|  |
| --- |
| IALA Guideline on the provision of mcp identities |

Gnnnn

Edition 1.0

June 2024 (Council 80)

urn:mrn:iala:pub:g1128:ed.1.0

Revisions to this document are to be noted in the table prior to the issue of a revised document.

|  |  |  |
| --- | --- | --- |
| Date | Details | Approval |
| June 2024 | First Edition | Council 80 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

List of Tables

[1. introductioN 5](#_Toc157863074)

[2. Specification 5](#_Toc157863075)

[3. The role of the MCP Consortium 6](#_Toc157863076)

[4. Identity Management 7](#_Toc157863077)

[5. Public Key Infrastructure 9](#_Toc157863078)

**No table of figures entries found.**

List of Figures

*No table of figures entries found.Figure 1 Hierarchical X.509 PKI Structure* 9

*Figure 2 14*

# introductioN

The goal of this document is to define the requirements for providing and using secure identities by means of MCP Maritime Identity Registry (MIR). It is, thus, intended both for organisations that are planning to become MCP Identity Service Providers and organisations that intend to make and run applications and services that use MIR certificates to implement secure identities. The IALA recommendation for such secure identities is stated in IALA R1019 'PROVISION OF MARITIME SERVICES IN THE CONTEXT OF E-NAVIGATION IN THE DOMAIN OF IALA'.

## 1.1 Rationale

The prerequisite for the digitalisation of the maritime domain is a trustworthy provision of digital services for information exchange. For example, when a vessel approaches a port or waters controlled by a Vessel Traffic Services (VTS) centre, it is dependent on receiving information from them. However, it is not only important to receive the respective information, but also to verify from whom the respective information was sent and who the service is provided by. Otherwise, arbitrary participants could, for example, deliberately send out false information in order to disrupt the processes. In this case the recipient would not be able to differentiate which information is the original and which is the falsified information. To solve this problem, the respective participants need to be able to authenticate each other securely. In the paper world, authentication is done by a handwritten signature of the authorised person. In the digital world this is done by using digital certificates and signatures.

The MCP features - as one of its core components - an identity registry, where all entities that wish to exchange information are registered and have a digital certificate issued to them. Thus, a vessel registered with the MCP identity registry (having a digital certificate issued from it), can authenticate itself (cryptographically prove its identity) to the VTS centre, and thus provide data to the VTS centre which the VTS centre can trust the origin of. The principle of authentication is a cornerstone in contemporary digital solutions.

# Specification

There are three aspects of MCP identity provisioning:

1. **Identity Management**: A MIR enables that each maritime entity (such as a device, human, organization, service, ship, etc.) can be registered as a participant of the MCP and be equipped with a unique identifier. The identifier is given in terms of an MRN (Maritime Resource Name, IALA guideline G1164). While MIR governance harmonizes the MRN namespace governed by the MCC and sets out criteria for the registration process, it is up to the MIR services to implement and have certified concrete identity registries. We use the following terminology:
   * MCP entity: An entity registered at some MIR services.
   * MCP namespace: The subspace of the MRN namespace that is governed by the MCC.

See Chapter 4 Identity Management for details.

1. **Public Key Infrastructure (PKI):** The MIR enables that each MCP entity holds a cryptographic identity in terms of a public/private key pair and a certificate bound to their MRN identifier within the MCP. The cryptographic identity of a MCP entity will change over time (due to updates of key material), but the MRN identifier within the MCP will be persistent.

See Chapter 5 Public Key Infrastructure for details.

1. **Federation between identity providers:** For a distributed identity system to work a system must be in place that allows the federation of trust between identity providers at some level. This may mean attestation protocols or trust networks. This aspect will be defined in future versions.

# The role of the MCP Consortium

The MCP consortium (MCC) was established in 2019, with the aim to realise the MCP. The specifications of the MCP have later become IALA guidelines (including this document), except for the Maritime Messaging Service (MMS), which has become an RTCM standard leaving the MCC with the following responsibilities / activities:

1. Maintain procedures for endorsing MCP service providers using this IALA guideline. The endorsement aims at checking compliance with the guideline, but in addition it includes a minimal vetting procedure for organisations for which identities are being provided. A document describing the endorsement procedure can be found on the web-page of the consortium.
2. Endorse MCP Identity Service Providers.
3. Issue MCP MRN namespaces to MCP Identity Service Providers.
4. Maintain a (signed) list of root certificates of endorsed MCP identity service providers. This provides the means to identify MCP identities with some level of basic trust.

