42 1F 88 14 37 C1 32 63'
	Brainpool (brainpoolP256r1):
	'A8 C6 8D F4 49 EB 71 EC 72 3E AC 13 2E 40 E4 B6 F5 46 44 FE'

Extension for keyUsage	Digital Signature ('80')
Extension for certificatePolicies	'2.23.146.1.2.1.5' (id-rspRole-dp-pb)

Field	Value
Extension for subjectAltName	'2.999.10'
Extension for crlDistributionPoints	<Value of CERT.CI.ECDSA."crlDistributionPoints" field>

Table 11: CERT.DPpb.ECDSA of SM-DP+ n°1

3.4.2.2 SM-DP+ n°1 Keys and Certificate

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_DPpb_ECDSA_NIST.pem	NIST P-256 Private Key of the SM-DP+ n°1 for creating signatures for Profile Package Binding
PK_S_SM_DPpb_ECDSA_NIST.pem	NIST P-256 Public Key of the SM-DP+ n°1 (part of the CERT_S_SM_DPpb_ECDSA_NIST.der)
CERT_S_SM_DPpb_ECDSA_NIST.der	Certificate of the SM-DP+ n°1 for its Public NIST P-256 key used for Profile Package Binding
SK_S_SM_DPpb_ECDSA_BRP.pem	Brainpool P256r1 Private Key of the SM-DP+ n°1 for creating signatures for Profile Package Binding
PK_S_SM_DPpb_ECDSA_BRP.pem	Brainpool P256r1 Public Key of the SM-DP+ n°1 (part of the CERT_S_SM_DPpb_ECDSA_BRP.der)
CERT_S_SM_DPpb_ECDSA_BRP.der	Certificate of the SM-DP+ n°1 for its Public Brainpool P256r1 key used for Profile Package Binding

Table 12: DPpb Keys and Certificates of SM-DP+ n°1

3.4.2.3 Input data for generation

The SK.DPpb.ECDSA and PK.DPpb.ECDSA of the SM-DP+ n°1 are generated using the command lines as described in section 2.2.

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

<input_csr_file_name>: `DP-csr.cnf` as defined in Annex A.

<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 3.4.2.1 for serialNumber data field.

<days> set with value defined in section 3.4.2.1 for validity data field.

<cert_ext_file_name>: `DPpb-ext.cnf` as defined in Annex A.

3.4.2.4 SM-DP+ n°2 Certificate for Profile Binding: definition of data to be signed

Field	Value
Version	Same as in section 3.4.2.1
serialNumber	'201'
Signature	Same as in section 3.4.2.1
Issuer	Same as in section 3.4.2.1
Validity	Same as in section 3.4.2.1

Subject	o = 'ACME'
	cn = 'TEST SM-DP+2'

subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters= '1.2.840.10045.3.1.7' (prime256v1) or '1.3.36.3.3.2.8.1.1.7' (brainpoolP256r1) subjectPublicKey= corresponding <PK.DPpb.ECDSA value> (see 3.4.2.5)
Extensions	Same as in section 3.4.2.1
Extension for authorityKeyIdentifier	Same as in section 3.4.2.1

Extension for subjectKeyIdentifier	NIST (prime256v1):
	'20 A3 A8 30 E9 2E E7 A4 68 C5 EB 27 BA 8D F1 84 59 AD FD D7'
	Brainpool (brainpoolP256r1):
	'31 03 8A 55 B6 BE CF 6C EA 59 DE 2F DA 14 F4 32 7F B8 B6 A9'

Extension for keyUsage	Same as in section 3.4.2.1
Extension for certificatePolicies	Same as in section 3.4.2.1
Extension for subjectAltName	'2.999.12'
Extension for crlDistributionPoints	Same as in section 3.4.2.1

Table 13: CERT.DPpb.ECDSA of SM-DP+ n°2

3.4.2.5 SM-DP+ n°2 Keys and Certificate

Hereafter the generated keys and certificates of the SM-DP+ n°2 for Profile Package Binding as defined in Annex A.

File name	Description
SK_S_SM_DP2pb_ECDSA_NIST.pem	NIST P-256 Private Key of the SM-DP+ n°2 for creating signatures for Profile Package Binding
PK_S_SM_DP2pb_ECDSA_NIST.pem	NIST P-256 Public Key of the SM-DP+ n°2 (part of the CERT_S_SM_DP2pb_ECDSA_NIST.der)

File name	Description
CERT_S_SM_DP2pb_ECDSA_NIST.der	Certificate of the SM-DP+ n°2 for its Public NIST P-256 key used for Profile Package Binding
SK_S_SM_DP2pb_ECDSA_BRP.pem	Brainpool P256r1 Private Key of the SM-DP+ n°2 for creating signatures for Profile Package Binding
PK_S_SM_DP2pb_ECDSA_BRP.pem	Brainpool P256r1 Public Key of the SM-DP+ n°2 (part of the CERT_S_SM_DP2pb_ECDSA_BRP.der)
CERT_S_SM_DP2pb_ECDSA_BRP.der	Certificate of the SM-DP+ n°2 for its Public Brainpool P256r1 key used for Profile Package Binding

Table 14: DPpb Keys and Certificates of SM-DP+ n°2

3.4.2.6 Input data for generation

The SK.DPpb.ECDSA and PK.DPpb.ECDSA of the SM-DP+ n°2 are generated using the command lines as described in section 2.2.

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

<input_csr_file_name>: `DP2-csr.cnf` as defined in Annex A.

<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 3.4.2.4 for `serialNumber` data field.

<days> set with value defined in section 3.4.2.4 for `validity` data field.

<cert_ext_file_name>: `DPpb2-ext.cnf` as defined in Annex A.

3.4.3 TLS

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

Field	Value
Version	2
serialNumber	'9'
signature	algorithm = '1.2.840.10045.4.3.2' (sha256ECDSA)
Issuer	<Value of CERT.CI.ECDSA."subject" field>
validity	1095 days (3years)

subject	o = 'ACME'
	cn = 'testsmdpplus1.example.com'

subjectPublicKeyInfo	algorithm.algorithm= '1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters=

Field	Value
	'1.2.840.10045.3.1.7' (Prime256v1) or '1.3.36.3.3.2.8.1.1.7' (brainpoolP256r1) subjectPublicKey = < PK.DP.TLS value> (see 3.4.3.2)
Extensions	(Sequence)
Extension for authorityKeyIdentifier	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for prime256v1 or brainpoolP256r1

Extension for subjectKeyIdentifier	NIST (prime256v1):
	'27 FE F1 F2 29 18 7E C7 83 ED F6 E0 29 64 A4 51 8D 57 D4 A9'
	Brainpool (brainpoolP256r1):
	'3D 33 09 83 F3 9F CC 5B D2 E4 AD 68 A6 19 A7 47 48 AE 8B 9D'

Extension for keyUsage	Critical digitalSignature ('80')
Extension for certificatePolicies	'2.23.146.1.2.1.3' (id-rspRole-dp-tls)
Extension for extendedKeyUsage	Critical TLS Web Server Authentication TLS Client Authentication
Extension for subjectAltName	DNS= testsmdpplus1.example.com SM-DP+OID = '2.999.10'
Extension for crlDistributionPoints	<Value of CERT.CI.ECDSA."crlDistributionPoints" field>

