SMKI Repository Interface Design Specification

Contents

1 Introduction

1.1 Purpose and Scope

The SMKI Repository Interface Design Specification describes the functionality of the SMKI Repository Interface (SRI) as set out in Section L6.4 of the SEC, in order to:

a) specify the technical details of the SMKI Repository Interface; and

b) set out the protocols and technical standards that apply to the SMKI Repository Interface.

1.2 Target Response Times

For the purposes of supporting the measurement of Target Response Times in accordance with Sections L8.4 and L8.5 of the SEC, the terms “send” and “receipt” should interpreted as follows:

a) “receipt” means, in relation to a request submitted over a DCC Gateway Connection to obtain any document lodged in the SMKI Repository, the successful completion by DCC of the validation checks in relation to such a request, as set out in the SMKI Repository Interface Design Specification; ~~a request issued by a DCC Gateway Connection user via the SMKI Repository Interface to obtain any document lodged in the SMKI Repository;~~ and

b) “send” means, in relation to a document lodged in the SMKI Repository, the making available of an electronic copy of that document by the DCC Systems such that it is immediately available to be retrieved over a DCC Gateway Connection via the SMKI Repository Interface.~~making a document available to be retrieved by a DCC Gateway Connection user, in response to the receipt of a request.~~

For the purposes of requests issued other than by a DCC Gateway Connection, the terms “send” and “receipt” should be interpreted in accordance with the meaning as set out in the SMKI Interface Design Specification.

2 Interface Definition

The DCC shall make three interfaces, collectively referred to as the SMKI Repository Interface, available via a DCC Gateway Connection through which Parties, RDPs, the SMKI PMA and the Panel (or the Code Administrator acting on their behalf) may access the SMKI Repository:

a) a SMKI Repository Portal interface accessed via a web browser (as set out in section 2.1 of this document);

b) a SMKI Repository Web Service interface that can be accessed by a Party's or an RDP's systems (as set out in section 2.2, Appendix A and Appendix B of this document); and

c) a SSH File Transfer Protocol Interface (the "SFTP Interface") via an SFTP client (as set out in section 2.3 of this document).

Any DCC Gateway Connection user wishing to manage its credentials used to access the SMKI Repository Web Service interface or SFTP Interface shall ensure that it has access to the SMKI Repository Portal interface, which allows management of such credentials.

The means by which Parties, RDPs, the SMKI PMA, Panel (or the Code Administrator acting on their behalf) may access SMKI Repository content without a DCC Gateway Connection, are set out in section 2.6 of the SMKI Interface Design Specification.

The SMKI Code of Connection sets out the methods by which such persons may, communicate over the SMKI Repository interfaces, and the methods by which connections to the SMKI Repository interfaces are authenticated and communications taking place over them are secured.

The DCC shall ensure that the SMKI Repository Interface is available in line with Section L6.2 of the Code and shall notify Parties and RDPs in advance of any planned outages of the SMKI Repository Interface. The DCC shall ensure that failover between the Live and Disaster Recovery (DR) environments will be achieved using dynamic routing and Network Address Translation on the DCC Gateway. The SMKI Repository interfaces will be accessed via the same Universal Resource Location (URL) in the event of a DR invocation; the traffic for this URL will be routed to the DR site and presented to the DR servers completely transparently to the user.

2.1 SMKI Repository Portal Interface via DCC Gateway Connection

2.1.1 General Obligations

The DCC shall ensure that the SMKI Repository Portal interface enables DCC Gateway Connection users to access the SMKI Repository Portal for the purposes:

a) of viewing, querying and / or obtaining a copy of those documents lodged in the SMKI Repository as set out in section 2.1.3 of this document; and

b) of updating password and user profile details in respect of authentication to the SMKI Repository Portal interface.

The DCC shall ensure that the SMKI Repository Portal interface via DCC Gateway Connection:

a) uses the HTTPS protocol, secured by TLS 1.2 in line with the protocols set out in Appendix C of this document;

b) is accessed via Uniform Resource Locators (URLs) that are set out in the SMKI Repository User Guide;

c) uses Javascript, Cascading Style Sheets (CSS) and images;

d) is compliant with the W3C Web Content Accessibility Guidelines (v2) at "AA" level; and

e) is only accessible using a DCC Gateway Connection.

The process for obtaining a DCC Gateway Connection is detailed in Section H3 of the Code.

The DCC shall ensure that all Certificates, CRLs and ARLs lodged in the SMKI Repository are in Base64 DER format. CRL and ARL validity is as set out in Appendix B to the Code and the IKI Certificate Policy.

