**3GPP TSG-SA3 Meeting #103-e *draft\_S3-212264-r2***

**e-meeting, 17th - 28th May 2021** *was S3-212066*

|  |
| --- |
| *CR-Form-v12.0* |
| **CHANGE REQUEST** |
|  |
|  | **33.310** | **CR** | Draft | **rev** | **-** | **Current version:** | **16.7.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps | **X** | ME | **X** | Radio Access Network |  | Core Network | **X** |

|  |
| --- |
|  |
| ***Title:***  | Security updates for algorithms and protocols in 33.310 |
| ***3*** |  |
| ***Source to WG:*** | Ericsson |
| ***Source to TSG:*** | S3 |
|  |  |
| ***Work item code:*** | eCryptPr |  | ***Date:*** | 2021-05-10 |
|  |  |  |  |  |
| ***Category:*** | B |  | ***Release:*** | Rel-17 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)Rel-12 (Release 12)**Rel-13 (Release 13)Rel-14 (Release 14)Rel-15 (Release 15)Rel-16 (Release 16)* |
|  |  |
| ***Reason for change:*** | - 33.310 already specifies OSCP certificate extentions and OCSP stapling is should support in TLS. However a 3GPP profile for OCSP is missing.- OCSP missing from definitions and abbreviations.- Several parts of the specification does not only mandate support of CRL, but also use of CRL. This limits flexibility in modern deployments that might want to use a mixture of CRL, OCSP, and OCSP stapling.- CA certificates typically use a higher security level than end-entity certificates. The certificate profile mandates support of RSA-4096 but does not mandate support of larger keys for ECDSA even though ECDSA is recommended. The same apply for CRL.- The assumption that a operator domain cannot include 100s TLS entities does not hold anymore.- ISAKMP (IKEv1) is mentioned but IKEv1 has not been allowed in 3GPP since Rel-11. |
|  |  |
| ***Summary of change:*** | - OCSP profile added.- OCSP added to definition and abbreviations.- While CRL is still kept mandatory to support, 33.310 is updated so that deployments are allowed to use OCSP.- Assumption that a operator domain cannot include 100s TLS entities removed.- Mention of ISAKMP removed |
|  |  |
| ***Consequences if not approved:*** | - As OCSP extentions are part of the certificate profiles but a profile is missing there is a risk that OCSP implementation does not fulfill 3GPP security requirements.- - Mandating use of CRL limits flexibility in modern deployments that might want to use a mixture of CRL, OCSP, and OCSP stapling. |
|  |  |
| ***Clauses affected:*** | 3, 5, , 6.3, 7.1, 7.4, 7.5, 7.6, 9.5.1 |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **X** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

\*\*\* BEGIN CHANGES \*\*\*

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the definitions given in TR 21.905 [8] and the following definitions apply:

**CA:** "Certification Authority", a PKI entity issuing X.509 certificates

**Interconnection CA:** The CA that issues cross-certificates on behalf of a particular operator to the SEG CAs of other domains with which the operator’s SEGs have interconnection.

**Interconnect Agreement**: In the context of this specification an interconnect agreement is an agreement by two operators to establish secure communications. This may be for the purpose of protecting various forms of communications between the operators, e.g. GPRS roaming, MMS interconnect, WLAN roaming and IMS interconnect.

**Local CR:** Repository that contains cross-certificates.

**Local CRL:** Repository that contains cross-certificate revocations.

**OSCP**: Online Certificate Status Protocol. Protocol for revocation checking which is can also be used offline in so called “OCSP stapling”. Can be used instead of CRL or together with CRL.

**PSK**: Pre-Shared Key. Method of authentication used by IKE between SEG in NDS/IP [1].

**Public CRL:** Repository that contains revocations of SEG and CA certificates and can be accessed by other operators.

**RA**: "Registration Authority", an optional PKI entity that does not issue certificates and is separate from the CA.

NOTE: An RA is delegated by a CA to receive and evaluate certificate signing requests, potentially verify them, and forward them to the CA which will issue an X.509 certificate.

**RA/CA:** The PKI entity or entities in the operator network issuing certificates, and making them available to base stations via CMPv2.

NOTE: If used in context of receiving certificate signing requests from a base station, the term may mean RA. If used in context of issuing certificates, the term means CA.

**SEG CA:** The CA that issues end entity certificates to SEGs within a particular operator’s domain.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [8] and the following abbreviations apply:

AF Authentication Framework

CA Certification Authority

CR Certificate Repository

CRL Certificate Revocation List

GBA Generic Bootstrapping Architecture

IMS IP Multimedia Subsystem

NDS Network Domain Security

OCSP Online Certificate Status Protocol

PKI Public Key Infrastructure

POP Proof Of Possession

PSK Pre-Shared Key

RA Registration Authority

SEG Security Gateway

VPN Virtual Private Network

Za Interface between SEGs belonging to different networks/security domains (a Za interface may be an intra or an inter operator interface).

Zb Interface between SEGs and NEs and interface between NEs within the same network/security domain

\*\*\* NEXT CHANGE \*\*\*

# 5 Architecture and use cases of the NDS/AF

The following types of certification authority are defined:

- SEG CA: A CA that issues end entity certificates to SEGs within a particular operator's domain.

- NE CA: A CA that issues end entity IPsec certificates to NE's within a particular operator's domain. Certificates issued by an NE CA shall be restricted to the Zb-interface.

- TLS client CA: A CA that issues end entity TLS client certificates to TLS entities within a particular operator's domain.

- TLS server CA: A CA that issues end entity TLS server certificates to TLS entities within a particular operator's domain.

- Interconnection CA: A CA that issues cross-certificates on behalf of a particular operator to the SEG CAs, TLS client CAs and TLS server CAs of other domains with which the operator's SEGs and TLS entities have interconnection.

The public key of the interconnection CA shall be stored securely in each SEG and TLS entity within the operator's domain. This allows the SEG and TLS entity to verify cross-certificates issued by its operator's Interconnection CA.

An operator may choose to combine two or more of the above CAs. For example, the same CA may be used to issue end entity TLS and IPsec certificates. Furthermore, the same CA may be used to issue both end entity certificates and cross-certificates.

The NDS/AF is initially based on a simple trust model (see Annex B) that avoids the introduction of transitive trust and/or additional authorisation information. The simple trust model implies manual cross-certification.

## 5.1 PKI architecture for NDS/AF

This chapter defines the PKI architecture for the NDS/AF. The goal is to define a flexible, yet simple architecture, which is easily interoperable with other implementations.

The architecture described below uses a simple access control method, i.e. every element which is authenticated is also provided service. More fine-grained access control may be implemented, but it is out of scope of this specification.

The architecture does not rely on bridge CAs, but instead uses direct cross-certifications between the security domains. This enables easy policy configurations in the SEGs and TLS entities.

### 5.1.1 General architecture

Unless the operator chooses to combine CAs, each security domain has at least one SEG CA, NE CA, TLS client CA or TLS server CA, and one Interconnection CA dedicated to it.

The SEG CA of the domain issues certificates to the SEGs in the domain that have interconnection with SEGs in other domains i.e. Za-interface. The SEG certificate can be used also in communication with an NE over the Zb-interface. An NE CA issues certificates to NE's for communication between NEs and between NE and SEGs within the responsible domain i.e. Zb interface. The TLS client CA of the domain issues certificates to the TLS clients in that domain that need to establish TLS connections with TLS servers in other domains. The TLS server CA of the domain issues certificates to the TLS servers in that domain that need to establish TLS connections with TLS clients in other domains. The Interconnection CA of the domain issues certificates to the SEG CAs, TLS client CA or TLS server CA, of other domains with which the operator’s SEGs and TLS entities have interconnection. This specification describes the profile for the various certificates that are needed. Also a method for creating the cross-certificates is described.