Table 15: CERT.DP.TLS for SM-DP+ n°1

3.4.3.2 SM-DP+ n°1 TLS Keys and Certificate

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+ n°1 for securing TLS connection
PK_S_SM_DP_TLS_NIST.pem	NIST P-256 Public Key of the SM-DP+ n°1 (part of the CERT_S_SM_DP_TLS_NIST.der)
CERT_S_SM_DP_TLS_NIST.der	Certificate of the SM-DP+ n°1 based on NIST P-256 for securing TLS
SK_S_SM_DP_TLS_BRP.pem	Brainpool P256r1 Private key of the SM-DP+ n°1 for securing TLS connection
PK_S_SM_DP_TLS_BRP.pem	Brainpool P256r1 Public Key of the SM-DP+ n°1 (part of the CERT_S_SM_DP_TLS_BRP.der)

File name	Description
CERT_S_SM_DP_TLS_BRP.der	Certificate of the SM-DP+ n°1 based on Brainpool P256r1 for securing TLS

Table 16: DP_TLS Keys and Certificates of SM-DP+ n°1

3.4.3.3 Input data for generation

The SK.DP.TLS and PK.DP.TLS are generated using the command lines as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: CERT_SM_DP_TLS.`csr.cnf` as defined in Annex A.

<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).

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: CERT_SM_DP_TLS.`ext.cnf` as defined in Annex A.

3.4.3.4 SM-DP+ n°2 TLS Certificate: definition of data to be signed

Field	Value
Version	2
serialNumber	'99'
Signature	algorithm = '1.2.840.10045.4.3.2' (sha256ECDSA)
Issuer	<Value of CERT.CI.ECDSA."subject" field>
Validity	1095 days (3years)

Subject	o = 'ACME'
	cn = 'testsmdpplus2.example.com'

subjectPublickeyInfo	algorithm.algorithm= '1.2.840.10045.2.1' (id-ecPublicKey)
	algorithm.parameters= '1.2.840.10045.3.1.7' (Prime256v1)
	subjectPublicKey = < PK.DP.TLS value> (see Section 3.4.3.2)

Extensions	(Sequence)
Extension for authorityKeyIdentifier	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for prime256v1
Extension for subjectKeyIdentifier	NIST (prime256v1): '9f 5f 6b 0c e7 00 32 25 2d ce 10 d3 49 a6 55 18 1b 85 3e ce'
Extension for keyUsage	Critical digitalSignature ('80')

Field	Value
Extension for certificatePolicies	'2.23.146.1.2.1.3' (id-rspRole-dp-tls)
Extension for extendedKeyUsage	Critical TLS Web Server Authentication TLS Client Authentication
Extension for subjectAltName	DNS= testsmdpplus2.example.com SM-DP+OID = '2.999.12'
Extension for crlDistributionPoints	<Value of CERT.CI.ECDSA."crlDistributionPoints" field>

Table 17: CERT.DP2.TLS

3.4.3.5 SM-DP+ n°2 TLS Keys and Certificate

Hereafter the generated SM-DP+ n°2 keys and certificates for TLS as defined in Annex A.

File name	Description
SK_S_SM_DP2_TLS_NIST.pem	NIST P-256 Private key of the SM-DP+ n°2 for securing TLS connection
PK_S_SM_DP2_TLS_NIST.pem	NIST P-256 Public Key of the SM-DP+ n°2 (part of the CERT_S_SM_DP2_TLS_NIST.der)
CERT_S_SM_DP2_TLS	CERT.DP.TLS certificate of the S_SM-DP+ n°2, based on NIST P-256

Table 18: DP_TLS Keys and Certificates of SM-DP+ n°2

3.4.3.6 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DP2_TLS.csr.cnf` as defined in Annex A.

<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).

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DP2_TLS.ext.cnf` as defined in Annex A.

3.4.3.7 SM-DP+ n°3 TLS Certificate: definition of data to be signed

Field	Value
Version	2
serialNumber	'994'
Signature	algorithm = '1.2.840.10045.4.3.2' (sha256ECDSA)
Issuer	<Value of CERT.CI.ECDSA."subject" field>
Validity	1095 days (3years)

Subject	o = 'ACME'
	cn = 'testsmdpplus4.example.com'

subjectPublicKeyInfo	algorithm.algorithm= '1.2.840.10045.2.1' (id-ecPublicKey)
	algorithm.parameters= '1.2.840.10045.3.1.7' (Prime256v1)
	subjectPublicKey = < PK.DP.TLS value> (see Section 3.4.3.2)

Extensions	(Sequence)
Extension for authorityKeyIdentifier	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for prime256v1
Extension for subjectKeyIdentifier	NIST (prime256v1): '13 0f 3d 7b b3 b0 65 ad 3c 58 78 76 bc bb 6b 84 fd 49 7a ab'
Extension for keyUsage	Critical digitalSignature ('80')
Extension for certificatePolicies	'2.23.146.1.2.1.3' (id-rspRole-dp-tls)
Extension for extendedKeyUsage	Critical TLS Web Server Authentication TLS Client Authentication
Extension for subjectAltName	DNS= testsmdpplus4.example.com SM-DP+OID = '2.999.14'
Extension for crlDistributionPoints	<Value of CERT.CI.ECDSA."crlDistributionPoints" field>

Table 19: CERT.DP4.TLS

3.4.3.8 SM-DP+ n°3 TLS Keys and Certificate

Hereafter the generated SM-DP+ n°3 keys and certificates for TLS as defined in Annex A.

File name	Description
SK_S_SM_DP4_TLS.pem	NIST P-256 Private key of the SM-DP+ n°3 for securing TLS connection

File name	Description
PK_S_SM_DP4_TLS.pem	NIST P-256 Public Key of the SM-DP+ n°3 (part of the CERT_S_SM_DP4_TLS.der)
CERT_S_SM_DP4_TLS.der	CERT.DP.TLS certificate of the S_SM-DP+ n°3, based on NIST P-256

Table 20: DP_TLS Keys and Certificates of SM-DP+ n°3

3.4.3.9 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DP4_TLS.csr.cnf` as defined in Annex A.

<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).

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DP4_TLS.ext.cnf` as defined in Annex A.

3.4.3.10 SM-DP+ n°4 TLS Certificate: definition of data to be signed

Field	Value
Version	2
serialNumber	'998'
Signature	algorithm = '1.2.840.10045.4.3.2' (sha256ECDSA)
Issuer	<Value of CERT.CI.ECDSA."subject" field>
Validity	1095 days (3years)

Subject	o = 'ACME'
	cn = 'testsmdpplus8.example.com'

subjectPublickeyInfo	algorithm.algorithm= '1.2.840.10045.2.1' (id-ecPublicKey)
	algorithm.parameters= '1.2.840.10045.3.1.7' (Prime256v1)
	subjectPublicKey = < PK.DP.TLS value> (see Section 3.4.3.2)

Extensions	(Sequence)
Extension for authorityKeyIdentifier	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for prime256v1
Extension for subjectKeyIdentifier	NIST (prime256v1):

Field	Value
	'b8 7e 0a 73 f2 44 d5 99 4c 28 61 e6 ea 6e 30 70 d6 34 2a 53'
Extension for keyUsage	Critical digitalSignature ('80')
Extension for certificatePolicies	'2.23.146.1.2.1.3' (id-rspRole-dp-tls)
Extension for extendedKeyUsage	Critical TLS Web Server Authentication TLS Client Authentication
Extension for subjectAltName	DNS= testsmdpplus8.example.com SM-DP+OID = '2.999.18'
Extension for crlDistributionPoints	<Value of CERT.CI.ECDSA."crlDistributionPoints" field>

Table 21: CERT.DP8.TLS

3.4.3.11 SM-DP+ n°4 TLS Keys and Certificate

Hereafter the generated SM-DP+ n°4 keys and certificates for TLS as defined in Annex A.

