Grid Certificate Profile

Status of This Memo

This memo provides information to the Grid community. It does not define any standards or technical recommendations. Distribution is unlimited.

Abstract

Interoperability for X.509 identity certificates between issuers of those certificates and the software that interprets the certificates has become increasingly important with the growth of the global grid community. As the number of participants in the grid that use certificates grows, the relationship between issuers and relying parties becomes weaker. This necessitates coordination, specification and in come cases restriction of the use of certain name forms and certificate extensions in order to ensure continued interoperability. This document provides guidance for the content of issuer and end-entity X.509 certificates for use with grid software.

Contents

1. Scope of this document

This document provides guidance for the use of attributes and extensions in X.509 certificates such that they are usable by the majority of the grid infrastructures today. This guidance must be interpreted in the context of RFC 3280 [RFC3280], *i.e.*, all certificates must be compliant to RFC 3280 in addition to any limitations imposed by the guidelines in this document, unless explicitly stated otherwise in this document.

Specific attention has been given to the representation of the subject and issuer distinguished names as strings, since in much of the grid software it is this string rendering, and not the actual sequence of relative distinguished names, which is used for identification and subsequent authorization purposes. This imposes specific additional constraints on such names, and on the set of attributes which can be used in these names, to ensure wide interoperability of the certificates.

If a particular extension or attribute is not discussed in this document, this should not be construed as to mean the extension or attribute is either useful or harmless; it means that at the time of writing it was not in widespread use, and was therefore not needed for interoperability. It may or may not be harmless and may of may not cause interoperability problems. It is recommended that specific interoperability testing is performed prior to including any such extensions or attributes.

The key words "MUST", "MUST NOT", “SHOULD”, “SHOULD NOT”, "REQUIRED", "SHALL", "SHALL NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2. Self-signed and subordinate Certification Authority certificates

2.1 General provisions

All Certification Authority (CA) certificates MUST be in X.509 version 3 format, *i.e.*, the version number MUST be set to the value “2”, as the use of specific extensions such as *basicConstraints* and *keyUsage* is required.

2.2 Serial Number

The serial number of each CA certificate SHOULD be unique[1].

If the end-entity certificates include an *authorityKeyIdentifier* extension with the issuer’s serial number, the serial number SHOULD remain the same on re-issuing of the CA certificate. Note that including the attribute *serial number* in *authorityKeyIdentifier* extension in end-entity certificates is discouraged.

For the message digest that protects the certificate integrity, known-weak signatures or hash functions, such as MD5, MUST NOT be used in new certificates. Note that modern hashes, such as SHA-256, are not supported by the majority of OpenSSL versions in use, so SHA1 is currently the only RECOMMENDED value.

---

[1] If a root or intermediate CA certificate is re-issued with the same serial number – for example in case only the lifetime is extended but the key pair remains the same – web browsers using the Mozilla NSS-base will issue a user warning and the import will fail, but if installation of the new certificate is attempted in Microsoft Internet Explorer it will overwrite the old one. Thus, for NSS-based browsers the old certificate has to be removed from the certificate store first.

If the serial number is changed, the import of the new root certificate in Microsoft Internet Explorer will result in both certificates being retained in the certificate store, and the original one is not overwritten.

2.3 Issuer and Subject names

Only a limited number of attribute types are well supported by the all of the current software implementations when used as part of the Issuer or Subject Distinguished Name (DN). Therefore, only the following attribute types SHOULD be used, as they can be considered "safe": *domainComponent* (DC), *country* (C), *state* (ST), *locality* (L), *organization* (O), *organizationalUnit* (OU) and *commonName* (CN). Use of other attributes in distinguished names MAY result in incompatible representations, and thus SHOULD NOT be used.

To ensure uniqueness and reproducibility of the string renderings of DNs, the ASN.1 SEQUENCE MUST only contain *SETs* of length 1. Other *SET* lengths MUST NOT be used.

Contrary to what may be deduced from the guidance given from X.521, multiple instances of the *organization* attribute MAY be used in a single DN. It has been confirmed by experience that all known software used in grid deployments today correctly handles their representation, and will collate the attributes in the proper order. Also, multiple instances of the *commonName* attribute MAY be used.

Note, however, that the visual rendering of a multiple *organization* (O) or multiple *commonName* (CN) attributes in many browsers may not be complete, and usually only the first or the last of these is displayed to the user. This only affects the visual representation, since all known grid middleware[2] uses the entire DN for subject identification. If no O or OU attributes appear in the DN, Mozilla-NSS based browsers will not use other components to show affiliation.

2.3.1 serialNumber

The attribute type *serialNumber* {id-at 5, i.e. 2.5.4.5} MUST NOT be used in any Name[3].