In general, all of the certificates shall be based on the Internet X.509 certificate profile [14].

#### 5.1.1.1 NDS/IP case

In the following, the architecture for issuing IPsec certificates using SEG CAs is described.

The SEG CA shall issue certificates for SEGs that implement the Za interface. When SEG of the security domain A establishes a secure connection with the SEG of the domain B, they shall be able to authenticate each other. The mutual authentication is checked using the certificates the SEG CAs issued for the SEGs. When an interconnect agreement is established between the domains, the Interconnection CA cross-certifies the SEG CA of the peer operator. The created cross-certificates need only to be configured locally to each domain. The cross-certificate, which Interconnection CA of security domain A created for the SEG CA of security domain B, shall be available for the domain A SEG which provides the Za interface towards domain B. Equally the corresponding certificate, which the Interconnection CA of the security domain B created for the SEG CA of security domain A, shall be available for the domain B SEG which provides Za interface towards domain A.

The general architecture for IPsec certificate based authentication of SEGs and NEs is illustrated in Figure 2.

NOTE 1: A potential NE CAA has not been depicted in the Figure 2, in order not to overload it.



Figure 2: Trust validation path in the context of NDS/IP

After cross-certification, the SEGa is able to verify the path: SEGb -> SEG CAB -> Interconnection CAA. Only the certificate of the Interconnection CAA in domain A needs to be trusted by entities in security domain A.

Equally the SEGb is able to verify the path: SEGa -> SEG CAA -> Interconnection CAB. The path is verifiable in domain B, because the path terminates to a trusted certificate (Interconnection CAB of the security domain B in this case).

The Interconnection CA signs the second certificate in the path. For example, in domain A, the certificate for SEG CA B is signed by the Interconnection CA of domain A when the cross-certification is done.

#### 5.1.1.2 TLS case

In the following, the architecture for issuing TLS certificates using TLS CAs is described.

The TLS client CA shall issue certificates for TLS clients in its domain. Similarly the TLS server CA shall issue certificates for TLS servers in its domain. When a TLS entity of the security domain A establishes a secure connection with a TLS entity of the domain B, they shall be able to authenticate each other. The mutual authentication is checked using the certificates the TLS client/server CAs issued for the TLS entities. When an interconnect agreement is established between the domains, the Interconnection CA cross-certifies the TLS client/server CAs of the peer operator. The created cross-certificates need only to be configured locally to each domain. The cross-certificate, which Interconnection CA of security domain A created for the TLS client/server CAs of security domain B, shall be available for the domain A TLS entities which need to communicate with domain B. Equally the corresponding certificate, which the Interconnection CA of the security domain B created for the TLS client/server CAs of security domain A, shall be available for the domain B TLS entities which need to communicate with domain A.

The general architecture for authentication of TLS entities is illustrated in Figure 2a.



Figure 2a: Trust validation path in the context of TLS

After cross-certification, the TLS client A is able to verify the path: TLS server B -> TLS server CAB -> Interconnection CAA. Only the certificate of the Interconnection CAA in domain A needs to be trusted by entities in security domain A.

Equally the TLS server B is able to verify the path: TLS client A -> TLS client CAA -> Interconnection CAB. The path is verifiable in domain B, because the path terminates to a trusted certificate (Interconnection CAB of the security domain B in this case).

The Interconnection CA signs the second certificate in the path. For example, in domain A, the certificates for TLS server CA B and TLS client CA B are signed by the Interconnection CA of domain A when the cross-certification is done.

## 5.2 Use cases

### 5.2.1 Operator Registration: Creation of interconnect agreement

SEGs or TLS entities of two different security domains need to establish a secure connection, when the operators make an interconnect agreement. The first technical step in creating the interconnect agreement between domains is the creation of cross-certificates by the Interconnection CAs of the two domains.

Inter-operator cross-certification can be done using different protocols, but the certification authority shall support the PKCS#10 method for certificate requests as specified in RFC 2986 [2]. The SEG CA, TLS client CA and TLS server CA create a PKCS#10 certificate request, and send it to the other operator's Interconnection CA. The method for transferring the PKCS#10 request is not specified, but the transfer method shall be secure. The PKCS#10 can be transferred e.g. HTTPS, in a flash drive, or be send in a signed email. The PKCS#10 request contains the public key of the authority and the name of the authority requesting the cross-certificate. When the Interconnection CA accepts the request, a new cross-certificate is created for the requesting CA. The Interconnection CA shall make the new cross-certificate available to SEGs and TLS entities in its own domain that need to use it. Cross-certificates on the other domain's SEG CA's are stored in a local CR (Certificate Repository) which all SEGs that need to communicate with the other domains shall access using LDAP as specified in RFC 2252  [5]. Cross-certificates on TLS client CAs and TLS server CAs are made available to TLS entities, e.g. by storing them in a file of trusted CAs on the TLS entity, or by storing them in a local CR (Certificate Repository) which all TLS entities that need to communicate with the other domain shall access e.g. using LDAP as specified in RFC 2252  [5].

The cross-certification is a manual operation, and thus PKCS#10 is a suitable solution for the interconnect agreement.

Creation of an interconnect agreement only involves use of the private keys of the Interconnection CAs. There is no need for the operators to use the private keys of their respective SEG CAs, TLS client CAs or TLS server CAs in forming an interconnect agreement.

When creating the new cross-certificate, the Interconnection CA should use basic constraint extension (according to section 4.2.1.9 of RFC 5280 [14]) and set the path length to zero. This inhibits the new cross-certificate to be used in signing new CA certificates. The validity of the certificate should be set sufficiently long. The cross-certification process needs to be done again when the validity of the cross-certificate is ending.

When the new cross-certificate is available to the SEG, all that needs to be configured in the SEG is the DNS name or IP address of the peering SEG gateway. The authentication can be done based on the created cross-certificates.

When the new cross-certificate is available to a TLS entity, it allows that TLS entity to authenticate TLS entities in the peering network. Authentication is done based on the created cross-certificates.

The certificate hierarchy in the case of two peering operators is illustrated in Figure 3.



Figure 3: Certificate Hierarchy

### 5.2.2 Establishment of secure communications

#### 5.2.2.1 NDS/IP case

#### 5.2.2.1.1 NDS/IP case for the Za interface

After establishing an interconnect agreement and finishing the required preliminary certificate management operations as specified in clause 5.2.1, the operators configure their SEGs for SEG-SEG connection, and the SAs are established as specified by NDS/IP [1].

In each connection configuration, the remote SEG DNS name or IP address is specified. Only the local Interconnection CA and SEG CA are configured as trusted CAs. Because of the cross-certification, any operator whose SEG CA has been cross-certified can get access using this VPN connection configuration.

The following is the flow of connection negotiation from the point of view of Operator A's SEG (initiator). Operator B's SEG (responder) shall behave in a similar fashion. In case of any failure in following steps, SEG A will treat this as an error and abort the procedure.

- During connection initiation, the initiating Operator A's SEG A provides its own SEG certificate and the corresponding digital signature in the IKE\_AUTH exchange for IKEv2;

- SEG A receives the remote SEG B certificate and signature;

- SEG A verifies the remote SEG B signature;

- SEG A checks the validity of the SEG B certificate by a revocation check to Operator B’s CRL databases or OCSP server. If a SEG cannot successfully perform the revocation check, it shall treat this as an error and abort tunnel establishment;

- SEG A verifies the SEG B certificate by executing the following actions:

- SEG A fetches the cross-certificate for Operator B's SEG CA from Operator A's Certificate Repository or from a local cache.