File name	Description
SK_S_SM_DP8_TLS.pem	NIST P-256 Private key of the SM-DP+ n°4 for securing TLS connection
PK_S_SM_DP8_TLS.pem	NIST P-256 Public Key of the SM-DP+ n°4 (part of the CERT_S_SM_DP8_TLS.der)
CERT_S_SM_DP8_TLS.der	CERT.DP.TLS certificate of the S_SM-DP+ n°4, based on NIST P-256

Table 22: DP_TLS Keys and Certificates of SM-DP+ n°4

3.4.3.12 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DP8_TLS.csr.cnf` as defined in Annex A.

<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).

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DP8_TLS.ext.cnf` as defined in Annex A.

3.5 SM-DS

3.5.1 DSauth

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

Field	Value
Version	2
serialNumber	'7495'
Signature	algorithm = '1.2.840.10045.4.3.2' (sha256ECDSA)
Issuer	<Value of CERT.CI.ECDSA."subject" field>
Validity	1095 days (3 years)

Subject	o = 'ACME'
	cn = 'TEST SM-DS'

subjectPublicKeyInfo	algorithm.algorithm= '1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters= '1.2.840.10045.3.1.7' (Prime256v1) or '1.3.36.3.3.2.8.1.1.7' (brainpoolP256r1) subjectPublicKey = < PK.DSauth.ECDSA value> (see 3.5.1.2)
Extensions	(Sequence)
Extension for Authority Key Identifier	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for prime256v1 or brainpoolP256r1

Extension for subjectKeyIdentifier	NIST (prime256v1):
	'C1 F4 06 4B 3B 25 8A FB 61 38 8B 3F F2 EE 6A 61 E2 C4 4D 72'
	Brainpool (brainpoolP256r1):
	'F0 5F 0B 54 AE E8 AE 01 08 F0 1D EF 54 8E D9 85 97 14 DD 48'

KeyUsage Extension	Digital Signature ('80')
Extension for Certificate Policy	'2.23.146.1.2.1.7' (id-rspRole-ds-auth)
Extension for subjectAltName	SM-DS OID = '2.999.15'
Extension for CRL Distribution Points	<Value of CERT.CI.ECDSA."crlDistributionPoints" field>

Table 23: CERT.DSauth.ECDSA

3.5.1.2 SM-DS Keys and Certificate

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

File name	Description
SK_S_SM_DSauth_ECDSA_NIST.pem	NIST P-256 Private Key of the SM-DS for creating signatures for SM-DS authentication
PK_S_SM_DSauth_ECDSA_NIST.pem	NIST P-256 Public Key of the SM-DS (part of the CERT_S_SM_DSauth_ECDSA_NIST.der)
CERT_S_SM_DSauth_ECDSA_NIST.der	Certificate of the SM-DS for its Public NIST P-256 key used for SM-DS authentication
SK_S_SM_DSauth_ECDSA_BRP.pem	Brainpool P256r1 Private Key of the SM-DS for creating signatures for SM-DS authentication
PK_S_SM_DSauth_ECDSA_BRP.pem	Brainpool P256r1 Public Key of the SM-DS (part of the CERT_S_SM_DSauth_ECDSA_BRP.der)
CERT_S_SM_DSauth_ECDSA_BRP.der	Certificate of the SM-DS for its Public Brainpool P256r1 key used for SM-DS authentication

Table 24: DSauth Keys and Certificates

3.5.1.3 Input data for generation

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

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

<input_csr_file_name>: `DSauth-csr.cnf` as defined in Annex A.

<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 3.5.1.1 for serialNumber data field.

<days> set with value defined in section 3.5.1.1 for validity data field.

<cert_ext_file_name>: `DSauth-ext.cnf` as defined in Annex A.

3.5.2 TLS

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

Field	Value
Version	2
serialNumber	'1223334444'
Signature	SHA256ECDSA
Issuer	<Value of CERT.CI.ECDSA."subject" field>
Validity	1095 days (3years)

Subject	o = 'RSPTEST'
	cn = 'testrootsmds.example.com'

Field	Value
subjectPublicKeyInfo	algorithm.algorithm= '1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters= '1.2.840.10045.3.1.7' (Prime256v1) or '1.3.36.3.3.2.8.1.1.7' (BrainpoolP256r1) subjectPublicKey = < PK.DS.TLS value>
Extensions	(Sequence)
Extension for Authority Key Identifier	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for Prime256v1 or BrainpoolP256r1
Extension for Subject Key Identifier	NIST: 'A0 36 C1 62 75 35 1E C7 B0 15 53 A1 3F 83 E2 8D 44 00 BD 0A' Brainpool: '73 99 CA C7 B1 5F AB 2F F9 33 CF 2D 22 15 E4 84 4A DE F8 05'
Extension for Key usage	Critical digitalSignature ('80')
Extension for Certificate Policies	'2.23.146.1.2.1.6' (id-rspRole-ds-tls)
Extension for Extended Key usage	Critical TLS Web Server Authentication , TLS Web Client Authentication
Extension for subjectAltName	DNS= testrootsmds.example.com SM-DS OID = '2.999.15'
Extension for CRL Distribution Points	<Value of CERT.CI.ECDSA."crlDistributionPoints" field>

Table 25: CERT.DS.TLS for SM-DS n°1

3.5.2.2 SM-DS n°1 TLS Keys and Certificate

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

File name	Description
SK_SM_DS_TLS_NIST.pem	NIST P-256 Private key of the SM-DS n°1 for securing TLS connection
PK_SM_DS_TLS_NIST.pem	NIST P-256 Public Key of the SM-DS n°1 (part of the CERT_S_SM_DS_TLS_NIST.der)
CERT_SM_DS_TLS_NIST.der	Certificate of the SM-DS n°1 based on NIST P-256 for securing TLS
SK_SM_DS_TLS_BRP.pem	Brainpool P256r1 Private key of the SM-DS n°1 for securing TLS connection
PK_SM_DS_TLS_BRP.pem	Brainpool P256r1 Public Key of the SM-DS n°1 (part of the CERT_S_SM_DP_TLS_BRP.der)
CERT_SM_DS_TLS_BRP.der	Certificate of the SM-DS n°1 based on Brainpool P256r1 for securing TLS

Table 26: DS_TLS Keys and Certificates for SM-DS n°1

3.5.2.3 Input data for generation

The SK.DS.TLS and PK.DS.TLS are generated using the command lines as described in section 2.2.

The CERT.DS.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: CERT_SM_DS_TLS.csr.cnf as defined in Annex A.

<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 3.5.2.1 for `serialNumber` data field.

<days> set with value defined in section 3.5.2.1 for `validity` data field.

<cert_ext_file_name>: CERT_SM_DS_TLS`.ext.cnf` as defined in Annex A.