2.1.2 Establishing connection to the SMKI Repository Portal interface via DCC Gateway Connection

In order to establish a connection to the SMKI Repository Portal interface, a DCC Gateway Connection user shall:

a) ensure that their browsers have Javascript enabled;

b) verify the CA/Browser Forum server certificate presented by the SMKI Repository Portal, as described below and, if successfully verified by the browser, accept the certificate;

c) enter a username and password that has been issued for the purpose of authenticating the user to the SMKI Repository Portal interface; and

d) establish a TLS 1.2 session.

The DCC shall ensure that a username and initial password is provided as set out in the SMKI Registration Authority Policies and Procedures (SMKI RAPP). The DCC shall ensure that the initial password must be changed by the user upon first use, and maintenance of the username and password is detailed in the SMKI Repository Code of Connection.

The DCC shall ensure that users are provided with a profile page which will enable users to view and update their SMKI Repository Portal username and password and to update contact information, as set out in the SMKI Repository Code of Connection.

The DCC shall ensure that the SMKI Repository Portal interface presents to the user a x.509 v3 certificate that is recognised by the CA/Browser Forum for the purposes of allowing the DCC Gateway Connection user's systems to authenticate the server as part of establishing the TLS session.

The DCC shall ensure that the SMKI Repository Portal denies access where a user does not present a valid username and password for authentication.

2.1.3 Retrieval of SMKI Repository content

The DCC shall ensure that the SMKI Repository Portal interface enables DCC Gateway Connection users to search for and download via a web form the following files that are lodged in the SMKI Repository, where they have successfully established a secured TLS 1.2 connection to the SMKI Repository Portal interface (as set out in the SMKI Repository User Guide):

a) IKI Certificates and ICA Certificates;

b) Organisation Certificates and OCA Certificates;

c) Device Certificate and DCA Certificates;

d) the latest IKI CRL and the latest IKI ARL;

e) the latest Organisation CRL and the latest Organisation ARL; and

f) other documents lodged in the SMKI Repository.

2.2 SMKI Repository Web Service interface

2.2.1 General Obligations

The DCC shall ensure that the SMKI Repository Web Service interface enables DCC Gateway Connection users' systems to search for and obtain content lodged in the SMKI Repository, as set out in section 2.2.3, Appendix A and Appendix B of this document.

The DCC shall ensure that the SMKI Repository Web Service interface via DCC Gateway Connection:

a) uses the HTTPS protocol, secured by TLS 1.2 in line with the protocols set out in Appendix C of this document;

b) is accessed via Uniform Resource Locators (URLs) that are set out in the SMKI Repository User Guide;

c) uses Extensible Markup Language (XML) over POST for message requests and responses;

d) provides XML message responses which conform with the details set out in Appendix A of this document and the XML schema set out in Appendix B of this document;

e) conforms with the XML Schema set out in Appendix B for message requests and responses; and

f) is only accessible using a DCC Gateway Connection.

Prior to gaining access to the SMKI Repository Web Service interface, a DCC Gateway Connection user shall access the profile page on the SMKI Repository Portal in order to obtain its credentials for the SMKI Repository Web Service interface, as set out in the SMKI RAPP. The DCC shall ensure that the credentials for the SMKI Repository Web Service interface shall be in the form of an API Key, which is generated by the DCC and is a 15 character UTF-8 case insensitive string.

The DCC shall, in accordance with the SMKI RAPP, provide the DCC Gateway Connection user with a CA/Browser Forum recognised certificate authority root certificate and all corresponding issuing authority certificates, for the purposes of enabling server authentication of the SMKI Repository Web Service interface.

The DCC shall ensure that each API Key shall:

a) remain valid until manually replaced by the DCC Gateway Connection user using the SMKI Repository Portal interface; and

b) once replaced, be invalid for authentication to the SMKI Repository Web Service interface.

2.2.2 Establishing connection to the SMKI Repository Web Service interface via DCC Gateway Connection

In order to establish a connection to the SMKI Repository Web Service interface, a DCC Gateway Connection user shall submit a request to establish a secured TLS1.2 session which:

a) accesses a URL as set out in section 2.2.1 of this document; and

b) includes its API Key in the querystring, in order that the SMKI Repository Interface can authenticate the user before attempting to parse the XML request document in accordance with Appendix A and Appendix B of this document; and

c) configures its systems such that the TLS session renegotiation timeout is set to 5 minutes.

The DCC shall ensure that the SMKI Repository Web Service presents a x.509 v3 certificate that is recognised by the CA/Browser Forum referenced in section 2.2.1 of this document, for the purposes of allowing the DCC Gateway Connection user's client to authenticate the server as part of establishing the TLS session. The DCC Gateway Connection user shall verify the CA/Browser Forum certificate and, if successfully verified, accept the certificate.