- SEG A checks the validity of the cross-certificate for Operator B's SEG CA by a revocation check to Operator A's Interconnection CA CRL database or OCSP server. If a SEG cannot successfully perform the revocation check, it shall treat this as an error and abort tunnel establishment;

- SEG A verifies the cross-certificate for Operator B's SEG CA using Operator A's Interconnection CA's certificate. Operator A's Interconnection CA's certificate shall be verified if the Interconnection CA is not a top-level CA, otherwise the Interconnection CA's public key is implicitly trusted.

- SEG A verifies the SEG B certificate using cross-certificate for Operator B’s SEG CA.

When IKEv2 has been initiated, then the IKE\_AUTH exchange is now completed. Now the IKEv2 CREATE\_CHILD\_SA exchange can be initiated as described in NDS/IP [1] with PSK authentication.

NOTE: This specification provides authentication of SEGs in an "end-to-end" fashion as regards to interconnect traffic (operator to operator). If NDS/AF (IKE) authentication were to be used for both access to the transport network (e.g. GRX) and for the end-to-end interconnect traffic, IPsec mechanisms and policies such as iterated tunnels or hop-by-hop security would need to be used. However, it is highlighted that the authentication framework specified is independent of the underlying IP transport network.

#### 5.2.2.1.2 NDS/IP case for the Zb-interface

In this case there is no need for cross-certification. Both end entity certificates belong to the same administrative domain and thus authorization check resolves to the same top level CA.

The following is the flow of connection negotiation from the point of view of NE-A (initiator). NE-B (or SEG-B) from the same domain (responder) shall behave in a similar fashion. In case of any failure in following steps, NE A will treat this as an error and abort the procedure.

- During connection initiation, the initiating Operator A's NE-A provides its own NE certificate and the corresponding digital signature in the IKE\_AUTH exchange for IKEv2;

- NE A receives the NE B (or SEG B) certificate and signature;

- NE A verifies the NE B (or SEG B) signature;

- NE A checks the validity of the NE B (or SEG B) certificate by a revocation check to the CRL databases or OCSP server of the same domain. If a NE cannot successfully perform the CRL check, it shall treat this as an error and abort tunnel establishment;

- NE A verifies the NE B (or SEG B) certificate using Operator NE CA certificate.

When IKEv2 has been initiated, then the IKE\_AUTH exchange is now completed. Now the IKEv2 CREATE\_CHILD\_SA exchange can be initiated as described in NDS/IP [1] with PSK authentication.

#### 5.2.2.2 TLS case

After establishing a interconnect agreement and finishing the required preliminary certificate management operations as specified in clause 5.2.1, the operators configure their TLS entities for secure interconnection. The exact process for establishing the TLS connections is dependent on the application protocol and is outside the scope of this specification. However, the general flow is described in the remainder of this clause.

The local Interconnection CA and TLS client/server CAs are configured as trusted CAs in the TLS entity typically by storing them in a file of trusted CAs on the TLS entity. The cross-certificates on the TLS client/server CAs of the remote operator are also made available to the TLS entity, e.g. by storing them in a file of trusted CAs on the TLS entity, or by storing them in a local CR (Certificate Repository) which all TLS entities that need to communicate with the other domain shall access e.g. using LDAP. Because of the cross-certification, any operator whose TLS client CA or TLS server CA has been cross-certified by another operator can establish TLS connections with that other operator.

The following is the connection establishment from the point of view of a TLS client in Operator A (TLSa) and a TLS server in Operator B (TLSb). The case where the TLS client is in Operator B and the TLS server is in Operator A is treated in a similar fashion. The flow is based on the TLS handshake protocol as described in RFC 8446 [49]. In case of any failure in following steps, TLSa or TLSb will treat this as an error and abort the procedure.

- During connection initiation, the TLSa sends a ClientHello message to TLSb. TLSb responds with a ServerHello message followed by a Certificate message, an optional CertificateRequest message, and other additional messages depending on the TLS version and options. The Certificate message will contain TLSb's certificate (or certificate chain)that was issued by Operator B's TLS server CA. The CertificateRequest message is sent if TLSb wants to authenticate TLSa using certificates in TLS, TLSa may otherwise be authenticated at a later stage using the application layer.

- TLSa receives the messages from TLSb

- TLSa verifies the received TLS messages using TLSb's public key

- TLSa checks the validity of TLSb's certificate by a revocation check to Operator B’s CRL databases or OCSP server. If a TLS peer cannot successfully perform the revocation check, it shall treat this as an error and abort the TLS handshake

- TLSa verifies TLSb's certificate using the cross-certificate for Operator B's TLS server CA by executing the following actions:

- TLSa fetches the cross-certificate for Operator B's TLS server CA from Operator A's Certificate Repository, from a local cache of the Certificate Repository on TLSa, or from a local certificate store on TLSa if a separate Certificate Repository is not used.

- TLSa checks the validity of the cross-certificate for Operator B's TLS server CA by a revocation check to Operator A's Interconnection CA CRL database or OCSP server. If a TLS peer cannot successfully perform the revocation check, it shall treat this as an error and abort the TLS handshake;

- TLSa verifies the cross-certificate for Operator B's TLS server CA using Operator A's Interconnection CA's certificate if the Interconnection CA is not a top-level CA, otherwise the Interconnection CA's public key is implicitly trusted.

- TLSa verifies TLSb’s certificate using the cross-certificate for Operator B’s TLS server CA.

- If TLSb requested a certificate using the CertificateRequest message, then TLSa responds with a Certificate message followed by a CertificateVerify message and a Finished message. The Certificate and CertificateVerify messages are only sent if the Server requests a certificate. If present, the Certificate message will contain TLSa's certificate (or certificate chain) that was issued by Operator A's TLS client CA. The CertificateVerify message is used to provide explicit verification of a client certificate.

- TLSb receives the messages from TLSa.

- If TLSb requested a certificate using the CertificateRequest message, then TLSb verifies the CertificateVerify message using TLSa’s public key.

- If TLSb requested a certificate using the CertificateRequest message, then TLSb checks the validity of TLSa's certificate by a revocation check to Operator A's CRL databases or OCSP server. If a TLS entity cannot successfully perform both revocation checks, it shall treat this as an error and abort the TLS handshake.

- If TLSb requested a certificate using the CertificateRequest message, then TLSb validates TLSa's certificate using the cross-certificate for Operator A's TLS client CA by executing the following actions:

- TLSb fetches the cross-certificate for Operator A's TLS client CA from Operator B's Certificate Repository, from a local cache of the Certificate Repository on TLSb, or from a local certificate store on TLSb if a separate Certificate Repository is not used.

- TLSb checks the validity of the cross-certificate for Operator A's TLS client CA by a revocation check to Operator B's Interconnection CA CRL database or OCSP server. If a TLS entity cannot successfully perform the revocation check, it shall treat this as an error and abort the TLS handshake

- TLSb verifies the cross-certificate for Operator A's TLS client CA using Operator B's Interconnection CA's certificate if the Interconnection CA is not a top-level CA, otherwise the Interconnection CA's public key is implicitly trusted.

- TLSb verifies TLSa’s certificate using the cross-certificate for Operator A’s TLS client CA.

When both Finished messages has been sent, then the secure communications can take place over the TLS connection.

### 5.2.3 Operator deregistration: Termination of interconnect agreement

When an interconnect agreement is terminated or due to an urgent service termination need, all concerned SEG peers shall remove the IPsec SAs using device-specific management methods, while all concerned TLS entities shall terminate any ongoing TLS sessions with the peer network and not permit those sessions to be resumed (e.g. by prohibiting TLS session resumption).

Each concerned operator shall also list the cross-certificate created for the Interconnection CA, SEG CA, TLS client CA and TLS server CA of the terminated operator in his own local CRL or OCSP server.