It is important to note, that the MCC is not a legal entity, and therefore, from a legal perspective, all activities of the MCC are performed by its members. Information about MCC, its activities, relevant documents and access to the public demonstrator can be found at www.maritimeconnectivity.net.

# Identity Management

The MCP namespace is a subspace of the *Maritime Resource Name (MRN)* space [1], which is an official URN namespace. The syntax definitions below use the Augmented Backus-Naur Form as specified in [RFC5234].

## The MCP Namespace

The syntax for an MRN is as follows [1]:

MRN = "urn" ":" "mrn" ":" OID ":" OSS

[ rq-components ]

[ "#" f-component ]

OID = (alphanum) 0\*20(alphanum / "-") (alphanum)

OSS = OSNID ":" OSNS

OSNID = (alphanum) 0\*32(alphanum / "-") (alphanum)

OSNS = pchar \*(pchar / "/")

The rules for alphanum and pchar are defined in [RFC3986]. The optional rq-components and f-component are specified in [RFC8141].

"mrn" specifies that the URN is within the MRN namespace. The *Organization ID (OID)* refers to an organization that is assigned a subspace of MRNs such as IMO, IALA, or the MCP. Syntactically, it is a string that must be unique across the "mrn" scheme. The *Organization Specific String (OSS)* is specified and managed by the governing organization in a consistent way conform to the definitions of the MRN namespace. In particular, each organization must structure the OSS into two parts: the *Organization Specific Namespace ID (OSNID),* and the *Organization Specific Namespace String (OSNS)*. The OSNID identifies a particular type of resource (uniquely within the governing organization), while the OSNS identifies the particular resource (uniquely for its type within the governing organization). Altogether, this ensures that the resulting URN is globally unique.

For a MRN governed by the MCC the OID reads "mcp", and the OSNID specifies one of the following types used within the MCP: any, device, organization, user, vessel, service, mir, mms, and msr. The latter three types are intended to be used for entities of the three MCP components: Maritime Identity Registry, Maritime Messaging Service, and Maritime Service Registry, respectively. Moreover, the definition of the OSNS takes into account the distributed structure of the MCP: identities can be provided and managed by several Identity Service Providers. In detail, the syntax of a *MRN governed by the MCC* (short: *MCP MRN* or *MCP name*) is as follows:

MCP-MRN = "urn" ":" "mrn" ":" "mcp" ":" MCP-TYPE ":" IPID ":" IPSS

MCP-TYPE = "entity" / "mir" / "mms" / "msr" / LEGACY

LEGACY = "device" / "org" / "user" / "vessel" / "service"

IPID = <CountryCode> / 3\*22IPID\_CHAR

IPID\_CHAR = unreserved / pct-encoded

IPSS = pchar \*(pchar / "/")

The rules for unreserved and pct-encoded are defined in [RFC3986].

Each element of the MCP MRN is defined as follows:

* "mcp" specifies that the governing organization is the MCC.
* *MCP-TYPE.* As mentioned above this specifies one of the types possibly used within the MCP. "mir", "msr" and "mms" are intended for internal MCP purposes. For other types, the "entity" type should be used. "device", "org", "user", "vessel" and "service" can be used to indicate identity types; however these are not formally defined and is considered to be lagacy. If an identity provider chose to use these, no specific information of the type should be assumed by other parties.
* The *Identity Provider ID (*IPID*)* refers to a national authority or other kind of organization that acts as an Identity Service Provider within the MCP. IPID country code as defined by ISO 3166-1 alpha-2 are reserved for national authorities that function as an Identity Service Provider. Otherwise, it will be a string of the same syntax as that for OIDs. The IPID must be unique across the urn:mrn:mcp namespace.
* The *Identity Provider Specific String (*IPSS*)* can be defined and managed by the respective Identity Service Provider in a way that is consistent and conforms to the definitions of the MRN namespace and requirements laid down by the MCC. In particular, the Identity Service Provider must ensure that the IPSS identifies a particular resource uniquely for its type within the domain of the Identity Service Provider. Altogether, this will ensure that the resulting URN is globally unique.