The DCC shall ensure that access to the SMKI Repository Web Service interface is denied where the user does not present a valid API Key for authentication.

2.2.3 Retrieval of SMKI Repository content

The DCC shall ensure that the SMKI Repository Portal Web Service interface enables DCC Gateway Connection users to search for and download the following files that are lodged in the SMKI Repository, where they have successfully established a connection to the SMKI Repository Portal Web Service interface:

a) IKI Certificates and ICA Certificates;

b) Organisation Certificates and OCA Certificates;

c) Device Certificate and DCA Certificates;

d) the latest IKI CRL and the latest IKI ARL; and

e) the latest Organisation CRL and the latest Organisation ARL.

2.3 SSH File Transfer Protocol (SFTP) Interface

2.3.1 General Obligations

The DCC shall ensure that the SFTP Interface to the SMKI Repository enables DCC Gateway Connection users' systems to download Certificates, CRLs and ARLs lodged in the SMKI Repository, as set out this section and Appendix C of this document.

The DCC shall ensure that the SFTP Interface to the SMKI Repository via DCC Gateway Connection:

a) is accessed via Uniform Resource Locators (URLs) that are set out in the SMKI Repository User Guide;

b) is implemented in a standard format conforming to:

i. Secure Shell (SSH) protocol, in accordance with RFC 4251, RFC4252 and RFC 4253;

ii. RFC 4251, RFC 4252, RFC 4253 and RFC 959 (File Transfer Protocol) for the purposes of error handling;

iii. the Transport Layer will use encrypted communications using the AES (Advanced Encryption Standard) cipher (FIPS-197) with a 128-bit key length, in CBC mode (aes128-cbc);

iv. the Transport Layer will use MAC communications using hmac in accordance with RFC2104, combined with sha1 (hmac-sha1); and

v. the Transport Layer will use RAW DSS Keys in ssh-dss format;

c) is implemented such that Quality of Service constraints (rate-limiting) are applied to the download of files via the SFTP Interface to protect other aspects of the overall DCC service;

d) the SFTP Interface provides access to the files specified in section 2.3.32.3.3 of this document; and

e) is only accessible using a DCC Gateway Connection.

Prior to gaining access to the SFTP Interface to the SMKI Repository, the DCC shall provide a username and initial password to the DCC Gateway Connection user as part of the process as set out in the SMKI RAPP. The DCC shall ensure that the initial password to authenticate to the SFTP Interface shall be provided to the DCC Gateway Connection user via the profile page on the SMKI Repository Portal.

The DCC shall ensure that the initial password must be changed by the DCC Gateway Connection user upon first use, using the profile page on the SMKI Repository Portal. The DCC shall ensure that a password used to authenticate to the SFTP Interface may be changed by the DCC Gateway Connection user at any time, via the profile page on the SMKI Repository Portal. The DCC shall ensure that each password used to authenticate to the SFTP Interface shall remain valid until replaced by the DCC Gateway Connection user via the SMKI Repository Portal interface and shall be invalid thereafter.

The DCC shall provide the DCC Gateway Connection user with the DCC SSH public key, which shall be available for download/viewing in the Help and Support section of the SMKI Repository Portal.

2.3.2 Establishing connection to the SFTP Interface to the SMKI Repository

In order to establish a connection to the SFTP Interface, each DCC Gateway Connection user shall:

a) make use of a standard SFTP client that supports the configuration as detailed in section 2.3.1 of this document;

b) authenticate to the SFTP interface by using a valid combination of its username and password, in accordance with the ‘password’ method; and

c) verify that the DCC SSH public key, as provided by the DCC as set out in section 2.3.1 of this document, matches the details within the SMKI Repository Portal prior to using the SFTP Interface.

Details of how a DCC Gateway Connection user should configure its SFTP client are set out in the SMKI Repository User Guide.

2.3.3 Retrieval of SMKI Repository content

The DCC shall ensure that the SMKI Repository SFTP interface enables DCC Gateway Connection users' systems to download the following files that are lodged in the SMKI Repository, where they have successfully established a connection to the SFTP Interface:

a) a file in .gz format and having a name of form *SMKIKR_FULL_YYYY-MM-DD.xml.gz*, updated daily by the time set out in the SMKI Repository User Guide, containing:

   i. an XML file which complies with the SMKI Repository Web Service interface schema as set out in Appendix B of this document, having a name of the form SMKIKR_FULL_YYYY-MM-DD.xml and which contains Certificates, comprising OCA Certificates, DCA Certificates, ICA Certificates, Organisation Certificates, Device Certificates with a status of 'In-Use' and IKI Certificates.