### 5.2.3a Interconnection CA registration

In principle only one Interconnection CA shall be used within the operator's network, but using more than one Interconnection CA is possible (in which case the public keys of all the operator’s interconnection CAs should be installed in the operator’s SEGs or TLS entities). The involved actions in Interconnection CA registration are those as described in the cross-certification part of clause 5.2.1: 'Operator Registration: creation of interconnect agreement'. Such a situation may exist if the Interconnection CA functions are to be moved from one responsible organisation to another (e.g. outsourcing of CA services).

### 5.2.3b Interconnection CA deregistration

If an Interconnection CA is removed from the network, it shall be assured that all certificates that have been issued by that CA to SEG or TLS CAs, and have not expired yet, shall be listed in the CRLs or OCSP servers.

### 5.2.3c Interconnection CA certification creation

The Interconnection CA certificate may not be the top-level CA of the operator, which means that the Interconnection CA certificate is not self-signed. If the Interconnection CA certificate is self-signed then it needs to be securely transferred to each SEG or TLS entity and stored within secure memory otherwise it can be managed in the same way as a SEG or TLS entity certificate.

The Interconnection CA certificate shall have a 'longer' lifetime than SEG CA or TLS CA certificates in order to avoid the cross-certification actions that are needed each time an Interconnection CA certificate has to be renewed.

NOTE: There is no need to involve other operators when creating an Interconnection CA certificate.

### 5.2.3d Interconnection CA certification revocation

If an Interconnection CA key pair gets compromised then a hacker could use the keys to issue himself SEG CA or TLS CA certificates which in turn could be used to issue SEG or TLS entity certificates. Since however the trusted Interconnection CA certificates are stored locally on the SEG or TLS entity device or in a dedicated repository (i.e. received Interconnection CA certificates within the IKE payload or TLS handshake shall not be accepted), the hacker also needs to compromise the SEG, TLS entity, or the local repository to be able to set up a secure connection.

Existing secure connections need not be torn down. The old cross-certificates - and any other certificates - issued by the Interconnection CA shall be taken out of service by listing them in the Interconnection CA’s CRL or OCSP server (provided the operator still has the key available to sign) and removing them from the dedicated repository. If the Interconnection CA certificate is self-signed then it shall be removed from each of the operator’s SEGs and TLS entities. If the Interconnection CA certificate is issued by a higher level CA of the operator, then it shall be revoked by this higher level CA.

The operator has to create a new Interconnection CA key pair, perform the actions as described within clause 5.2.3c for Interconnection CA certification creation, and perform the actions as described within clause 5.2.1 to generate new cross-certificates for all his interconnected networks SEG CAs or TLS CAs.

NOTE: There is no need to involve other operators when revoking an Interconnection CA certificate.

### 5.2.3e Interconnection CA certification renewal

The Interconnection CA certificate has to be renewed before the old Interconnection CA certificate expires. The renewing of an Interconnection CA certificate involves repeating the actions as described in clause 5.2.3c. This should be done before the old certificate expires.

NOTE: There is no need to involve other operators when renewing an Interconnection CA certificate.

### 5.2.4 SEG/TLS CA registration

In principle only one SEG CA, one TLS client CA and one TLS server CA shall be used within the operator's network, but using more than one of each of these CAs is possible. The involved actions are those as described in the cross-certification part of clause 5.2.1: 'Operator Registration: creation of interconnect agreement'. Such a situation of having multiple CAs of each type may exist if the CA functions are to be moved from one responsible organisation to another (e.g. outsourcing of CA services).

### 5.2.5 SEG/TLS CA deregistration

If a SEG CA or TLS CA is removed from the network, it shall be assured that the SEG CA or TLS CA certificates and all certificates that have been issued by the SEG CA or TLS CA to SEGs or TLS entities, and have not expired yet, shall be listed in CRLs or OCSP servers. The cross-certificates that are issued to these SEG CAs or TLS CAs, and have not expired yet, should also be listed in CRLs and OCSP servers.

### 5.2.6 SEG/TLS CA certificate creation

The involved actions are those as described in the cross-certification part of clause 5.2.1: 'Operator Registration: creation of interconnect agreement'.

The SEG CA or TLS CA certificate does not have to be the top-level CA of the operator, which means that the SEG CA or TLS CA certificate is not self-signed. One option is to sign the operator's SEG CA and TLS CAs with the operator’s own Interconnection CA, as this will already be a trust point established in the operator's own SEGs and TLS entities. If the SEG CA or TLS CA certificates are self-signed then they should be securely transferred to each of the operator's SEGs and TLS entities and stored within secure memory (see NOTE to clause 7.5).

### 5.2.7 SEG/TLS CA certificate revocation

This compromise is a serious event as it will require all the cross-certificates issued by other operators' Interconnection CAs to that SEG CA or TLS CA to be revoked.

Existing secure connections need not be torn down, unless they were formed very recently i.e. after the time at which the operator suspects the CA key became compromised, but before the cross-certificate used to establish the tunnel was revoked.

It shall be assured that the SEG CA or TLS CA certificates and all certificates that have been issued by the SEG CA or TLS CA to SEGs or TLS entities, and have not expired yet, shall be listed in CRLs or OCSP servers. The cross-certificates that are issued to these SEG CAs or TLS CAs, and have not expired yet, should also be listed in CRLs or OCSP servers.

To restore inter-domain interoperability, the operator has to create a new SEG CA or TLS CA key pair and use it to issue certificates to all the SEGs and TLS entities in the operator’s own domain. The operator shall then provide a cross-certification request (see clause 5.2.1) for the new SEG CA or TLS CA key pair to the operators with whom it has interconnect agreements.

It is recommended that operators carefully protect their SEG CA and TLS CA keys to limit this knock-on effect across the operator community.

### 5.2.8 SEG/TLS CA certificate renewal

The SEG CA and TLS CA certificate has to be renewed before the old SEG CA and TLS CA certificate expires. The renewing of a SEG CA or TLS CA certificate involves repeating the actions as described in the cross-certification part of clause 5.2.1: 'Operator Registration: creation of interconnect agreement'. This should be done before the old certificate expires.

### 5.2.9 End entity registration

#### 5.2.9.1 SEG registration

If not already done, a SEG certificate has to be created (see clause 5.2.11 for a description on certificate creation).

If a SEG is added to the network, the policy database of this SEG has to be configured using device-specific management methods.

Other operators have to be informed of the new SEG: The SEG policy databases of SEGs in other networks may have to be adapted.

#### 5.2.9.2 TLS client registration

If not already done, a TLS client certificate has to be created (see clause 5.2.11 for a description on certificate creation).

If a TLS client is added to the network, then some local configuration may be needed to take the new TLS client into use for secure inter-operator communication. In addition, other operators may need to be informed of the new TLS client.

#### 5.2.9.3 TLS server registration

If not already done, a TLS server certificate has to be created (see clause 5.2.11 for a description on certificate creation).

If a TLS server is added to the network, then some local configuration may be needed to take the new TLS server into use for secure inter-operator communication. In addition, other operators may need to be informed of the new TLS server.

#### 5.2.9.4 NE registration

If not already done, an NE certificate has to be created (see clause 5.2.11 for a description on certificate creation).

If an NE is added to the network, the policy database of this NE has to be configured using device-specific management methods.

### 5.2.10 End entity deregistration

#### 5.2.10.1 SEG deregistration

If a SEG is removed from the network, the SAs shall be removed using device-specific management methods. The operator of the SEG shall have the certificate of the SEG listed in his CRL or OCSP server. The SPD of the partner network may have to be adapted.

#### 5.2.10.2 TLS client deregistration

If a TLS client is removed from the network, the TLS connections shall be terminated using device-specific management methods. The operator of the TLS client shall have the certificate of the TLS client listed in his CRL or OCSP server.

#### 5.2.10.3 TLS server deregistration

If a TLS server is removed from the network, the TLS connections shall be terminated using device-specific management methods. The operator of the TLS server shall have the certificate of the TLS server listed in his CRL or OCSP server.