Examples:

1. urn:mrn:mcp:entity:dma:alice - valid MCP MRN, where dma specifies the ID Provider, and the subsequent IPSS string is defined to give the username.
2. urn:mrn:mcp:entity:mirX:aton:gb:sco:6789-1 - valid MCP MRN for the same AtoN, where mirX specifies the ID Provider, and the subsequent IPSS string is defined to first specify the type of the device, and then to follow the country-specific convention of the IALA scheme.
3. urn:mrn:iala:aton:gb:sco:6789-1 - valid MRN for a marine aid to navigation (AtoN), where gb stands for United Kingdom, sco for Scotland, and the number is the Scottish asset identifier. The example is from [4]. This is *not* a MCP MRN.

The following requirements pin down that and how the MCP namespace can be managed decentrally.

**ID1** The MCC can delegate the assignment of part of the MCP namespace to other organizations that act as Identity Service Providers. More concretely, this means that the organization, say X, must hold an IPID, say string "nameofx", and is then responsible for the namespace with the prefix "urn:mrn:mcp:<MCP-TYPE>:nameofx".

**ID1.1** The MCC must ensure that each IPID refers to at most one Identity Service Provider.

**ID1.2** Each Identity Service Provider must ensure to respect all syntax prescribed in the MRN specification. Moreover, each Identity Service Provider must ensure that each IPSS of their name space refers to at most one entity of their domain.

Note that ID1.1 and the second part of ID1.2 together ensure uniqueness: one MCP MRN is assigned to at most one entity. This is a general requirement for any URN.

Example:

Say there are two ID providers, MIR X and MIR Y. Assume the MCC assigns the IPID "mirX" to MIR X, and "mirY" to MIR Y respectively. The MCC must ensure that the strings "mirX" and "miry" are not assigned to any other MIR. MIR X is responsible for the namespace "urn:mrn:mcp:<MCP-TYPE>:mirX:\*", and MIR Y is responsible for the namespace "urn:mrn:mcp:<MCP-TYPE>:miry:\*" respectively. They might decide to employ the same syntax for the IP specific string, and make this part of a profile they both adhere to. Other ID providers are not bound to use the same syntax. However, if they do not comply to it they cannot be compliant to that profile.

## Further Requirements for a Strong Notion of Maritime Identity

The vision of the MCP is to enable a strong concept of digital maritime identity. Hence, we put down requirements that go beyond what is commonly required of URNs. Firstly, we require that every MCP entity must have a name within the MCP namespace. This gives a clear concept of MCP entity: those entities that are registered under an MCP MRN name. Secondly, we require that one MCP entity cannot have several MCP MRNs.

**ID2.1** Each Identity Service Provider shall ensure that each entity they register holds exactly one MCP MRN within their namespace. This does not exclude that an MCP entity can hold other MRNs, but these must be within namespaces governed by other organizations (e.g. IMO). Also, we will formulate exceptions concerning legacy MRNs within the MCP namespace.

Hence, the AtoN in the example above can be identified by its IALA MRN, or its MCP MRN respectively. However, Requirement ID2.1 rules out that the AtoN can be referred to by a second MCP MRN from the same identity provider.

**ID2.2** Each MCP MRN registered at an MCP Identity Service Provider is to be interpreted as a distinct entity.

The rule for ID2.1 ensures that all identities at MCP Identity Service Providers holds an MCP MRN. ID2.2 states that different MCP MRNs are to be interpreted to be different entities; this is both at a specific and across multiple MCP Identity Service Providers.

# Public Key Infrastructure

In addition to a unique ID in the form of an MCP MRN each MCP entity is provided with a cryptographic identity. This consists of a public/private key pair and a certificate for the public key bound to their ID. In the following, we describe the concept of the PKI that enables this, and a first set of requirements for it. We also identify issues that need to be addressed and refined in the future.

We proceed as follows. In Section 1 we explain the MCP core concepts of cryptographic identity. Section 2 details the decentral PKI. In Section 3 we specify the requirements on cryptographic keys and mechanisms. In Section 4 the format of MCP certificates is described. Moreover, in Section 5 we show how a service can use an intermediary level of service certificates. For example, this is necessary if a service comes with cryptographic requirements that do not allow the direct use of the MCP ID credentials. Finally, in Section 6 we identify further aspects to be considered.