3.5.2.4 SM-DS n°2 TLS Certificate: definition of data to be signed

Field	Value
Version	2
serialNumber	'122333444455555'
Signature	SHA256ECDSA
Issuer	<Value of CERT.CI.ECDSA."subject" field>
Validity	1095 days (3years)

Subject	o = 'RSPTEST'
	cn = 'testsmds1.example.com'

subjectPublicKeyInfo	algorithm.algorithm= '1.2.840.10045.2.1' (id-ecPublicKey)
	algorithm.parameters=
	'1.2.840.10045.3.1.7' (Prime256v1)
	subjectPublicKey = < PK.DS.TLS value> (see Section 3.5.2.2)

Extensions	(Sequence)
Extension for Authority Key Identifier	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for Prime256v1
Extension for Subject Key Identifier	NIST: '53 82 04 27 91 71 ed 3d 0a 79 c0 ad 61 a5 35 31 2c 86 48 6c'
Extension for Key usage	Critical digitalSignature ('80')
Extension for Certificate Policies	'2.23.146.1.2.1.6' (id-rspRole-ds-tls)
Extension for Extended Key usage	Critical TLS Web Server Authentication , TLS Web Client Authentication

Field	Value
Extension for subjectAltName	DNS= testsmds1.example.com SM-DS OID = '2.999.15.2'
Extension for CRL Distribution Points	<Value of CERT.CI.ECDSA."crlDistributionPoints" field>

Table 27: CERT.DS2.TLS

3.5.2.5 SM-DS n°2 TLS Keys and Certificate

Hereafter the generated SM-DS n°2 keys and certificates for TLS as defined in Annex A.

File name	Description
SK_S_SM_DS2_TLS_NIST.pem	NIST P-256 Private key of the SM-DS n°2 for securing TLS connection
PK_S_SM_DS2_TLS_NIST.pem	NIST P-256 Public Key of the SM-DS n°2 (part of the CERT_S_SM_DS2_TLS_NIST.der)
CERT_S_SM_DS2_TLS_NIST.der	CERT.DS.TLS certificate of the S_SM-DS n°2, based on NIST P-256

Table 28: DS_TLS Keys and Certificates for SM-DS n°2

3.5.2.6 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DS.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DS2_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DS2_TLS.ext.cnf` as defined in Annex A.

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 eUICC

Void

4.2 SM-DP+

4.2.1 DPauth

4.2.1.1 DPAuth – Invalid Signature

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

All the data to be signed are the same as the ones defined in 3.4.1.1.

4.2.1.1.2 SM-DP+ Certificate

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.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.der	Certificate of the SM-DP+ with invalid signature for its Public Brainpool P256r1 key used for SM-DP+ authentication

Table 29: DPauth_INV_SIGN Certificates

4.2.1.1.3 Input data for generation

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

- NIST signature: 10 bytes are replaced by random values
- Brainpool signature: 8 bytes are replaced by random values

4.2.1.2 DPAuth – 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].

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

Field	Value
Version	See section 3.4.1.1
serialNumber	900
Signature	See section 3.4.1.1
Issuer	See section 3.4.1.1
Validity	See section 3.4.1.1
Subject	See section 3.4.1.1

Field	Value
subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters= '1.2.840.10045.3.1.1' (prime192v1) or '1.3.36.3.3.2.8.1.1.3' (brainpoolP192r1) subjectPublicKey= corresponding <PK.DPauth.ECDSA value> (see 3.4.1.1)
Extensions	(Sequence)
Extension for authorityKeyIdentifier	See section 3.4.1.1

Extension for subjectKeyIdentifier	NIST (prime192v1):
	'9B 3A 9E 3D 46 E7 8F 19 27 29 A8 EF 4A 46 20 6A 2C CA B2 D2'
	Brainpool (brainpoolP192r1):
	'0F 80 D8 E3 DF 68 58 8D 6E AC 72 35 A6 8F 9° 59 E1 9A 3B E9'

Extension for keyUsage	See section 3.4.1.1
Extension for certificatePolicies	See section 3.4.1.1
Extension for subjectAltName	See section 3.4.1.1
Extension for crlDistributionPoints	See section 3.4.1.1

Table 30: CERT.DPauth.ECDSA with Invalid Curve

4.2.1.2.2 SM-DP+ Keys and Certificate

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

File name	Description
SK_S_SM_DPauth_ECDSA_NIST192.pem	NIST P-192 Private Key of the SM-DP+ for creating signatures for SM-DP+ authentication
PK_S_SM_DPauth_ECDSA_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.der	Certificate of the SM-DP+ for its Public NIST P-192 key used for SM-DP+ authentication
SK_S_SM_DPauth_ECDSA_BRP192.pem	Brainpool P-192 Private Key of the SM-DP+ for creating signatures for SM-DP+ authentication
PK_S_SM_DPauth_ECDSA_BRP192.pem	Brainpool P-192 Public Key of the SM-DP+ (part of the CERT_S_SM_DPauth_INV_CURVE_BRP192.der)

File name	Description
CERT_S_SM_DPauth_INV_CURVE_BRP 192.der	Certificate of the SM-DP+ for its Public Brainpool P-192 key used for SM-DP+ authentication

Table 31: DPauth Keys and Certificates with invalid curve

4.2.1.2.3 Input data for generation

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

```
openssl ecparam -name prime192v1 -genkey -out SK_S_SM_DPauth_ECDSA_NIST192.pem

openssl ec -in SK_S_SM_DPauth_ECDSA_NIST192.pem -pubout -out
   PK_S_SM_DPauth_ECDSA_NIST192.pem
```

Command lines for the generation of the SK.DPauth.ECDSA and the corresponding PK.DPauth.ECDSA for Brainpool P192r1 curve:

```
openssl ecparam -name brainpoolP192r1 -genkey -out SK_S_SM_DPauth_ECDSA_BRP192.pem

openssl ec -in SK_S_SM_DPauth_ECDSA_BRP192.pem -pubout -out
   PK_S_SM_DPauth_ECDSA_BRP192.pem
```

The CERT.DPauth.ECDSA are generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `DP-csr.cnf` as defined in Annex A.

<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 4.2.1.2.1 for `serialNumber` data field.

<days> set with value defined in section 4.2.1.2.1 for `validity` data field.

<cert_ext_file_name>: `DPauth-ext.cnf` as defined in Annex A.

4.2.2 DPpb

4.2.2.1 DPpb – Invalid Signature

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

All the data to be signed are the same as the ones defined in 3.4.2.1.