#### 5.2.10.4 NE deregistration

If a NE is removed from the network, the SAs shall be removed using device-specific management methods. The operator of the NE shall have the certificate of the NE listed in his CRL or OCSP server.

### 5.2.11 End entity certificate creation

Using device-specific management methods, the certificate creation shall be initiated. As specified in section 7.2, either the CMPv2 protocol for automatic certificate enrolment or manual certificate installation using PKCS#10 formats can be used. This is an operator decision depending for example on the number of NEs or SEGs and TLS entities.

### 5.2.12 End entity certificate revocation

If a SEG or TLS entity key pair gets compromised then the existing SAs shall be removed using device-specific management methods. The operator of the SEG or TLS entity shall include the revoked certificate in his CRL or OCSP server.

### 5.2.13 End entity certificate renewal

A new NE, SEG or TLS entity certificate needs to be in place before the old certificate expires. The procedure is similar to the certificate creation and can be either fully automated by using CMPv2 as specified in section 7.2 or done manually using PKCS#10 formats. This is an operator decision depending for example on the number of NEs, SEGs and TLS entities.

### 5.2.14 NE CA deregistration

If an NE CA is removed from the network, it shall be assured that the NE CA certificate and all certificates that have been issued by the NE CA to the NEs, and have not expired yet, shall be listed in CRLs or OCSP servers.

### 5.2.15 NE CA certification creation

The NE CA certificate does not have to be the top-level CA of the operator, which means that the NE CA certificate is not self-signed. If the NE CA certificates are self-signed then they should be securely transferred to each of the operator's NEs and stored within secure memory (see NOTE to clause 7.5).

NOTE: There is no need to involve other operators when creating an NE CA certificate.

### 5.2.16 NE CA certificate revocation

This serious event will require that all NE certificates needs to be revoked.

Existing intra-security domain security connections need not be torn down, unless they were formed very recently i.e. after the time at which the operator suspects the NE CA key became compromised but before the certificate has been listed as revoked.

It shall be assured that the NE CA certificate and all certificates that have been issued by the NE CA to NEs, and have not expired yet, shall be listed in CRLs or OCSP servers.

To restore intra-domain security, the operator has to create a new NE CA key pair and use it to issue certificates to all the NEs in the operator’s own domain.

NOTE: There is no need to involve other operators when revoking an NE CA certificate.

### 5.2.17 NE CA certificate renewal

The NE CA certificate has to be renewed before the old NE CA certificate expires.

NOTE: There is no need to involve other operators when renewing an NE CA certificate.

\*\*\* NEXT CHANGE \*\*\*

## 6.1 Certificate profiles

NOTE: The present clause contains the general 3GPP certificate profile. Other 3GPP specifications (e.g. TS 33.203 [9], TS 33.220 [10], etc.) point to the present clause. Thus parts of the present clause may also apply to devices and network nodes as specified in other specifications. New specifications using certificates should refer to this profile with as few exceptions as possible.

The present clause profiles the certificates to be used for NDS/AF. An NDS/AF component shall not expect any specific behaviour from other entities, based on certificate fields not specified in this section.

Certificate profiling requirements as contained in this specification have to be applied in addition to those contained within RFC5280 [14]. In case of conflicting requirements, the requirements in this specification override and obsolete the requirements in RFC5280 [14]. This applies for the SEG, NE, the TLS entity, the SEG CA and the Interconnection CA.

A receiving SEG or TLS entity shall be able to process an extension marked as critical in the present document.

Before fulfilling any certificate signing request, the NE CA, SEG CA and Interconnection CA shall make sure that the request suits the profiles defined in this section. Furthermore, the CAs shall check the Subject's DirectoryString order for consistency, and that the Subject's DirectoryString belongs to its own administrative domain.

NEs, SEGs and TLS entities shall check compliance of certificates with the NDS/AF profiles and shall only accept compliant certificates.

### 6.1.1 Common rules to all certificates

- Version 3 certificate according to RFC5280 [14].

- Hash algorithm for use before signing certificate: SHA-256 shall be supported, SHA-384 should be supported, MD5, MD2, and SHA-1 shall not be supported.

NOTE 1: Void.

- Signature algorithm: RSAEncryption and ecdsa shall be supported. RSAEncryption is not recommended as it uses PKCS#1v1.5 padding.

- Public key algorithm: rsaEncryption and id-ecPublicKey shall be supported.

 - Parameters: For ecdsa and id-ecPublicKey, secp256r1 shall be supported. secp384r1 should be supported.

- ECDSA is recommended for newly created certificates.

- For RSA certificates: The public key length shall be at least 2048-bit. A public key length of at least 4096-bit shall be supported. Public key lengths of less than 2048-bit shall not be supported. PKCS#1v1.5 padding and key lengths less than 3072-bits should not be used in certificates that expire after 2030. RSA public exponent shall be no less than 65537.

- For ECDSA certificates: Except curve25519, ed25519, and W-25519, elliptic curve groups of less than 256 bits shall not be supported. A public key length of at least 384-bit shall be supported.

NOTE 2: Void.

NOTE 3: In practice, certificates often have a long lifetime, for example about ten years. The use of RSA with PKCS#1v1.5 padding and key lengths less than 3072-bits is planned to be prohibited by several organisations no later than 2030.

- The security level of the public key used to sign the certificate shall be at least the same as the public keys in the certificate.

- Subject and issuer name format.

- (C=<country>), O=<Organization Name>, CN=<Some distinguishing name>. Organization and CN shall be in UTF8 format. Note that C is optional element.

or

- cn=<hostname>, (ou=<servers>), dc=<domain>, dc=<domain>. Note that ou is optional element.

- CRLs as specified in subclause 6.1a shall be supported for certificate revocation verification.

- OCSP as specified in subclause 6.1b should be supported for certificate revocation verification.

- Certificate extensions which are not mandated by this specification but which are mentioned within RFC5280 [14] are optional for implementation. If present, such optional extensions shall be marked as “non critical“.

NOTE 3: The above requirement implies that an NE, SEG or TLS entity receiving such optional extensions marked as “critical” will react with an error because, according to the introduction to clause 6.1 of the present document, NEs, SEGs and TLS entities shall only accept compliant certificates.

### 6.1.2 Interconnection CA Certificate profile

In addition to clause 6.1.1, the following requirements apply:

- Extensions:

- Optionally non critical authority key identifier;

- Optionally non critical subject key identifier;

- Mandatory critical key usage: At least keyCertSign and cRLSign should be asserted;

- Mandatory critical basic constraints: CA=True, path length unlimited or at least 1.

### 6.1.3 SEG Certificate profile

SEG certificates shall be directly signed by the SEG CA in the operator domain that the SEG belongs to. Any SEG shall use exactly one certificate to identify itself within the NDS/AF.

In addition to clause 6.1.1 and the provisions of RFC4945 [15], the following requirements apply:

- Issuer name is the same as the subject name in the SEG CA certificate.

- Extensions:

- Optionally non critical authority key identifier;

- Optionally non critical subject key identifier;

- Mandatory non-critical subjectAltName;

- Mandatory critical key usage: At least digitalSignature or nonRepudiation bits shall be set;

- Mandatory non-critical Distribution points: CRL distribution point;

NOTE: Depending on the availability of DNS between peer SEGs, the following rule is applied:

- subjectAltName should contain IP address (in case DNS is not available);

- subjectAltName should contain FQDN (in case DNS is available).

### 6.1.3a TLS entity certificate profile

TLS client certificates shall be directly signed by the TLS client CA in the operator domain that the TLS client belongs to. TLS server certificates shall be directly signed by the TLS server CA in the operator domain that the TLS server belongs to.

In addition to clause 6.1.1, the following requirements apply:

- For SIP domain certificates, the recommendations in RFC 5922 [21] and RFC 5924 [22] should be followed.