## cryptographic Identity

The cryptographic identity of an MCP entity consists of at least one public/private key pair and a certificate bound to their MRN. The certificate must be issued by the Identity Service Provider responsible for the entity. The latter is clearly defined by the IPID string within the MRN of the entity.

Given an entity with MRN A (short: entity A), and its Identity Service Provider P, we use the following notation:

* + pkA is the public key of A, and prA is the private key of A respectively.
  + certP(A, pkA, V) is the certificate of A signed by its Identity Service Provider P. The certificate contains the MRN A, the public key of A, and the validity period V of the certificate. (The precise format is provided in Section 3.3.)

The key pair is for use with a digital signature scheme. Hence, each MCP entity A can be verified by another party B to be the originator of a message or other data. As usual this involves the following steps:

1. Entity A signs the message, say M, using its private key prA. The result is a cyphertext C.
2. Entity A makes available its certificate certP(A, pkA, V), and transmits the signed message (M concatenated with C).
3. Entity B obtains the certificate and receives the signed message.
4. Entity B validates the certificate. As a result, B trusts that pkA is the valid public key of the MCP entity with MRN A. (Necessary requirements on certificate validation will be specified.).
5. Entity B uses pkA to verify whether the ciphertext C is indeed the digital signature of M. If the verification is successful, then B has assurance that M indeed originates from A. (Note that without the fourth step B only has assurance that M originates from the holder of the private key counterpart of pkA.)

Note that B does not necessarily need to be an MCP entity.

At the time of writing the MCC does not prescribe a policy on how to use ID credentials. They could be used as long-term credentials to obtain short-term credentials for use for a service, or they could be directly used as working credentials.

## decentral PKI

One of the principles of the MCP is to make do without a global notion of trust: in the international context of the MCP we cannot expect that all parties trust each other and each other's security management uniformly. Rather the goal of the MCP is to provide the transparency that enables organizations to decide on whom to trust in which context, and to provide the technical framework to translate such decisions into executable policies. This motivates the following requirements:

**PKI1.1 (PKI Structure) There shall be no root CA at the top level of the MCC. Every Identity Service Provider that hosts a PKI instance is to provide their own root CA.**

**PKI1.2 (Validation of IPID) When a receiving party verifies a MCP certificate, say certP(A, pkA, V), it must verify that the certificate is indeed signed by the Identity Service Provider responsible for A. The Identity Service Provider responsible for A can be read by the receiving party from the IPID string within the MRN A.**

For example: For the vessel with MCP MRN urn:mrn:mcp:entity:duckville:scrooge-lines:dollar1 the identity service provider is found with the IPID duckville and the responsible MIR has the MRN urn:mrn:mcp:mir:duckville. The identity service provider for duckville must also have an entity MRN, i.e. urn:mrn:mcp:entity:duckville.

The following requirements ensure that information on root certificates and security levels are made publicly available.

**PKI1.3 Every Identity Service Provider is to publish their currently valid root certificate in a suitable fashion. For example, this can be made accessible via their web page, or they can commission a generally accepted authority or assurer to do so.**

**PKI1.4 Every Identity Service Provider must publish the Certificate Policy, and Certification Practice Statement detailing the actual operation of the MIR service. The Certificate Policy and Certification Practice Statement must follow best practice and include the Basic Requirement with implementation details where relevant.**

**PKI1.5 Every Identity Service Provider is to generate and publish a root certificate revocation list (CRL) containing any revoked issuing CAs. All active issuing CAs must include an endpoint to the root CRL.**

**PKI1.6 Every Identity Service Provider is to generate and publish CRLs containing any revoked MCP ID certificates for each of its issuing CA’s.**

**PKI1.7 Every Identity Service Provider is to support and provide an endpoint for an online certificate status protocol (OCSP) responder.**

From this the MCC will provide a secure way to automatically find and give basic trust in the authenticity of the MCP Identity Service Providers.