b) seven files in .gz format and having names of the form *SMKIKR_DELTA_YYYY-MM-DD.xml.gz*, updated daily, each of which contains:

   i. an XML file which complies with the SMKI Repository Web Service interface schema as set out in Appendix B of this document, having a name of the form SMKIKR_DELTA_YYYY-MM-DD.xml and which contains Certificates comprising OCA Certificates, DCA Certificates, ICA Certificates, Organisation Certificates, Device Certificates and IKI Certificates Issued and lodged in the SMKI Repository during the preceding twenty four hours or whose Certificate status has change. This will enable the user to maintain a daily synchronised copy of the Certificates in the SMKI Repository. Each of the seven daily files will be available for 7 days from publication and shall then be removed by the DCC from the SMKI Repository.

c) a file with extension 'arl' that is the latest Organisation ARL;

d) a file with extension 'arl' that is the latest IKI ARL;

e) a file with extension 'crl' that is the latest Organisation CRL;

f) a file with extension 'crl' that is the latest IKI CRL;

g) a file in .gz format, updated as necessary, containing the base set of Organisation Certificates and OCA Certificates required to populate Device anchor slots prior to installation for the North Region; and

h) a file in .gz format, updated as necessary, containing the base set of Organisation Certificates and OCA Certificates required to populate Device anchor slots prior to installation for the Central Region and South Region.

The DCC shall ensure that SFTP files holding Certificates will be made available in .gz format, with all versions of .gz being supported. Each .gz file will contain a single XML file which complies with the XML schema as set out in Appendix B, containing individual Certificates, represented as Base64 encoded strings.

The DCC shall ensure that the Organisation Certificates and OCA Certificates contained within the two Device anchor slot Certificate files shall be the same, other than the Organisation Certificates required to populate the WAN provider Device anchor slot.

The DCC shall lodge a document in the SMKI Repository, which sets out details of which of the base set of Organisation Certificates and OCA Certificates may be placed in specific Device anchor slots.

Appendix A SMKI Repository Web Service interface specification

Response Codes

The DCC shall ensure that the following HTTP response codes are returned in the response to each attempted access to the SMKI Repository Web Service interface:

a) HTTP response code 200, where a Web Service request is successful;

b) HTTP response code 4xx (400-499) where there is an error that is anticipated;

c) HTTP response code 404, where a request is made but the User should not have access to the SMKI Repository Web Service interface; and

d) HTTP response code 5xx (500-599) for unanticipated error conditions.

Response codes are also replicated in the XML response body as **`ResponseCode`**, along with a human readable description as **`ResponseMessage`** as set out in the 'Service Specific Error Codes' sections within this Appendix A.

Audit References

The SRI will include an **`AuditReference`** entity in each response body. This is a globally unique reference for the request served, and can be considered both as a receipt reference and a diagnostic tool in relation to the investigation of problems encountered in using the SRI web service interfaces.

HTTP POST Certificate Search

The table immediately below sets out the way in which a user may search the SMKI Repository by means of the SMKI Repository Web Service interface.

Web Service URL	**`/services/certificatesearch`**
Required Parameters	apikey=*<SRI User's API Key>*
Example URL	https://site.name.com/services/certificatesearch?apikey=u3bg9gt38htd0j2
Required POST	UTF8 encoded XML v1.0 Request Document with a top level **`CertificateSearchRequest`** entity, containing search term entities from the following list: • **`CertificateSubjectName`** (23 character EUI-64 format in the case of it being the Unique Identifier for an organisation or other up to 23 character subject name in the case of it being the common name of a CA certificate) • **`CertificateSubjectAltName`** (23 character EUI-64 format) • **`CertificateSerial`** (Up to 50 character string) • **`CertificateStatus`** (1 character string) • **`PubDateRangeStart`** (yyyy-mm-dd) • **`PubDateRangeEnd`** (yyyy-mm-dd) • **`ExpDateRangeStart`** (yyyy-mm-dd) • **`ExpDateRangeEnd`** (yyyy-mm-dd)

	• **RevDateRangeStart** (yyyy-mm-dd) • **RevDateRangeEnd** (yyyy-mm-dd) • **InUseDateRangeStart** (yyyy-mm-dd) • **InUseDateRangeEnd** (yyyy-mm-dd) • **CertificateIssuer** (Up to 23 character string) • **CertificateRole** (Integer) • **ManufacturingFlag** (true/false) All search term entities are optional, but it is mandatory to provide either **CertificateSubjectName**, **CertificateSubjectAltName**~~, **CertificateIssuer**~~ or **CertificateSerial**