4.2.2.1.2 SM-DP+ Certificate

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.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.der	Certificate of the SM-DP+ with invalid signature for its Public Brainpool P256r1 key used for Profile Package Binding

Table 32: DPpb Certificates with invalid signature

4.2.2.1.3 Input data for generation

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

- NIST signature: 10 bytes are replaced by random values
- Brainpool signature: 8 bytes are replaced by random values

4.2.2.2 DPpb – Invalid Curve

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

Field	Value
Version	See section 3.4.2.1
serialNumber	901
Signature	See section 3.4.2.1
Issuer	See section 3.4.2.1
Validity	See section 3.4.2.1
Subject	See section 3.4.2.1
subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters= '1.2.840.10045.3.1.1' (prime192v1) '1.3.36.3.3.2.8.1.1.3' (brainpoolP192r1) subjectPublicKey= corresponding <PK.DPpb.ECDSA value> (see 3.4.2.1)
Extensions	(Sequence)
Extension for authorityKeyIdentifier	See section 3.4.2.1

Extension for subjectKeyIdentifier	NIST (prime192v1):
	'B5 49 B2 F1 2B FB 70 B8 BE 10 3E A5 6E D9 D8 21 1E 62 AB 89'
	Brainpool (brainpoolP192r1):
	'E9 B4 02 A4 55 F7 CE A5 25 A1 56 5D 16 7D 94 A3 0C B1 A5 5E’

Extension for keyUsage	See section 3.4.2.1

Field	Value
Extension for certificatePolicies	See section 3.4.2.1
Extension for subjectAltName	See section 3.4.2.1
Extension for crlDistributionPoints	See section 3.4.2.1

Table 33: CERT.DPpb.ECDSA with invalid curve

4.2.2.2.2 SM-DP+ Keys and Certificate

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_ECDSA_NIST192.pem	NIST P-192 Private Key of the SM-DP+ for creating signatures for Profile Package Binding
PK_S_SM_DPpb_ECDSA_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.der	Certificate of the SM-DP+ for its Public NIST P-192 key used for Profile Package Binding
SK_S_SM_DPpb_ECDSA_BRP192.pem	Brainpool P-192 Private Key of the SM-DP+ for creating signatures for Profile Package Binding
PK_S_SM_DPpb_ECDSA_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.der	Certificate of the SM-DP+ for its Public Brainpool P-192 key used for Profile Package Binding

Table 34: DPpb Keys and Certificates with invalid curve

4.2.2.2.3 Input data for generation

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

```
openssl ecparam -name prime192v1 -genkey -out SK_S_SM_DPpb_ECDSA_NIST192.pem

openssl ec -in SK_S_SM_DPpb_ECDSA_NIST192.pem -pubout -out
   PK_S_SM_DPpb_ECDSA_NIST192.pem
```

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

```
openssl ecparam -name brainpoolP192r1 -genkey -out SK_S_SM_DPpb_ECDSA_BRP192.pem

openssl ec -in SK_S_SM_DPpb_ECDSA_BRP192.pem -pubout -out
    PK_S_SM_DPpb_ECDSA_BRP192.pem
```

The CERT.DPpb.ECDSA are generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `DP-csr.cnf` as defined in Annex A.

<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 4.2.2.2.1 for `serialNumber` data field.

<days> set with value defined in section 4.2.2.2.1 for `validity` data field.

<cert_ext_file_name>: `DPpb-ext.cnf` as defined in Annex A.

4.2.3 TLS

4.2.3.1 TLS – Invalid Signature

4.2.3.1.1 SM-DP+ TLS Certificate: Definition of data to be signed

All the data to be signed are the same as the ones defined in 3.4.3.1.

4.2.3.1.2 SM-DP+ Certificate

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.der	Certificate of the SM-DP+ with invalid signature for its Public NIST P-256 key
CERT_S_SM_DP_TLS_INV_SIGN_BRP.der	Certificate of the SM-DP+ with invalid signature for its Public Brainpool P256r1 key

Table 35: DP_TLS Certificates with invalid signature

4.2.3.1.3 Input data for generation

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

- Least significant byte of CERT_S_SM_DP_TLS_NIST.der signature increased by 1
- Least significant byte of CERT_S_SM_DP_TLS_BRP.der signature increased by 1

4.2.3.2 TLS – Invalid Curve

4.2.3.2.1 SM-DP+ TLS Certificate: definition of data to be signed

Field	Value
Version	Same as in section 3.4.3
serialNumber	Same as in section 3.4.3
Signature	Same as in section 3.4.3
Issuer	Same as in section 3.4.3
Validity	Same as in section 3.4.3
Subject	Same as in section 3.4.3

subjectPublicKeyInfo	algorithm.algorithm = '1.2.840.10045.2.1' (id-ecPublicKey)
	algorithm.parameters = '1.3.132.0.34' (secp384r1) subjectPublicKey = < PK.DP.TLS value> (see 3.4.3.2)

Extensions	Same as in section 3.4.3
Extension for authorityKeyIdentifier	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for secp384r1
Extension for subjectKeyIdentifier	NIST (secp384r1): '0a 8f 46 e4 bd df e3 3f b0 1c 4b 0c c6 2f 14 0b 3b 11 91 c6'
Extension for keyUsage	Same as in section 3.4.3
Extension for certificatePolicies	Same as in section 3.4.3
Extension for extendedKeyUsage	Same as in section 3.4.3
Extension for subjectAltName	Same as in section 3.4.3
Extension for crlDistributionPoints	Same as in section 3.4.3

Table 36: CERT_S_SM_DP_TLS_INV_CURVE

4.2.3.2.2 SM-DP+ TLS Keys and Certificate

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

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_CERT_S_SM_DP_TLS_INV_CURVE.pem	NIST P-384 Private key of the SM-DP+ for securing TLS connection with

File name	Description
PK_CERT_S_SM_DP_TLS_INV_CURVE.pem	NIST P-384 Public Key of the SM-DP+ (part of the CERT_S_SM_DP_TLS_INV_CURVE.der)
CERT_S_SM_DP_TLS_INV_CURVE.der	CERT.DP.TLS certificate of the S_SM-DP+, based on NIST P-384 curve

Table 37: DP_TLS Keys and Certificates with invalid curve

4.2.3.2.3 Input data for generation

The Private Key is generated using the following command line:

```
openssl ecparam -name secp384r1 -genkey -out <sk_file_name>
```

The Public Key is generated as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DP_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DP_TLS.ext.cnf` as defined in Annex A.

4.2.3.3 TLS – Invalid Certificate Policy

4.2.3.3.1 SM-DP+ TLS Certificate: definition of data to be signed

Field	Value
Version	Same as in section 3.4.3.1
serialNumber	Same as in section 3.4.3.1
Signature	Same as in section 3.4.3.1
Issuer	Same as in section 3.4.3.1
Validity	Same as in section 3.4.3.1
Subject	Same as in section 3.4.3.1
subjectPublicKeyInfo	Same as in section 3.4.3.1
Extensions	Same as in section 3.4.3.1
Extension for authorityKeyIdentifier	Same as in section 3.4.3.1

Field	Value
Extension for subjectKeyIdentifier	Same as in section 3.4.3.1
Extension for keyUsage	Same as in section 3.4.3.1
Extension for certificatePolicies	'2.23.146.1.2.1.4' (id-rspRole-dp-auth)
Extension for extendedKeyUsage	Same as in section 3.4.3.1
Extension for subjectAltName	Same as in section 3.4.3.1
Extension for crlDistributionPoints	Same as in section 3.4.3.1

Table 38: CERT_S_SM_DP_TLS_INV_CERT_POL

4.2.3.3.2 SM-DP+ TLS Keys and Certificate

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 .der	CERT.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.

Table 39: DS_TLS Keys and Certificate with invalid certificatePolicies extension

4.2.3.3.3 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DP_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DP_TLS_INV_CERT_POL.ext.cnf` as defined in Annex A.