**PKI1.8 The MCC will publish one current and valid root certificate that is used to authenticate (sign) each Identity Service Provider certificate.**

**PKI1.9 The MCC will provide a list of Identity Service Providers, links to obtain their root certificates, security levels, and signatures of certificates signed with the given root certificate. Including a revocation list.**

**PKI1.10 Each Root Certificate is assigned to the entity MRN (e.g. urn:mrn:mcp:entity:duckville) and the intermediate certificate used to sign entity certificates is assigned to the respective MIR (i.e. urn:mrn:mcp:mir:duckville).**

**PKI1.11 The entity with a Root Certificate must also have a client certificate for normal certificate usage. Root Certificates must only be used for signing of MIR intermediate certificates.**

**PKI1.12 The Intermediate Certificate of a MIR must only be used to sign client certificates. The MIR must use another certificate for other usage.**

The MCC board will manage this root certificate, and detail guidelines and rules for its operation; this includes the Certificate Policy and Certification Practice Statement. These rules should follow best practice and will be published on the MCC website. This will also include location of valid certificates, signed certificates, and revocation lists. There will also be example code on how to interact with this. The management can be delegated by the board to a specific host member.

Note, that this does not break with the above claim that the MCC will not work as a root CA. This certificate is intended to only give a basic trust, meaning that the authenticated MCP instances are endorsed by the MCC and, to the best of MCCs knowledge, are operating within rules and guidelines as defined by the MCC. As stated earlier, full trust can only be established between each organisation and if deeper trust is needed, we must refer to other PKI systems or external certification organisations. Details of this is ongoing work and will be addressed at a future point in time.

### 5.2.1. security Requirements and Profiles

Security requirements to be defined will fall into the following categories:

1. Requirements on vetting. This can be specified similarly to classes such as EV (extended validation).
2. Requirements on certificate revocation.
3. Requirements on the validity period of certificates.
4. Requirements on security of keys and origin of signing - CA side (including requirements on Hardware Security Modules (HSMs)).
5. Requirements on security of keys and origin of signing - MCP entity side (including requirements on HSMs).

The requirements will be dependent on the currently emerging profiles:

MCP entities generate their ID key pair themselves and in own responsibility and provide this to the responsible CA for certification.

The CA (perhaps together with a manufacturer) provisions the initial ID key pair and certificate securely within HSMs (for/within endpoints) to be distributed to the MCP entities.

## 5.3. Cryptographic Requirements

The cryptographic mechanism approved for ID digital signatures is the Elliptic Curve Digital Signature Algorithm (ECDSA) [FIPS 186-3] with the appropriate hash algorithm from the SHA-2 family [FIPS 180-3]. The approved elliptic curve domain parameters are specified by reference to standardized curves. Currently the following combinations are approved:

|  |  |  |
| --- | --- | --- |
| **ECDSA Key Size (bits)** | **Hash Algorithm** | **Elliptic Curve Domain Parameters** |
| 384 | SHA-384 | P-384 [FIPS 186-3] (= secp384r1) |
| 256 | SHA-256 | P-256 [FIPS 186-3] (= secp256r1) |

**Future extensions:**

* Requirements on key pair generation and checks for key pair validity will be given by reference to standards. Also, we will check whether there are relevant recommendations in the last version [FIPS 186-5].
* Currently the only approved curve parameters are the NIST recommended curves. It will be checked whether this needs to be extended with regards to cryptographic recommendations of member states' security agencies (e.g., BSI and brainpool curves). Also, if a curve is found to be weak in the future it will be good to have an alternative curve per key size already approved.
* We will also consider matters of crypto agility.

## 5.4. Certificate Format

The format of the MCP ID certificates is as follows. The format is based on the X.509 standard [2]. The standard information present in an X.509 certificate includes:

* **Version** – which X.509 version applies to the certificate (which indicates what data the certificate must include).
* **Serial number** – A unique assigned serial number that distinguishes it from other certificates.
* **Algorithm information** – the algorithm used to sign the certificate.
* **Issuer distinguished name** – the name of the entity issuing the certificate (MCP).
* **Validity period of the certificate** – start/end date and time.
* **Subject distinguished name** – the name of the identity the certificate is issued to.
* **Subject public key information** – the public key associated with the identity.