- Issuer name is the same as the subject name in the TLS CA certificate.

- Extensions:

- Optionally non critical authority key identifier;

- Optionally non critical subject key identifier;

- Mandatory critical key usage: At least digitalSignature or keyEncipherment shall be set; According to RFC 8446 [49] keyAgreement shall be set on Diffie-Hellman certificates;

- Optional non-critical extended key usage: If present, at least id-kp-serverAuth shall be set for TLS server certificates, and at least id-kp-clientAuth shall be set for TLS client certificates;

- Mandatory non-critical Distribution points: CRL distribution point.

### 6.1.3b NE Certificate profile

NE certificates shall be directly signed by the NE CA in the operator domain that the NE belongs to. Any NE shall use exactly one certificate to identify itself within the NDS/AF.

The same requirements as listed in section 6.1.3 apply.

### 6.1.3c SBA Certificate profile

#### 6.1.3c.1 Introduction

Clause 6.1.3c profiles the certificates to be used for 5GC Service Based Architecture (SBA).

Different TLS entity certificate profile requirements may be applied to intra-domain and/or inter-domain SBA for NF producers, NF consumers and NRF instances, and Security Edge Protection Proxy (SEPP) nodes applicable to 3GPP 5GC roaming.

A separate TLS entity certificate profile is also needed to cover the usage of the certificates issued by the SEPP CA(s) for inter-domain SBA context for TLS connections between SEPP nodes.

Furthermore, separate TLS entity certificate profile requirements may be applied forService Communication Proxy (SeCoP) needed for 3GPP 5GC SBA Indirect Communication model architectural Options C and D.

#### 6.1.3c.2 General SBA Certificate profile

The following additions and deviations to the common profiles shall hold for all SBA-related entities (NFs, SECOPs, SEPPs):

- Signature algorithm: RSAEncryption need not be supported.

- ECDSA is recommended for TLS entity certificates with 5GC Service Based Architecture (SBA).

#### 6.1.3c.3 NF Certificate profile

TLS certificates shall be directly signed by the CA in the operator domain that the entity belongs to.

NOTE: RFC 6125 [52] describes guidelines and procedures for representing and verifying the identity of application service using X.509 PKIX certificates with TLS. It mandates use of subjectAltName entries (DNS-ID, SRV-ID, URI-ID, etc.) over use of the subject field (CN-ID) where available. Furthermore, it is stated that a client does not seek a match for a reference identifier of CN-ID if the presented identifiers include a DNS-ID, SRV-ID, URI-ID, or any application-specific identifier types supported by the client.

In addition to clause 6.1.1 and the provisions of RFC 5280 [14] the following table captures the certificate profile for NF:

Table 6.1.3c.3-1: NF TLS Client and Server Certificate Profile

|  |
| --- |
| NF TLS Client and Server Certificate Profile |
| Version | v3 |
| Serial Number | Unique Positive Integer in the context of the issuing Root CA and not longer than 20 octets. |
| Subject DN | C=<Country>O= Home Domain Name (e.g., in "5gc.mnc<MNC>.mcc<MCC>.3gppnetwork.org" format) as defined in clause 28.2 of TS 23.003 [55]) |
| Validity Period | 3 years or less |
| Signature  | See clause 6.1.1 for the list of supported signature algorithms. |
| Subject Public Key Info  | See clause 6.1.1 for the list of supported public key types. |
| Extensions | OID | Mandatory | Criticality | Value |
| keyUsage | {id-ce 15} | TRUE | TRUE | digitalSignature for TLS clients and servers |
| keyEncipherment for TLS 1.2 [54] serversNF that may be both TLS 1.2 [54] client and server shall have both flags set. |
| extendedKeyUsage | {id-ce 37} | TRUE | FALSE | id-kp-clientAuth TLS clients  |
| id-kp-serverAuth for TLS servers NF that may be both client and server shall have both OIDs set. |
| authorityKeyIdentifier | {id-ce 35} | TRUE | FALSE | This shall be the same as subjectKeyIdentifier of the Issuer’s certificate. CA shall utilitize the method (1) as defined in clause 4.2.1.2 of RFC 5280 [14] to generate the value for this extension. |
| subjectKeyIdentifier | {id-ce 14} | FALSE | FALSE | This shall be calculated by the issuing CA utilitizing the method (1) as defined in clause 4.2.1.2 of RFC 5280 [14] to generate the value for this extension. |
| cRLDistributionPoint | {id-ce 31} | TRUE | FALSE | distributionPointAc cording to RFC 5280 [14] this indicates if the CRL is available for retrieval using access protocol and location with LDAP or HTTP URI. |
| subjectAltName | {id-ce 17} | TRUE | TRUE | Multiple subjectAltName entries can be used as a sequence, see below for the detailed instructions. |
| authorityInfoAccess | {id-pe 1} | FALSE | FALSE | id-ad-caIssuersAccording to RFC 5280 [14] id-ad-caIssuers describes the referenced description server and the access protocol and location, for example, using one or multiple HTTP and/or LDAP URIs.  |
| id-ad-ocspAccording to RFC 5280 [14] id-ad-ocsp defines the location of the OCSP responder using HTTP URI. |
| TLS feature extension | {id-pe 24} | FALSE | FALSE | id-pe-tlsfeatureThis can be used according to RFC 7633 [53] to prevent downgrade attacks that are not otherwise prevented by the TLS protocol; also to be used with OCSP stapling with TLS server end-entity certificates. |

With (intra-domain) SBA, the following rules are applied:

- subjectAltName should (in TLS client and server certificates) contain a URI-ID with the URI for the NF Instance ID as an URN; this URI-ID shall contain the nfInstanceID of the Network Function instance using the format of the NFInstanceId as described in clause 5.3.2 of TS 29.571 [57].

NOTE 1: Since the format of the NF instance ID according to clause 5.3.2 of TS 29.571 [57] is a universally unique identifier (UUID), the URN formed using the UUID is the string "urn: uuid:" followed by a hexadecimal representation of the UUID. For example, "urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6" is the string representation of the NF Instance ID "f81d4fae-7dec-11d0-a765-00a0c91e6bf6" as a URN.

- subjectAltName should (in TLS server certificates) contain SRV-ID with the HTTPS URI(s) for the apiRoot of a Network Function producer instance for the NF service API(s) that it provides; using wildcard URIs should be avoided;

- subjectAltName should (in TLS server certificates) contain SRV-IDs with the HTTPS URI(s) for the apiRoot of a Network Function consumer instance for the NF service callback URI(s) that it provides; using wildcard URIs should be avoided;

- subjectAltName should (in TLS client certificates) or shall (for TLS server certificates) contain a DNS-ID with the FQDN (host DNS name) for the Network Function instance, for example, using the instructions for Network Function (host DNS) names in FQDN format as used for Network Function producers in NFProfile and/or in NFService profile according to clause 6.1.6.2 in TS 29.510 [56], and in general as described in clause 28.3 of TS 23.003 [55] (regardless if DNS is available or not); for AMF, this is the AMF Name as described in clause 28.3.2.5 of TS 23.003 [55]; for NRF, this is the NRF FQDN as described in clause 28.3.2.3.2 of TS 23.003 [55]; the rules for using wildcard certificates in DNS-ID are defined in RFC 6125 [51].

NOTE 2: RFC 7540 [50] mandates using the Server Name Indication (SNI) extension to TLS with HTTP/2. RFC 6066 [51], which is applicable to TLS 1.2, defines that currently only server names supported in SNI extension to TLS are DNS hostnames where "HostName" contains the fully qualified DNS hostname (FQDN) of the TLS server. RFC 6066 [51] also defines that literal IPv4 and IPv6 addresses are not permitted in "HostName". In practice, this means that at least one subjectAltName attribute with FQDN is to be included in server-side TLS end-entity certificates.