4.2.3.4 TLS – Missing Critical Extension

4.2.3.4.1 SM-DP+ TLS Certificate: definition of data to be signed

Field	Value
Version	Same as in section 3.4.3.1
serialNumber	Same as in section 3.4.3.1
Signature	Same as in section 3.4.3.1
Issuer	Same as in section 3.4.3.1
Validity	Same as in section 3.4.3.1
Subject	Same as in section 3.4.3.1
subjectPublicKeyInfo	Same as in section 3.4.3.1
Extensions	Same as in section 3.4.3.1
Extension for authorityKeyIdentifier	Same as in section 3.4.3.1
Extension for subjectKeyIdentifier	Same as in section 3.4.3.1
Extension for keyUsage	Same as in section 3.4.3.1
Extension for certificatePolicies	Same as in section 3.4.3.1
Extension for extendedKeyUsage	Absent
Extension for subjectAltName	Same as in section 3.4.3.1
Extension for crlDistributionPoints	Same as in section 3.4.3.1

Table 40: CERT_S_SM_DP_TLS_INV_CRITICAL_EXT

4.2.3.4.2 SM-DP+ TLS Keys and Certificate

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.der	CERT.DP.TLS certificate of the S_SM-DP+ with one of the critical extensions not present, formatted as X.509 certificate.

Table 41: DP_TLS Keys and Certificates with critical extension not present

4.2.3.4.3 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DP_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DP_TLS_INV_CRITICAL_EXT.ext.cnf` as defined in Annex A.

4.2.3.5 TLS – Invalid Extended Key Usage

4.2.3.5.1 SM-DP+ TLS Certificate: definition of data to be signed

Field	Value
Version	Same as in section 3.4.3.1
serialNumber	Same as in section 3.4.3.1
Signature	Same as in section 3.4.3.1
Issuer	Same as in section 3.4.3.1
Validity	Same as in section 3.4.3.1
Subject	Same as in section 3.4.3.1
subjectPublicKeyInfo	Same as in section 3.4.3.1
Extensions	Same as in section 3.4.3.1
Extension for authorityKeyIdentifier	Same as in section 3.4.3.1
Extension for	Same as in section 3.4.3.1

Field	Value
subjectKeyIdentifier	
Extension for keyUsage	Same as in section 3.4.3.1
Extension for certificatePolicies	Same as in section 3.4.3.1
Extension for extendedKeyUsage	Critical TLS Client Authentication.1
Extension for subjectAltName	Same as in section 3.4.3.1
Extension for crlDistributionPoints	Same as in section 3.4.3.1

Table 42: CERT_S_SM_DP_TLS_INV_EXT_KEY_USAGE

4.2.3.5.2 SM-DP+ TLS Keys and Certificate

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.der	CERT.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 43: DP+ TLS Certificates with invalid 'extended key usage'

4.2.3.5.3 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DP_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DP_TLS_INV_EXT_KEY_USAGE.ext.cnf` as defined in Annex A.

4.2.3.6 TLS – Invalid Key Usage

4.2.3.6.1 SM-DP+ TLS Certificate: definition of data to be signed

Field	Value
Version	Same as in section 3.4.3.1
serialNumber	Same as in section 3.4.3.1
Signature	Same as in section 3.4.3.1
Issuer	Same as in section 3.4.3.1
Validity	Same as in section 3.4.3.1
Subject	Same as in section 3.4.3.1
subjectPublicKeyInfo	Same as in section 3.4.3.1
Extensions	Same as in section 3.4.3.1
Extension for authorityKeyIdentifier	Same as in section 3.4.3.1
Extension for subjectKeyIdentifier	Same as in section 3.4.3.1
Extension for keyUsage	Critical 'keyAgreement' ('08')
Extension for certificatePolicies	Same as in section 3.4.3.1
Extension for extendedKeyUsage	Same as in section 3.4.3.1
Extension for subjectAltName	Same as in section 3.4.3.1
Extension for crlDistributionPoints	Same as in section 3.4.3.1

Table 44: CERT_S_SM_DP_TLS_INV_KEY_USAGE

4.2.3.6.2 SM-DP+ TLS Keys and Certificate

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.der	CERT.DP.TLS certificate of the S_SM-DP+ with invalid 'key usage' extension (not set to 'digitalSignature'), formatted as X.509 certificate.

Table 45: DP+ TLS Keys and Certificates with invalid 'key usage' extension

4.2.3.6.3 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DP_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DP_TLS_INV_KEY_USAGE.ext.cnf` as defined in Annex A.

4.2.3.7 TLS – Expired Certificate

4.2.3.7.1 SM-DP+ TLS Certificate: definition of data to be signed

Field	Value
version	Same as in section 3.4.3.1
serialNumber	Same as in section 3.4.3.1
signature	Same as in section 3.4.3.1
Issuer	Same as in section 3.4.3.1
Validity	expired on 2nd April 2020
Subject	Same as in section 3.4.3.1
subjectPublicKeyInfo	Same as in section 3.4.3.1
Extensions	Same as in section 3.4.3.1
Extension for authorityKeyIdentifier	Same as in section 3.4.3.1
Extension for subjectKeyIdentifier	Same as in section 3.4.3.1
Extension for keyUsage	Same as in section 3.4.3.1
Extension for certificatePolicies	Same as in section 3.4.3.1
Extension for extendedKeyUsage	Same as in section 3.4.3.1
Extension for subjectAltName	Same as in section 3.4.3.1
Extension for	Same as in section 3.4.3.1

Field	Value
crlDistributionPoints	

Table 46: CERT_S_SM_DP_TLS_EXPIRED

4.2.3.7.2 SM-DP+ TLS Keys and Certificate

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

Table 47: DP+ TLS Keys and expired Certificates

4.2.3.7.3 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following changes:

<input_csr_file_name>: `CERT_S_SM_DP_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 4.2.7.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DP_TLS.ext.cnf` as defined in Annex A.

4.3 SM-DS

4.3.1 DSauth

4.3.1.1 DSauth – Invalid Signature

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

All the data to be signed are the same as the ones defined in 3.5.1.1.

4.3.1.1.2 SM-DS Certificate

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.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.der	Certificate of the SM-DS with invalid signature for its Public Brainpool P256r1 key used for SM-DP+ authentication

Table 48: DS TLS Certificates with invalid signature

4.3.1.1.3 Input data for generation

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

- NIST signature: 10 bytes are replaced by random values
- Brainpool signature: 8 bytes are replaced by random values

4.3.1.2 DSauth - 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].

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

Field	Value
Version	See section 3.5.1.1
serialNumber	903
Signature	See section 3.5.1.1
Issuer	See section 3.5.1.1
Validity	See section 3.5.1.1
Subject	See section 3.5.1.1
subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters= '1.2.840.10045.3.1.1' (prime192v1) '1.3.36.3.3.2.8.1.1.3' (brainpoolP192r1) subjectPublicKey= corresponding <PK.DPauth.ECDSA value> (see 3.5.1.2)
Extensions	(Sequence)
Extension for authorityKeyIdentifier	See section 3.5.1.1

Field	Value

Extension for subjectKeyIdentifier	NIST (prime192v1):
	'61 20 11 BC 54 84 9B EE AF 59 79 49 4E FC 56 2F FB 3E 0D 72'
	Brainpool (brainpoolP192r1):
	'58 E0 39 F8 09 8E 21 81 0C 66 9A F3 4A 2D E9 24 C3 D1 A0 7E'

Extension for keyUsage	See section 3.5.1.1
Extension for certificatePolicies	See section 3.5.1.1
Extension for subjectAltName	See section 3.5.1.1
Extension for crlDistributionPoints	See section 3.5.1.1