Example Request XML	`<?xml version="1.0" encoding="utf-8"?>`
	`<CertificateSearchRequest>`
	`<CertificateSubjectAltName>01-02-03-04-05-06-07-08</CertificateSubjectAltName>`
	`<PubDateRangeStart>2014-01-01</PubDateRangeStart>`
	`<PubDateRangeEnd>2018-01-01</PubDateRangeEnd>`
	`</CertificateSearchRequest>`

Response Format	UTF8 encoded XML v1.0 Response Document with a top level **CertificateSearchResponse** entity, containing the following entities: • **ResponseCode** (up to 3 character string) • **ResponseMessage** (up to 50 character string) • **AuditReference** (up to 20 character string) Followed by a variable number of **Result** entities (in the case of a successful response), each comprising: • **CertificateSerial** (Up to 50 character string) • **CertificateSubjectAltName** (23 character EUI-64 format) • **CertificateSubjectName** (23 character EUI-64 format or other up to 23 character subject name in the case of a CA certificate common name) • **CertificateStatus** (1 character string) • **CertificateRole** (Integer) • **CertificateUsage** (2 character string) • **ManufacturingFlag** (true/false)

Example Response XML	`<?xml version="1.0" encoding="utf-8"?>`
	`<CertificateSearchResponse>`
	`<ResponseCode>200</ResponseCode>`
	`<ResponseMessage>Success</ResponseMessage>`
	`<AuditReference>1234567890-abc123456</AuditReference>`
	`<Result>`
	`<CertificateSerial>00926EAABE07B701DF</CertificateSerial>`
	`<CertificateSubjectAltName>01-02-03-04-05-06-07-08</CertificateSubjectAltName>`
	`<CertificateStatus>I</CertificateStatus>`
	`<CertificateRole>2</CertificateRole>`
	`<CertificateUsage>DS</CertificateUsage>`
	`<ManufacturingFlag>false</ManufacturingFlag>`
	`</Result>`
	`<Result>`
	`<CertificateSerial>1234567890</CertificateSerial>`
	`<CertificateSubjectAltName>01-02-03-04-05-06-07-08</CertificateSubjectAltName>`

	<CertificateStatus>P</CertificateStatus> <CertificateRole>2</CertificateRole> <CertificateUsage>DS</CertificateUsage> <ManufacturingFlag>false</ManufacturingFlag> </Result> </CertificateSearchResponse>
Service Specific Error Codes	**401** = Invalid Search Parameters **402** = No Certificates Match Search Parameters
Notes	

Retrieve Certificate

The table immediately below sets out the way in which a user may retrieve a certificate from the SMKI Repository by means of the SMKI Repository Web Service interface.

Web Service URL	**/services/retrievecertificate**
Required Parameters	apikey=*<SRI User's API Key>*
Example URL	https://site.name.com/services/retrievecertificate?apikey=u3bg9gt38htd0j2
Required POST	UTF8 encoded XML v1.0 Request Document with a top level **CertificateDataRequest** entity, containing the following, mandatory, entity: • **CertificateSerial** (Up to 50 character string)
Example Request XML	<?xml version="1.0" encoding="utf-8"?> <CertificateDataRequest> <CertificateSerial>00926EAABE07B701DF</CertificateSerial> </CertificateDataRequest>
Response Format	UTF8 encoded XML v1.0 Response Document with a top level **CertificateSearchResponse** entity, containing the following entities: • **ResponseCode** (up to 3 character string) • **ResponseMessage** (up to 50 character string) • **AuditReference** (up to 20 character string) • **CertificateResponse** (in the case of a successful response) consisting of: o **CertificateSubjectName** (23 character EUI-64 format in the case of it being the Unique Identifier for an organisation or other up to 23 character subject name in the case of the common name of a CA certificate) o **CertificateSubjectAltName** (23 character string) o **CertificateSerial** (Up to 50 character string) o **CertificateStatus** (1 character string) o **CertificateBody** (Base64 representation of the DER encoded ASN.1 notated certificate data) o **CertificateRole** (Integer)

	○ **CertificateUsage** (2 character string) ○ **ManufacturingFlag** (true/false)
Example Response XML	<?xml version="1.0" encoding="utf-8"?> <CertificateDataResponse> <ResponseCode>200</ResponseCode> <ResponseMessage>Success</ResponseMessage> <AuditReference>1234567890-abc123456</AuditReference> <CertificateResponse> <CertificateSubjectAltName>01-02-03-04-05-06-07-08</CertificateAltName> <CertificateSerial>00926EAABE07B701DF</CertificateSerial> <CertificateStatus>I</CertificateStatus> <CertificateBody> *(base64 certificate data)* </CertificateBody> <CertificateRole>2</CertificateRole> <CertificateUsage>DS</CertificateUsage> <ManufacturingFlag>false</ManufacturingFlag> </CertificateResponse> </CertificateDataResponse>
Service Specific Error Codes	**401** = Invalid Input Parameters **402** = No Certificates Match Input Parameters **403** = No Matching Certificates Are Valid
Notes	Certificates are requested by serial number, and serial numbers are globally unique across the SMKI, therefore only one certificate (i.e. one CertificateResponse entity) will be returned. The certificatesearch service should be used to determine the serial number for the certificate required.