- subjectAltName should (in TLS client certificates) contain NF type as DNS-ID (that is, using dNSName subjectAltName) for the Network Function instance using the Enumerated NF Type format according to clause 6.1.6.3.3 of TS 29.510 [56].

NOTE: If NF type is used in DNS-ID format in subjectAltName then it is considered as case-insensitive.

- subjectAltName shall not contain only IP address in TLS server certificates;

Editor’s Note: It is ffs whether subjectAltName contains URI for the NF Instance ID mandatory or optional in the TLS client and server certificates.

### 6.1.4 SEG CA certificate profile

In addition to clause 6.1.1, the following requirements apply:

- Subject name is the same as the issuer name in the SEG certificate;

- Issuer name depends on the usage of the certificates issued by the SEG CA:

- if used for interconnections between security domains with different root CAs the issuer name is the same as the subject name in the Interconnection CA certificate;

- if used for connections with elements having the same root CA certificate installed as used in the domain the SEG CA is located in, the issuer name is the subject name of either this root CA or an intermediate CA whose certificate has a valid certificate chain up to this root CA;

- Extensions:

- Optionally non critical authority key identifier;

- Optionally non critical subject key identifier;

- Mandatory critical key usage: At least keyCertSign and cRLSign, should be asserted;

- Mandatory critical basic constraints: CA=True, path length 0.

### 6.1.4a TLS client/server CA certificate profile

In addition to clause 6.1.1, the following requirements apply:

- Subject name is the same as the issuer name in the TLS entity certificate;

- Issuer name depends on the usage of the certificates issued by the TLS client/server CA:

- if used for interconnections between security domains with different root CAs the issuer name is the same as the subject name in the Interconnection CA certificate;

- if used for connections with elements having the same root CA certificate installed as used in the domain the TLS client/server CA is located in, the issuer name is the subject name of either this root CA or an intermediate CA whose certificate has a valid certificate chain up to this root CA;

- if used for TLS clients with certificates not issued by an operator CA, the issuer name is the subject name of either a root CA trusted by the operator or an intermediate CA whose certificate has a valid certificate chain up to a root CA trusted by the operator;

- Extensions:

- Optionally non critical authority key identifier;

- Optionally non critical subject key identifier;

- Mandatory critical key usage: At least keyCertSign and cRLSign, should be asserted;

- Mandatory critical basic constraints: CA=True, path length 0.

### 6.1.4b NE CA certificate profile

The same requirements as listed in section 6.1.4 apply except that there is no restriction in the issuer name.

## 6.1a CRL profile

- Version 2 CRL according to RFC5280 [14].

- Hash algorithm for use before signing CRL: SHA-256 shall be supported SHA-384 should be supported, MD5 MD2, and SHA-1 shall not be supported.

NOTE: Void.

- Signature algorithm: RSAEncryption and ecdsa shall be supported. RSAEncryption is not recommended as it uses PKCS#1v1.5 padding.

- Parameters: For ecdsa, secp256r1 shall be supported, secp384r1 should be supported.

- ECDSA is recommended for newly created CRLs.

- The security level of the public key used to sign the CRL shall be at least the same as the public keys used to sign the revoked certificates.

- For RSA: The key length shall be at least 2048-bit. A key length of at least 4096-bit shall be supported. Key lengths of less than 2048-bit shall not be supported. PKCS#1v1.5 padding and key lengths less than 3072-bits should not be used in certificates that expire after 2030.

- For ECDSA: Except curve25519, ed25519, and W-25519, elliptic curve groups of less than 256 bits shall not be supported. A key length of at least 384-bit shall be supported.

NOTE 1: In practice, certificates often have a long lifetime, for example about ten years. The use of RSA with PKCS#1v1.5 padding and key lengths less than 3072-bits is planned to be prohibited by several organisations no later than 2030.

CRL retrieval with LDAPv3 [5] shall be supported as the primary method. HTTP may be used for checking the revocation status of TLS and NE certificates.

## 6.1b OCSP profile

OCSP is protocol for obtaining the revocation status of an X.509 certificate. It can be used in addition to or instead of CRL. With OCSP stapling, a OSCP response is transported together with the certificate in the security protocol and can therefore be used also by offline nodes. The following requirements apply:

- Version 1 OCSP according to RFC6960 [47].

- Hash algorithm for use before signing OCSP response: SHA-256 and SHA-384 shall be supported, MD5 MD2, and SHA-1 shall not be supported.

- Signature algorithm: RSAEncryption and ecdsa shall be supported. RSAEncryption is not recommended as it uses PKCS#1v1.5 padding.

- Parameters: For ecdsa, secp256r1 and secp384r1 shall be supported.

- ECDSA is recommended for newly created OCSP servers.

- The security level of the public key used to sign OCSP shall be at least the same as the public keys used to sign the certificates.

- For RSA: The key length shall be at least 2048-bit. A key length of at least 4096-bit shall be supported. Key lengths of less than 2048-bit shall not be supported. PKCS#1v1.5 padding and key lengths less than 3072-bits should not be used in certificates that expire after 2030.

- For ECDSA: Except curve25519, ed25519, and W-25519, elliptic curve groups of less than 256 bits shall not be supported. A key length of at least 384-bit shall be supported.

NOTE 1: In practice, certificates often have a long lifetime, for example about ten years. The use of RSA with PKCS#1v1.5 padding and key lengths less than 3072-bits is planned to be prohibited by several organisations no later than 2030.

OCSP over HTTP shall be supported as the primary transport mechanism.

\*\*\* NEXT CHANGE \*\*\*

## 6.3 Path validation

### 6.3.1 Path validation profiling

- Validity of certificates received from the peer end entity shall be verified by CRLs or OCSP responses retrieved via the mechanisms specified in section 6.1.1, based on the CRL Distribution Point or Authority Information Access extentions in the certificates.

- Validity of certificates received from the TLS entity shall be verified by CRLs or OCSP responses retrieved via the mechanisms specified in section 6.1.1, based on the CRL Distribution Point or Authority Information Access extentions in the certificates.

- Any NE, SEG or TLS entity shall not validate received certificates from a peer entity whose validity time has expired, but end the path validation with a negative result.

- Any NE, SEG shall not validate received certificates from a peer entity whose CRL distribution point field is empty, but end the path validation with a negative result.

- Certificate validity calculation results shall not be cached in a SEGs or NEs for longer than the lifetime enforced by the end entity.

- Certificate validity calculation results shall not be cached in TLS entities for longer than the TLS connection lifetime.

\*\*\* NEXT CHANGE \*\*\*

## 7.1 Repositories

During secure connection establishment, each NE, SEG or TLS entity has to verify the validity of its peer's certificate according to clause  5.2.2. Any certificate could be invalid because it was revoked (and replaced by a new one) or a NE, SEG, TLS entity or operator has been deregistered.

Consider secure connection establishment between PeerA in network A and PeerB in network B.

PeerB has to verify that:

a) the cross-certificate of the PeerA's CAA is still valid;

b) the certificate of PeerA is still valid,

and be able to:

c) fetch the cross-certificate of PeerA CAA (if not found in PeerA 's cache or local store).

PeerA performs the same checks from its own perspective.

Check a) can be performed by querying the local CRL or OCSP server. For check b), a CRL or OCSP server of the PeerA's CA shall be queried. At this point of time, the secure connection is not yet available, therefore the public CRL or OCSP server of the PeerA's CA shall be accessible without relying on a secure connection.

Figure 4 and Figure 4a illustrate the repositories and the above-mentioned steps a) – c). The local Certificate Repository (CR) contains cross-certificates for SEG CAs and possibly cross-certificates for TLS CAs if these are not locally stored in the TLS entities. Local CRLs contains SEG CA and TLS CA cross-certificate revocations, and the public CRL or OCSP server contain revocations of SEG, TLS entity, SEG CA, and TLS CA certificates, and can be accessed by other operators.