The Subject distinguished name field should consists of the following items:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field** | **User** | **Vessel** | **Device** | **Service** | **AtoN** | **Organization** |
| CN (CommonName) | Full name | Vessel name | Device name | Service Domain Name | AtoN name | Organization Name |
| O (Organization) | Organisation MRN | | | | |  |
| E (Email) | User email |  |  |  |  | Organization email |
| C (Country) | Organization country code | | | | | |
| UID | Entity MRN | | | | | |

Table 1: Subject distinguished names fields.

*Example:* The following gives an example of the Subject distinguished name field for a vessel with Identity Service Provider idp1:

C=DK, O=urn:mrn:mcp:entity:dk, CN=Ship Name, UID=urn:mrn:mcp:entity:dk:shipname

In addition to the information stored in the standard X.509 attributes listed above, the X509v3 extension SubjectAlternativeName (SAN) extension is used to store extra information. There already exists some predefined fields for the SAN extension, but they do not match the need we have for maritime related fields. Therefore the “otherName” field is used, which allows for using an Object Identifier (OID) to define custom fields. The OIDs currently used are not registered at ITU, but are randomly generated using a tool provided by ITU (see <http://www.itu.int/en/ITU-T/asn1/Pages/UUID/uuids.aspx>). See the table below for suggested the fields defined, the OIDs of the fields and which kind of entities that use the fields.

|  |  |  |
| --- | --- | --- |
| **Field** | **OID** | **Used by** |
| Flagstate | 2.25.323100633285601570573910217875371967771 | Vessels, Services |
| Callsign | 2.25.208070283325144527098121348946972755227 | Vessels, Services |
| IMO number | 2.25.291283622413876360871493815653100799259 | Vessels, Services |
| MMSI number | 2.25.328433707816814908768060331477217690907 | Vessels, Services |
| AIS shiptype | 2.25.107857171638679641902842130101018412315 | Vessels, Services |
| Port of register | 2.25.285632790821948647314354670918887798603 | Vessels, Services |
| Ship MRN | 2.25.268095117363717005222833833642941669792 | Services |
| MRN | 2.25.271477598449775373676560215839310464283 | Vessels, Users, Devices, Services |
| Permissions | 2.25.174437629172304915481663724171734402331 | Vessels, Users, Devices, Services |
| Alternate MRN | 2.25.133833610339604538603087183843785923701 | Vessels, Users, Devices, Services |
| URL | 2.25.245076023612240385163414144226581328607 |  |

Table 1: Suggested defined fields and their usage by entity kinds.

Encoding of string values in certificates must follow the specifications defined in RFC 5280, and where possible it is highly recommended to use UTF-8.

To be able to check the revocation status of a given certificate all MCP ID certificates must include an endpoint to an up-to-date certificate revocation list that is signed by the issuing CA that has signed the certificate in question according to RFC 5280[2].

Additionally, all MCP ID certificates must also include an endpoint to an OCSP responder that is able to return the revocation status of the certificate in question according to RFC 6960[3].

## 5.5. recommendations for the validity period of the certificate

A diagram of a graph

Description automatically generatedFor different entities/levels in a certificate hierarchy, different validity periods are appropriate. At the highest level, root certificates should have a longer validity that overlaps with periodically newly added root certificates. At the intermediate level(s) the validity period should be shorter, but still long enough to allow distribution before they can be actively used.

Figure 1: Certificate renewal strategy.

For certificate validity periods in a root-intermediate-client hierarchy are:

* MCP Identity Service Provider Root Certificate: They should behave as Root-CA: 10 years validity and renewed every 3 years, start signing when 1 year old.
* MCP Identity Service Provider Intermediate Certificate: 3 years validity, renew every 1 year, start signing when 1 year old.
* End-system or client certificate (signed by intermediate): 6 months validity, renew 2 months before expiry.

When renewing a certificate, it is recommended to always renew keys as well as signing requests.

When creating a certificate, one could a​​lso choose a postponed validity starting date to cover the starting period for distribution before active use. In the graphic to the right, the above scheme is depicted.

It is suggested to standardise all periods and have specified periods for the root, the intermediate (MCP instance, issuer) and the end user, whether client or server.

Keys and certificates must be archived and/or deleted according to the guidelines specified in [6].