CRL and ARL Retrieval

The latest version of the IKI CRL, IKI ARL, Organisation CRL and Organisation ARL will be available from separate static URLs enabling the automation of the CRL or ARL download via the SMKI Repository Web Service interface. Informational text will be displayed on the Portal to inform the user how they may automate downloads of these files using URLs of the form /revocationlists/<common_name>?apikey=<user_api_key>. The actual URL will be detailed in the SMKI Repository User Guide.

Meaning of XML schema codes

The table immediately below sets out the meaning of codified elements with the XML schema for the SMKI Repository Web Service Interface, where such XML schema is as set out in Appendix B of this document.

XML Schema Element Name	Possible Values	Meaning
CertificateStatus	P I N E R	Pending In use Not In use Expired Revoked
ManufacturingFlag	true false	Can be used during manufacturing Cannot be used during manufacturing
CertificateUsage	DS KA CS	Digital Signing Key Agreement Certificate Signing
CertificateRole	0 1 2 3 4 5 6 7 127	Root Recovery Supplier Network Operator Access Control Broker Transitional CoS WAN Provider Issuing Authority Other

Appendix B SMKI Repository Web Service interface schema

This section specifies the XML schema that must be used for the SMKI Repository Web Service, as set out immediately below.

The DCC shall ensure that the version number of the SMKI Repository Web Service interface is contained within the XML schema, as set out below. Each user of the SMKI Repository Web Service interface shall ensure that the version number is included within:

a) the URL used to access the Repository Web Service interface;

b) the schema filename; and

c) XML requests and responses.

There will be a different end point for each version of the SMKI Repository Web Service interface. Different versions will be supported on separate URLs.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <!--
Version (string)
  -->
            <xsd:element name="Version">
                  <xsd:simpleType>
                   <xsd:restriction base="xsd:string">
                      <xsd:enumeration value="1.0"/>
                   </xsd:restriction>
                 </xsd:simpleType>
  <!--
  ResponseCode (up to 3 character string)
  -->
  <xs:simpleType name="ResponseCode">
            <xs:restriction base="xs:string">
                        <xs:maxLength value="3"/>
                        <xs:minLength value="1"/>
            </xs:restriction>
  </xs:simpleType>
  <!--
  ResponseMessage (up to 50 character string)
  -->
  <xs:simpleType name="ResponseMessage">
            <xs:restriction base="xs:string">
                        <xs:maxLength value="50"/>
                        <xs:minLength value="1"/>
            </xs:restriction>
  </xs:simpleType>
  <!--
  AuditReference (up to 20 character string)
  -->
  <xs:simpleType name="AuditReference">
            <xs:restriction base="xs:string">
                        <xs:maxLength value="20"/>
                        <xs:minLength value="1"/>
            </xs:restriction>
  </xs:simpleType>
<!--
  Status (1 character enum values: S = Success, F = Failure)
 -->
  <xs:simpleType name="Status">
            <xs:restriction base="xs:string">
                        <xs:enumeration value="S"/>
                        <xs:enumeration value="F"/>
            </xs:restriction>
  </xs:simpleType>
  <xs:element name="ResponseCode" type="ResponseCode"/>
  <xs:element name="ResponseMessage" type="ResponseMessage"/>
  <xs:element name="AuditReference" type="AuditReference"/>
!--
```