Table 49: CERT.DSauth.ECDSA with Invalid Curve

4.3.1.2.2 SM-DS Keys and Certificate

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

File name	Description
SK_S_SM_DSauth_ECDSA_NIST192.pem	NIST P-192 Private Key of the SM-DS for creating signatures for SM-DS authentication
PK_S_SM_DSauth_ECDSA_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.der	Certificate of the SM-DS for its Public NIST P-192 key used for SM-DS authentication
SK_S_SM_DSauth_ECDSA_BRP192.pem	Brainpool P-192 Private Key of the SM-DS for creating signatures for SM-DS authentication
PK_S_SM_DSauth_ECDSA_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.der	Certificate of the SM-DS for its Public Brainpool P-192 key used for SM-DS authentication

Table 50: DS TLS Certificates with invalid curve

4.3.1.2.3 Input data for generation

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

```
openssl ecparam -name prime192v1 -genkey -out SK_S_SM_DSauth_ECDSA_NIST192.pem

openssl ec -in SK_S_SM_DSauth_ECDSA_NIST192.pem -pubout -out
   PK_S_SM_DSauth_ECDSA_NIST192.pem
```

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

```
openssl ecparam -name brainpoolP192r1 -genkey -out SK_S_SM_DSauth_ECDSA_BRP192.pem

openssl ec -in SK_S_SM_DSauth_ECDSA_BRP192.pem -pubout -out
   PK_S_SM_DSauth_ECDSA_BRP192.pem
```

The CERT.DSauth.ECDSA are generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `DSauth-csr.cnf` as defined in Annex A.

<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 4.3.1.2.1 for `serialNumber` data field.

<days> set with value defined in section 4.3.1.2.1 for `validity` data field.

<cert_ext_file_name>: `DSauth-ext.cnf` as defined in Annex A.

4.3.2 TLS

4.3.2.1 TLS – Invalid Signature

4.3.2.1.1 SM-DS TLS Certificate: definition of data to be signed

All the data to be signed are the same as the ones defined in 3.5.2.1.

4.3.2.1.2 SM-DS Certificate

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.der	Certificate of the SM-DS with invalid signature for its Public NIST P-256 key
CERT_S_SM_DS_TLS_INV_SIGN_BRP.der	Certificate of the SM-DS with invalid signature for its Public Brainpool P256r1 key

Table 51: DS TLS Certificates with invalid signature

4.3.2.1.3 Input data for generation

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

- Least significant byte of CERT_S_SM_DS_TLS_NIST.der signature increased by 1
- Least significant byte of CERT_S_SM_DS_TLS_BRP.der signature increased by 1

4.3.2.2 TLS – Invalid Curve

4.3.2.2.1 SM-DS TLS Certificate: definition of data to be signed

Field	Value
version	Same as in section 3.5.2.1
serialNumber	Same as in section 3.5.2.1
signature	Same as in section 3.5.2.1
issuer	Same as in section 3.5.2.1
validity	Same as in section 3.5.2.1
subject	Same as in section 3.5.2.1

subjectPublicKeyInfo	algorithm.algorithm = '1.2.840.10045.2.1' (id-ecPublicKey)
	algorithm.parameters = '1.3.132.0.34' (secp384r1) subjectPublicKey = < PK.DS.TLS value> (see 3.5.2.1)

Extensions	Same as in section 3.5.2.1
Extension for authorityKeyIdentifier	<Value of CERT.CI.ECDSA."subjectKeyIdentifier" field> for secp384r1
Extension for subjectKeyIdentifier	NIST (secp384r1): '0a 8f 46 e4 bd df e3 3f b0 1c 4b 0c c6 2f 14 0b 3b 11 91 c6'
Extension for keyUsage	Same as in section 3.5.2.1
Extension for certificatePolicies	Same as in section 3.5.2.1
Extension for extendedKeyUsage	Same as in section 3.5.2.1
Extension for subjectAltName	Same as in section 3.5.2.1
Extension for crlDistributionPoints	Same as in section 3.5.2.1

Table 52: CERT_S_SM_DS_TLS_INV_CURVE

4.3.2.2.2 SM-DS TLS Keys and Certificate

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

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_CERT_S_SM_DP_TLS_INV_CURVE.pem	NIST P-384 Private key of the SM-DP+ for securing TLS connection with
PK_CERT_S_SM_DP_TLS_INV_CURVE.pem	NIST P-384 Public Key of the SM-DP+ (part of the CERT_S_SM_DS_TLS_INV_CURVE.der)
CERT_S_SM_DS_TLS_INV_CURVE.der	CERT.DS.TLS certificate of the S_SM-DS, based on NIST P-384 curve

Table 53: DS TLS Certificates with invalid curve

4.3.2.2.3 Input data for generation

The Private and Public Keys are the same as for CERT_S_SM_DP_TLS_INV_CURVE.der.

The CERT.DS.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DS_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DS_TLS.ext.cnf` as defined in Annex A.

4.3.2.3 TLS – Invalid Certificate Policy

4.3.2.3.1 SM-DS TLS Certificate: definition of data to be signed

Field	Value
Version	Same as in section 3.5.2.1
serialNumber	Same as in section 3.5.2.1
Signature	Same as in section 3.5.2.1
Issuer	Same as in section 3.5.2.1
validity	Same as in section 3.5.2.1
subject	Same as in section 3.5.2.1
subjectPublickeyInfo	Same as in section 3.5.2.1
Extensions	Same as in section 3.5.2.1
Extension for	Same as in section 3.5.2.1

Field	Value
authorityKeyIdentifier	
Extension for subjectKeyIdentifier	Same as in section 3.5.2.1
Extension for keyUsage	Same as in section 3.5.2.1
Extension for certificatePolicies	'2.23.146.1.2.1.4' (id-rspRole-dp-auth)
Extension for extendedKeyUsage	Same as in section 3.5.2.1
Extension for subjectAltName	Same as in section 3.5.2.1
Extension for crlDistributionPoints	Same as in section 3.5.2.1

Table 54: CERT_S_SM_DS_TLS_INV_CERT_POL

4.3.2.3.2 SM-DS TLS Keys and Certificate

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 .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 55: DS TLS Certificates with invalid 'certificate policies'

4.3.2.3.3 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DS.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DS_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DS_TLS_INV_CERT_POL.ext.cnf` as defined in Annex A.

4.3.2.4 TLS – Missing Critical Extension

4.3.2.4.1 SM-DS TLS Certificate: definition of data to be signed

Field	Value
version	Same as in section 3.5.2.1
serialNumber	Same as in section 3.5.2.1
signature	Same as in section 3.5.2.1
issuer	Same as in section 3.5.2.1
validity	Same as in section 3.5.2.1
subject	Same as in section 3.5.2.1
subjectPublicKeyInfo	Same as in section 3.5.2.1
Extensions	Same as in section 3.5.2.1
Extension for authorityKeyIdentifier	Same as in section 3.5.2.1
Extension for subjectKeyIdentifier	Same as in section 3.5.2.1
Extension for keyUsage	Same as in section 3.5.2.1
Extension for certificatePolicies	Same as in section 3.5.2.1
Extension for extendedKeyUsage	Absent
Extension for subjectAltName	Same as in section 3.5.2.1
Extension for CrlDistributionPoints	Same as in section 3.5.2.1

Table 56: CERT_S_SM_DS_TLS_INV_CRITICAL_EXT

4.3.2.4.2 SM-DS TLS Keys and Certificate

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)

File name	Description
CERT_S_SM_DS_TLS_INV_CRITICAL_EXT.der	CERT.DS.TLS certificate of the S_SM-DS with one of the critical extensions not present, formatted as X.509 certificate.