## 5.6. CERTIFICATE RENEWAL

All Identity Service Providers and their MIR implementations must allow automatic renewal of certificates according to [7].

All device and software suppliers that use MCP identities must allow the automated renewal of certificates without the need for manual maintenance activities.

All device and software suppliers must allow the purchaser to use their own certificates and define the MIR instance that is used to automatically renew the certificates.

## 5.7. service Certificates

Several maritime services come with requirements concerning cryptography and/or certificate formats that might make it impossible to employ MCP ID credentials directly. For example, if an Identity Service Provider issues certificates for ECDSA with 384 bits key size this will not meet the real-time requirements and low bandwidth conditions of AIS and VDES [4]. While the service must then provide its own CA the service CA can automatically issue its service certificates based on MCP ID credentials. We provide an example of how this can be done based on the concept of *certificate signing requests (CSRs)*, also known as *certification requests*. The most common format for CSRs is defined by the PKCS#10 standard [RFC 2986].

*Example:* In the following we show the steps carried out by an MCP entity to request a service certificate, and the steps performed by the service CA to issue the certificate respectively. The example follows the implementation of the Haptik CA from the project Haptik [5].

The MCP entity

1. generates a fresh key pair for use with the service,
2. builds a X.500 name for use in the service certificate,
3. builds a corresponding PKCS#10 CSR,
4. signs the CSR with their private MCP ID key, and
5. sends the CSR together with their MCP ID certificate to the service CA.

On receipt the service CA

1. checks whether the CSR is valid,
2. builds a X.509v3 certificate according to the CSR and additional information provided by the CA such as issuer, serial number, and validity period,
3. signs this with their CA private key, and
4. sends the new certificate to the requesting MCP party.

**Note:** This pattern is also applicable when the MCP ID keys are mainly used as enrolment keys to obtain shorter lived "working keys".

## 5.7. Obtaining the certificate of an MCP entity

An MCP Identity Service Provider must provide an interface that can be used by an actor to get either a specific certificate based on its serial number or cryptographic thumbprint, or any active certificates of an MCP entity with a given MCP MRN.

This interface must follow the *GetPublicKey* service interface specification described in section 8.6.3 of IEC 63173-2 (SECOM)[6] with the modification that when providing an MRN as the input parameter the multiplicity of the return value must be **0..\*** instead of **0..1**, effectively meaning that the interface can return zero or more certificates for an entity with a given MRN.

References

1. RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile; Internet Engineering Taskforce
2. RFC 6960: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP; Internet Engineering Taskforce
3. Public Key Authentication for AIS and the VHF Data Exchange System (VDES)  
   G.Wimpenny, J. Šafář, A. Grant, M. Bransby & N. Ward  
   31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018)  
   October 2018  
   DOI:10.33012/2018.15948
4. [https://haptik.io](https://haptik.io/)
5. IEC 63173-2:2022: Maritime navigation and radiocommunication equipment and systems – Data interfaces – Part 2: Secure communication between ship and shore (SECOM)
6. NIST SP 800-57 <https://doi.org/10.6028/NIST.SP.800-57pt1r5>
7. RFC 8555 Autmatic Certificate Management Environment https://doi.org/10.17487/RFC8555

**ANNEX A**

**The MCP Consortium**

The MCP consortium undertakes various tasks in relation to the MCP. In addition to the endorsement mechanism mentioned in this document, the consortium;

* Develops and maintains an open source implementation of all MCP components
* Operates a non operational public demonstrator of the MCP, which all maritime stakeholders can use freely for assessment and testing purposes
* Control access to the distributed ledger, which will hold globally discoverable information from all endorsed MCP service registry service providers (facilitating global service discoverability). This is only relevant for the Service Registry part of the MCP, which is described in IALA guideline XXXX.