```
		CertificateSubjectName (23 character EUI-64 format)
-->
  <xs:simpleType name="CertificateSubjectName">
		<xs:restriction base="xs:string">
			<xs:maxLength value="23"/>
			<xs:minLength value="1"/>
		</xs:restriction>
  </xs:simpleType>
<!--
		CertificateSubjectAltName (23 character string)
-->
  <xs:simpleType name="CertificateSubjectAltName">
		<xs:restriction base="xs:string">
			<xs:maxLength value="23"/>
			<xs:minLength value="1"/>
		</xs:restriction>
  </xs:simpleType>
<!--
		CertificateStatus (1 character string)
-->
  <xs:simpleType name="CertificateStatus">
		<xs:restriction base="xs:string">
			<xs:enumeration value="P"/>
			<xs:enumeration value="I"/>
			<xs:enumeration value="N"/>
			<xs:enumeration value="E"/>
			<xs:enumeration value="R"/>
		</xs:restriction>
  </xs:simpleType>
<!--
		PubDateRangeStart (yyyy-mm-dd)
-->
  <xs:simpleType name="PubDateRangeStart">
     <xs:restriction base="xs:date"/>
  </xs:simpleType>
<!--
		PubDateRangeEnd (yyyy-mm-dd)
-->
  <xs:simpleType name="PubDateRangeEnd">
     <xs:restriction base="xs:date"/>
  </xs:simpleType>
<!--
		ExpDateRangeStart (yyyy-mm-dd)
-->
  <xs:simpleType name="ExpDateRangeStart">
     <xs:restriction base="xs:date"/>
  </xs:simpleType>
<!--
		ExpDateRangeEnd (yyyy-mm-dd)
-->
  <xs:simpleType name="ExpDateRangeEnd">
     <xs:restriction base="xs:date"/>
  </xs:simpleType>
<!--
		RevDateRangeStart (yyyy-mm-dd)
-->
  <xs:simpleType name="RevDateRangeStart">
     <xs:restriction base="xs:date"/>
  </xs:simpleType>
<!--
		RevDateRangeEnd (yyyy-mm-dd)
-->
  <xs:simpleType name="RevDateRangeEnd">
     <xs:restriction base="xs:date"/>
  </xs:simpleType>
<!--
		InUseDateRangeStart(yyyy-mm-dd)
-->
  <xs:simpleType name="InUseDateRangeStart">
     <xs:restriction base="xs:date"/>
  </xs:simpleType>
<!--
		InUseDateRangeEnd(yyyy-mm-dd)
-->
  <xs:simpleType name="InUseDateRangeEnd">
```

```
    <xs:restriction base="xs:date"/>
  </xs:simpleType>
<!--
            CertificateIssuer (23 character string)
-->
  <xs:simpleType name="CertificateIssuer">
            <xs:restriction base="xs:string">
                        <xs:maxLength value="23"/>
                        <xs:minLength value="1"/>
            </xs:restriction>
  </xs:simpleType>
<!--
  CertificateSerial (50 character string)
-->
  <xs:simpleType name="CertificateSerial">
            <xs:restriction base="xs:string">
                        <xs:maxLength value="50"/>
                        <xs:minLength value="1"/>
            </xs:restriction>
  </xs:simpleType>
<!--
  Result
-->
    <xs:complexType name="Result">
      <xs:sequence>
        <xs:element name="CertificateSerial" type="CertificateSerial" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CertificateSubjectAltName" type="CertificateSubjectAltName" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="CertificateSubjectName" type="CertificateSubjectName" minOccurs="0" maxOccurs="1"/>
        <xs:element name="CertificateStatus" type="CertificateStatus" minOccurs="1" maxOccurs="1"/>
              <xs:element name="CertificateRole" type="xs:integer" minOccurs="0" maxOccurs="1"/>
              <xs:element name="CertificateUsage" type="CertificateUsage" minOccurs="1" maxOccurs="1"/>
              <xs:element name="ManufacturingFlag" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
<!--
            CertificateResponse (in the case of a successful response) consisting of:
-->
    <xs:complexType name="CertificateResponse">
      <xs:sequence>
        <xs:element name="CertificateSubjectName" type="CertificateSubjectName" minOccurs="0" maxOccurs="1"/>
        <xs:element name="CertificateSubjectAltName" type="CertificateSubjectAltName" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="CertificateSerial" type="CertificateSerial" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CertificateStatus" type="CertificateStatus" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CertificateBody" type="xs:base64Binary" minOccurs="0" maxOccurs="1"/>
                        <xs:element name="CertificateRole" type="xs:integer" minOccurs="0" maxOccurs="1"/>
              <xs:element name="CertificateUsage" type="CertificateUsage" minOccurs="1" maxOccurs="1"/>
              <xs:element name="ManufacturingFlag" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
     </xs:sequence>
    </xs:complexType>

<!--
            CertificateDataRequest
-->
  <xs:complexType name="CertificateDataRequest">
    <xs:sequence>
                  <xs:element name="CertificateSerial" type="CertificateSerial" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
<!--
            CertificateUsage (2 character string)
-->
  <xs:simpleType name="CertificateUsage">
            <xs:restriction base="xs:string">
                        <xs:enumeration value="DS"/>
                        <xs:enumeration value="KA"/>
                        <xs:enumeration value="CS"/>
            </xs:restriction>
  </xs:simpleType>
<!--
            CertificateDataResponse
-->
  <xs:complexType name="CertificateDataResponse">
    <xs:sequence>
```