Table 57: DS TLS Certificate missing critical extension

4.3.2.4.3 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DS.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DS_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DS_TLS_INV_CRITICAL_EXT.ext.cnf` as defined in Annex A.

4.3.2.5 TLS – Invalid Extended Key Usage

4.3.2.5.1 SM-DP+ TLS Certificate: definition of data to be signed

Field	Value
version	Same as in section 3.5.2.1
serialNumber	Same as in section 3.5.2.1
signature	Same as in section 3.5.2.1
issuer	Same as in section 3.5.2.1
validity	Same as in section 3.5.2.1
subject	Same as in section 3.5.2.1
subjectPublicKeyInfo	Same as in section 3.5.2.1
Extensions	Same as in section 3.5.2.1
Extension for authorityKeyIdentifier	Same as in section 3.5.2.1
Extension for subjectKeyIdentifier	Same as in section 3.5.2.1
Extension for keyUsage	Same as in section 3.5.2.1

Field	Value
Extension for certificatePolicies	Same as in section 3.5.2.1
Extension for extendedKeyUsage	Critical TLS Client Authentication
Extension for subjectAltName	Same as in section 3.5.2.1
Extension for crlDistributionPoints	Same as in section 3.5.2.1

Table 58: CERT_S_SM_DS_TLS_INV_EXT_KEY_USAGE

4.3.2.5.2 SM-DS TLS Keys and Certificate

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.der	CERT.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 59: DS TLS Certificate with invalid 'extended key usage'

4.3.2.5.3 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DS.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DS_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DS_TLS_INV_EXT_KEY_USAGE.ext.cnf` as defined in Annex A.

4.3.2.6 TLS – Invalid Key Usage

4.3.2.6.1 SM-DS TLS Certificate: definition of data to be signed

Field	Value
version	Same as in section 3.5.2.1
serialNumber	Same as in section 3.5.2.1
signature	Same as in section 3.5.2.1
Issuer	Same as in section 3.5.2.1
Validity	Same as in section 3.5.2.1
Subject	Same as in section 3.5.2.1
subjectPublicKeyInfo	Same as in section 3.5.2.1
Extensions	Same as in section 3.5.2.1
Extension for authorityKeyIdentifier	Same as in section 3.5.2.1
Extension for subjectKeyIdentifier	Same as in section 3.5.2.1
Extension for keyUsage	Critical 'keyAgreement' ('08')
Extension for certificatePolicies	Same as in section 3.5.2.1
Extension for extendedKeyUsage	Same as in section 3.5.2.1
Extension for subjectAltName	Same as in section 3.5.2.1
Extension for crlDistributionPoints	Same as in section 3.5.2.1

Table 60: CERT_S_SM_DS_TLS_INV_KEY_USAGE

4.3.2.6.2 SM-DS TLS Keys and Certificate

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.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 61: DS TLS Certificate with invalid 'key usage'

4.3.2.6.3 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DP.TLS is generated using the command lines described in section 2.4 with the following input data:

<input_csr_file_name>: `CERT_S_SM_DS_TLS.csr.cnf` as defined in Annex A.

<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 3.4.3.1 for `serialNumber` data field.

<days> set with value defined in section 3.4.3.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DS_TLS_INV_KEY_USAGE.ext.cnf` as defined in Annex A.

4.3.2.7 TLS – Expired Certificate

4.3.2.7.1 SM-DS TLS Certificate: definition of data to be signed

Field	Value
version	Same as in section 3.5.2.1
serialNumber	Same as in section 3.5.2.1
signature	Same as in section 3.5.2.1
issuer	Same as in section 3.5.2.1
validity	expired on 2nd April 2020
subject	Same as in section 3.5.2.1
subjectPublicKeyInfo	Same as in section 3.5.2.1
Extensions	Same as in section 3.5.2.1
Extension for authorityKeyIdentifier	Same as in section 3.5.2.1
Extension for subjectKeyIdentifier	Same as in section 3.5.2.1
Extension for keyUsage	Same as in section 3.5.2.1
Extension for certificatePolicies	Same as in section 3.5.2.1
Extension for extendedKeyUsage	Same as in section 3.5.2.1

Field	Value
Extension for subjectAltName	Same as in section 3.5.2.1
Extension for crlDistributionPoints	Same as in section 3.5.2.1

Table 62: CERT_S_SM_DS_TLS_EXPIRED

4.3.2.7.2 SM-DS TLS Keys and Certificate

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.der	Expired CERT.DS.TLS certificate of the S_SM-DS with a valid signature, correctly formatted as X.509 certificate.

Table 63: DS TLS keys and expired Certificate

4.3.2.7.3 Input data for generation

The Private and Public Keys are generated using the command lines as described in section 2.2.

The CERT.DS.TLS is generated using the command lines described in section 2.4 with the following changes:

<input_csr_file_name>: `CERT_S_SM_DS_TLS.csr.cnf` as defined in Annex A.

<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 3.5.2.1 for `serialNumber` data field.

<days> set with value defined in section 4.3.2.7.1 for `validity` data field.

<cert_ext_file_name>: `CERT_S_SM_DS_TLS.ext.cnf` as defined in Annex A.

Annex A RSP Certificates and Keys Files (Normative)

All certificates, keys and configuration files are provided within the SGP.26_v1.3_Files.ZIP package which accompanies the present document.

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.ECDSA) 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.ECDSA) 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.DPauth.ECDSA and Test CERT.DPpb.ECDSA to be provisioned onto a Test SM-DP+ platform,
- The Test CERT.DP.TLS to be provisioned onto a Test SM-DP+ platform,
- The Test CERT.EUM.ECDSA and CERT.EUICC.ECDSA certificates to be provisioned onto the Test eUICCs.

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	2
serialNumber	**Vendor-specific**
signature	sha256ECDSA
Issuer	See 'subject'
Validity	**Vendor-specific**
Subject	**Vendor-specific**
subjectPublicKeyInfo	algorithm.algorithm='1.2.840.10045.2.1' (id-ecPublicKey) algorithm.parameters '1.2.840.10045.3.1.7' (prime256v1) or '1.3.36.3.3.2.8.1.1.7' (brainpoolP256r1) subjectPublicKey=[CI public key value]
Extension	(Sequence)
subjectKeyIdentifier extension	NIST: **Vendor-specific** Brainpool: **Vendor-specific**
keyUsage Extension	Certificate Signing, Off-line CRL Signing, CRL Signing (06)
certificatePolicies Extension	'2.23.146.1.2.1.0' (id-rspRole-ci)
basicConstraints Extension	CA = true

Field	Value
subjectAltName Extension	**Vendor-specific**
crlDistributionPoints Extension	**Vendor-specific**

Table 64: Self-Signed CERT.CI.ECDSA

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 Document Management

E.1 Document History

Version	Date	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

Other Information

Type	Description
Document Owner	eSIMG
Editor / Company	Alejandro Pulido / 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.