## Annex decentral PKI

One of the principles of the MCP is to make do without a global notion of trust: in the international context of the MCP we cannot expect that all parties trust each other and each other's security management uniformly. Rather the goal of the MCP is to provide the transparency that enables organizations to decide on whom to trust in which context, and to provide the technical framework to translate such decisions into executable policies. For the PKI we put forward the following three principles:

1. A security breach within the realm of one Identity Service Provider's PKI instance shall not enable an attacker to impersonate an entity within the realm (i.e. namespace) of another Identity Service Provider;
2. A security breach within an organization or set of organizations predefined by the MCC to carry out some tasks shall not enable an attacker to impersonate any MCP entity unless the Identity Service Provider of the entity coincides with (one of) the organization(s). In short this means Identity Service Providers' PKI instances can always remain secure independently from any central or distributed management by the MCC;
3. It is possible for everyone to obtain assurance as to the security level of any Identity Service Provider's PKI instance.A diagram of a structure

   Description automatically generated

Figure 1: Hierarchical X.509 PKI Structure

The first two principles immediately imply that a classical hierarchical PKI with a root CA hosted by the MCC won't do. We illustrate this by giving examples of impersonation attacks. Assume a hierarchical PKI structure with MCC root CA as shown in Fig. 2. To verify a MCP certificate certP(A, pkA, V) a receiving party has to verify the signature of the Identity Service Provider P with the public key of P provided in an intermediary certificate certMCC(P, pkP, V) issued by the MCC root CA. Further, to verify the intermediary certificate the receiving party has to verify the signature of the MCC with the public key provided in the MCC root certificate certMCC(MCC, pkMCC, V). The latter provides the trust anchor accepted by the receiving party.

*Examples:*

1. *Single point of attack:* Assume the MCC root key prMCC is compromised. This will allow an attacker to impersonate any MCP entity A. Say P is the Identity Service Provider responsible for A. First, the attacker generates a key pair pkI(P), prI(P) and generates a fake certificate for P with his own key pkI(P): certMCC(P, pkI(P), V). This is possible since the attacker knows prMCC. Second, the attacker generates a key pair pkI(A), prI(A) and generates a fake certificate for A and his own key pkI(A): certI(P)(P, pkI(A), V). This is possible since he knows prI(P). Altogether, the attacker can now present a valid certificate chain that establishes pkI(A) to be the public key of A while he knows the private counterpart prI(A). Hence, he can impersonate A. Altogether this violates principle 2 above.
2. *Weakest Link I:* Say the attacker wishes to impersonate entity A of Identity Service Provider P. Note that in classic X.509 certificate validation it is only verified that there is a certificate chain up to a trusted root certificate. Say the attacker can easily obtain fake certificates signed by another Identity Service Provider P', perhaps, because the attacker is a state actor and P' is under his governance. Then, analogously to above, he only needs to generate his own key pair pbI(A), prI(A) and generate a fake certificate for A and pbI(A) signed by P': certP'(A, pbI(A), V). Since there is no check whether P' is indeed the Identity Service Provider of A this gives the attacker a valid certificate chain and corresponding private key, with which he can impersonate A. This violates principle 1 above.
3. *Weakest Link II:* Assume P is an Identity Service Provider of low security level, e.g., with a vetting procedure that can easily be undermined. Assume an attacker aims to join the MCP under a false identity so that he is able to inject fake messages without the risk of being traced. The attacker will simply choose P as the Identity Service Provider from whom to obtain his false identity. Without principle 3 in place a receiving party has no way to consider the low security level of P when processing the information within the message.

In the spirit of decentralisation, the PKI shall remain open for PKI systems other than X.509, and be agile for updates of certificate formats. Care will be taken to accommodate the necessary flexibility when defining usage of certificates.

## Annex recommendations for the validity period of the certificate

For different entities/levels in a certificate hierarchy, different validity periods are appropriate. At the highest level, root certificates should have a longer validity that overlaps with periodically newly added root certificates. At the intermediate level(s) the validity period should be shorter, but still long enough to allow distribution before they can be actively used.

Considering the validity period of client-certificates for end-systems/users, looking at current (2023) certificate expiry policies in ‘industry best practices’, we see browsers (Chrome, Safari, Edge) accepting max. 13 months (397 days) of certificate validity without warnings and Google wanting to move to 90 days, which is also the Let’s Encrypt standard.

For end-system/user certificates, the general trend is the shorter the better, more secure. But this cannot be achieved in every environment, so balance should be sought. For example, for entity types such as devices and vessels, the certificate validity period should be proportional to the maintenance interval.

Motivation for a short certificate validity period:

* Practice – do it often, don’t forget how to
* Endorses automated renewal instead of manual renewal
* Helps in damage control when a private key is leaked