```
    <xs:element name="ResponseCode" type="ResponseCode" minOccurs="1" maxOccurs="1"/>
    <xs:element name="ResponseMessage" type="ResponseMessage" minOccurs="1" maxOccurs="1" />
    <xs:element name="AuditReference" type="AuditReference" minOccurs="1" maxOccurs="1"/>
    <xs:element name="CertificateResponse" type="CertificateResponse" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:complexType>
<!--
            CertificateSearchRequest. Note: At least one of CertificateSerial, CertificateSubjectName or
CertificateSubjectAltName must be supplied. This is not enforced by the XSD, but is enforced by the SMKI
Repository code.
-->
  <xs:complexType name="CertificateSearchRequest">
   <xs:sequence>
      <xs:element name="CertificateSerial" type="CertificateSerial" minOccurs="0" maxOccurs="1"/>
      <xs:element name="CertificateSubjectName" type="CertificateSubjectName" minOccurs="0" maxOccurs="1"/>
      <xs:element name="CertificateSubjectAltName" type="CertificateSubjectAltName" minOccurs="0"
maxOccurs="1"/>
      <xs:element name="CertificateStatus" type="CertificateStatus" minOccurs="0" maxOccurs="1"/>
      <xs:element name="PubDateRangeStart" type="PubDateRangeStart" minOccurs="0" maxOccurs="1"/>
      <xs:element name="PubDateRangeEnd" type="PubDateRangeEnd" minOccurs="0" maxOccurs="1"/>
      <xs:element name="ExpDateRangeStart" type="ExpDateRangeStart" minOccurs="0" maxOccurs="1"/>
      <xs:element name="ExpDateRangeEnd" type="ExpDateRangeEnd" minOccurs="0" maxOccurs="1"/>
      <xs:element name="RevDateRangeStart" type="RevDateRangeStart" minOccurs="0" maxOccurs="1"/>
      <xs:element name="RevDateRangeEnd" type="RevDateRangeEnd" minOccurs="0" maxOccurs="1"/>
      <xs:element name="InUseDateRangeStart" type="InUseDateRangeStart" minOccurs="0" maxOccurs="1"/>
      <xs:element name="InUseDateRangeEnd" type="InUseDateRangeEnd" minOccurs="0" maxOccurs="1"/>
            <xs:element name="CertificateIssuer" type="CertificateIssuer" minOccurs="0" maxOccurs="1"/>
            <xs:element name="CertificateRole" type="xs:integer" minOccurs="0" maxOccurs="1"/>
            <xs:element name="ManufacturingFlag" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
   </xs:sequence>
  </xs:complexType>
<!--
            CertificateSearchResponse
-->
  <xs:complexType name="CertificateSearchResponse">
   <xs:sequence>
     <xs:element name="ResponseCode" type="ResponseCode" minOccurs="1" maxOccurs="1"/>
     <xs:element name="ResponseMessage" type="ResponseMessage" minOccurs="1" maxOccurs="1"/>
     <xs:element name="AuditReference" type="AuditReference" minOccurs="1" maxOccurs="1"/>
     <xs:element name="Result" type="Result" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:complexType>
  <xs:element name="CertificateDataRequest" type="CertificateDataRequest"/>
  <xs:element name="CertificateDataResponse" type="CertificateDataResponse"/>
  <xs:element name="CertificateSearchRequest" type="CertificateSearchRequest"/>
  <xs:element name="CertificateSearchResponse" type="CertificateSearchResponse"/>
 </xs:schema>
```

Appendix C Authentication Credentials

The SMKI Repository Portal Interface via DCC Gateway Connection, SMKI Repository Web Service interface and SFTP interface shall use server certificates with the following properties:

Criteria	***Version***
Protocol	*TLS1.2**
Protocol Cyphers	*ECDHE-RSA-AES256-GCM-SHA384*
	ECDHE-RSA-AES128-GCM-SHA256
	ECDHE-RSA-AES128-SHA256
Client Certificate Key	*RSA 2048 bit*
Client Certificate Hash Algorithm	*SHA256*
Server Certificate Key	*RSA 2048 bit*
Server Certificate Hash Algorithm	*SHA256*

* TLS 1.2 should be implemented in accordance with Java and Apache standards. Java 7 and above supports TLS1.2. The TLS version is specified in the HTTP client protocol initialisation. To enable AES256, the Java runtime should be patched with "JCE Unlimited Strength Jurisdiction Policy Files" for the version of Java being used. This is obtained from the public Oracle Java download web pages.

Appendix D Definitions

Term	Meaning as defined in SEC
In-Use	Means a valid Certificate that has not been operationally superseded and which the device has acknowledged as being successful installed/updated