An operator's internal repository may contain the revocations of NE and NE CA if not contained in the Public CRL or OCSP repository.



Figure 4: Repositories for NDS/IP to support Za interface

Figure 5: Repositories for TLS case



Figure 6: Repositories for NDS/IP to support Zb interface

If the SEG CA, TLS CA or Interconnection CA are combined then the public and local repositories of the CA may be implemented as separate databases or as a single database which is accessible via two different interfaces. Access to the "public" CRL or OCSP server is public with respect to the interconnecting transport network (e.g. GRX). The public CRL or OCSP should be adequately protected (e.g by a firewall) and the owner of the public CRL or OCSP server may limit access to it according to his interconnect agreements. Access to a public CRL database or OCSP server does not need to be secured.

NOTE 1: First it is not necessary to secure access to the CRL database or OCSP as the retrieved CRL or OCSP response is integrity protected and contains no confidential information. Secondly access via an unprotected interface is anyhow necessary in case no currently valid security association is available to access the public CRL database or OCSP server.

SEGs shall use LDAP to access the CRL and cross-certificate repositories. TLS entities shall use LDAP or HTTP to access the CRL repositories. TLS entities may use LDAP to access the cross-certificate repositories, if the cross certificates are not stored locally in the TLS entity. NE's may use LDAP or HTTP to access the CRL repositories. OCSP servers shall always be accessed via HTTP.

NOTE 2: Interfaces a) and c) for locating the data used to establish secure communications between operators belong to the scope of NDS/AF (in addition to public b) interface) as the purpose is to guarantee the interoperability between different SEGs, TLS entities and repository implementations. The possible migration to the cross-certification with a Bridge CA would also require these interfaces to be specified.

\*\*\* NEXT CHANGE \*\*\*

## 7.4 Revoking a SEG/TLS CA cross-certificate

The following procedure is used to revoke a SEG CA cross-certificate:

1. The cross-certificate is added into the Interconnection CA's CRL or OCSP server;

2. The cross-certificate is removed from the Interconnection CA's CR.

The following procedure is used to revoke a TLS CA cross-certificate:

1. The cross-certificate is added into the Interconnection CA's CRL or OCSP server;

2. If the TLS CA cross certificates are stored in the Interconnection CA's CR, then the cross-certificate is removed.

3. If the TLS CA cross-certificates are stored locally in the TLS entities, then the locally stored cross-certificates are deleted in the TLS entities.

\*\*\* NEXT CHANGE \*\*\*

## 7.5 Establishing secure connections between NDS/IP end entities using IKE on the Za interface

Certificate based authentication during the IKEv2 IKE\_INIT\_SA/IKE\_AUTH exchanges is shown in figure 4 above. The SEGa uses the following procedure to authenticate SEGb:

1. SEGa requests SEGb's certificate using the CERTREQ payload;

2. SEGa receives SEGb's certificate inside the CERT payload;

3. SEGa authenticates SEGb (verifies signatures);

4. SEGa performs a revocation check with CRL or OCSP to verify the status of SEGb's certificate. If the locally cached CRL has expired, SEGa fetches a CRL from the (public) CRL database of SEC CAb before using CRL;

5. SEGa uses either the locally cached cross-certificate or fetches the cross-certificate from the (local) Interconnection CAa CR to verify SEGb's certificate;

6. SEGa performs a revocation check with CRL or OCSP to verify the status of the SEG CA cross-certificate. If the locally cached CRL has expired, SEGa fetches a CRL from the (local) Interconnection CAa CRL database before using CRL;;

8. SEG A verifies the cross-certificate for Operator B's SEG CA using Operator A's Interconnection CA's certificate. SEGa verifies the status of the Interconnection CAa certificate if the Interconnection CAa is not a top-level CA, otherwise Interconnection CAa is implicitly trusted;

NOTE: If the local SEG CA public key is securely installed on every SEG within an operator's domain, then a cross-certificate does not need to be checked when SEGa and SEGb belong to the same operator's domain.

\*\*\* NEXT CHANGE \*\*\*

## 7.6 CRL management

NDS/AF compliant SEGs and NEs shall not send an IKEv2 CERTREQ where the Certificate Type is "Certificate Revocation List (CRL)". Receiving NEs and SEGs may ignore this request as section 6.1.3 specifies that CRLs shall be retrieved via a CRL distribution point.

The CRL issuer (which is in most cases the CA) shall only issue full CRLs. The use of delta CRLs is not allowed because of possible interoperability problems and because in the NDS/AF environment the full CRL is not expected to grow too large. The full CRL shall only contain revoked certificates applicable for use within NDS/AF. The CRL issuer shall issue a CRL also in cases that there are no revoked certificates. A SEG, NE or TLS entity is not obliged to query for a CRL via the CRL Distribution Point if a cached one is still available and valid. If no valid cached CRL is available, the NE, SEG or TLS entity shall fetch a new CRL. If no valid CRL can be fetched, the NE, SEG or TLS entity shall treat this as an error and cancel tunnel establishment.

\*\*\* NEXT CHANGE \*\*\*

### 9.5.1 General Requirements

The following requirements shall apply to CMPv2 usage end-to-end between base station and RA/CA:

- This CMPv2 profile shall only include certificate request and key update functions. Revocation processing and PKCS#10 requests shall not be part of this CMPv2 profile.

- For PKI Message Protection, this CMP profile shall only use an asymmetric algorithm. PasswordBasedMac is not used in the scope of the present document.

- The base station shall be pre-provisioned with a private/public key pair (vendor key pair) and with the related vendor base station certificate signed by a vendor CA.

- If there is a certificate chain from the base station certificate up to the vendor root CA, also the intermediate certificates shall be pre-provisioned to the base station.

- The base station may be pre-provisioned with the operator root CA certificate.

- If the base station is not pre-provisioned with the operator root CA certificate, then the base station shall take the operator root certificate from the certificates received in the initialization response. The selection shall be based on checking which root certificate can be used to validate the received base station certificate.

NOTE 1: Certificate renewal for operator root certificates is not in scope of this clause on base station enrolment. Thus it is assumed that the base station always has a valid operator root certificate available for validation of key update responses.

- The RA/CA shall authenticate initialization requests based on signatures which are validated against the vendor root CA.

- The RA/CA shall authenticate key update requests based on signatures which are validated against the operator root CA.

- The RA/CA shall be configured with the root certificate of the vendor and with the root certificate of the operator.

- The RA/CA shall be configured with a RA/CA certificate which is signed either by the operator root CA or by an intermediate CA under the operator root CA.

- If the RA/CA uses different private keys to sign the generated certificates and the CMPv2 messages, the RA/CA shall be configured with the two related certificates, i.e. the RA/CA certificate for signing signatures and the RA/CA certificate for signing CMP messages.

- If the RA/CA certificate or certificates (two in case separate private keys are used for signing of certificates and CMP messages) are not signed directly by the operator root CA, also the certificates of the intermediate CAs shall be configured into the RA/CA.

- The hash algorithms used before generating signatures in the protection field of PKIMessage and for proof-of-possession shall be the same as the hash algorithms specified in subclause 6.1.1 for certificate signatures. The signature algorithms shall be the same as that used in the related certificate profile.

The certificate profiles are specified in subclause 9.4.

NOTE 2: These certificate profiles implicitly specify which algorithms are to be used for the different signatures for proof-of-possession and PKIMessage signing specified in the following subclauses.

NOTE 3: Policies within RA/CA governing the generation and issuing of certificates are not in scope of the present document and left to operator decision.

\*\*\* END OF CHANGES \*\*\*