2.3.2 emailAddress

The attribute type *emailAddress* SHOULD NOT be used in DNs. It has been obsoleted in RFC 3280, in favour of having an *rfc822EmailAddress* in the *subjectAlternativeName* X.509v3 extension, and many recent mail clients can deal with *subjectAlternativeName*.[4]

In all cases, the CA certificate itself is not usually used to send email, so mail client support is not an issue to be considered for CA certificates.

---

[2] This has been tested also for the latest version of FreeRadius.

[3] The *serialNumber* attribute was originally intended to describe the serial number of a device [X.520]. There have been discussion on the PKIX mailing lists on whether it was also appropriate for persons, and then only to distinguish different persons with the same commonName from each other. In particular, it is not intended to contain the certificate serial number.

There is a another reason not to use the *serialNumber* attribute: versions of OpenSSL up to and including version 0.9.6 use a non-standard string representation "SN" for this attribute. This representation collides with the recognised abbreviated representation of the *surname* attribute. This representation has changed in OpenSSL 0.9.7+ to read “serialNumber”, so depending on the OpenSSL version used the string representations of DNs with the *serialNumber* RDN attribute type will differ, leading to problems in authorization.

[4] String representation issues with the *emailAddress* attribute in DNs are caused by OpenSSL, where versions up to and including 0.9.6 used the non-standard string representation "Email" for this attribute type, and later versions use “emailAddress”, thus resulting in different string representations for the same DN and leading to problems in subsequent authorisation decisions.

2.3.3 userID or uid

The attribute type *userID* or *uid* {0.9.2342.19200300.100.1.1} MUST NOT be used in Names. Additionally, it is not relevant for CA certificates of any kind.[5]

2.3.4 DomainComponent, country, organization, organizationalUnit, etc.

The distinguished name is usually made up of a combination of the attribute types "DC", "C"[6], "ST", "L", "O", "OU" and "CN".

To ensure uniqueness and proper delegation, the use of *domainComponent* (DC) naming corresponding to a registered DNS name owned by the authority at the beginning of the issuer and subject name RDN sequence is strongly encouraged. In that case, the ASN.1 *SEQUENCE* MUST start with the *domainComponent* representing the top-level domain, for example "DC=org" or "DC=eu".

The use of at least one descriptive *organization* O attribute in the DN is encouraged.

2.3.5 commonName

The commonName SHOULD be used in the subject distinguished name of a CA root certificate, as it allows easy visual recognition of the CA name. As the CN of the subject DN is often the most prominent displayed name of the CA the CN (in addition to the O entry, whose addition is encouraged) SHOULD be a descriptive explicit string distinguishing the authority's name.[7]

2.4 Extensions in CA certificates

For operation as a CA certificate, only *basicConstraints* and *keyUsage* extensions need to be present in the (root or subordinate) certificate. To be functional as an issuer certificate, there is no a priori requirement by (grid) software for any other extensions in the certificate.

2.4.1 basicConstraints

The *basicConstraints* extension MUST be included in CA certificates, and it MUST be set to "CA: TRUE". This extension MUST be marked as critical.

2.4.2 keyUsage

The *keyUsage* extension MUST be included in CA certificates, and it SHOULD[8] be marked as critical.

---

[5] The string representation of the *userID* or *uid* attribute is not uniquely defined. OpenSSL versions up to and including 0.9.6 have no string representation for this, and this omission has resulted in some versions of the Globus Toolkit that use this OpenSSL version to forcibly re-code the string representation of this attribute to read "USERID". Recent OpenSSL versions stringify it to the RFC 2253 standard representation "uid", resulting in a non-unique representation. Note that both "uid" and "userid" are valid standard string representation of the attribute with OID 0.9.2342.19200300.100.1.1, with "userid" defined in RFC1274 and "uid" in 2253.

[6] If a *Country* (C) component is included in the issuer DN, it SHOULD reflect the country in which the issuer is based.

[7] Having a *commonName* of just "CN=CA" will result in the display name of the CA in many browsers to show just the string 'CA' as the name, which may result in confusion.

[8] The CA must ensure that the use of public keys is minimal and relevant to the goals of its PKI, particularly for its own public key (in the CA certificate). It does this by defining acceptable and unacceptable uses in the policy, but also by setting the appropriate extensions in the certificates. Compliant software will then find it harder to use the CA's public keys for inappropriate purposes. If it is found that the CA's public keys are used for purposes contrary to the defined goals of its PKI, it can adversely affect the CA's name, reputation, or operations, and, ultimately, the most precious thing it has - trust.

For a CA certificate, *keyCertSign* MUST be set, and *crlSign* MUST be set if the CA certificate is used to directly sign issued CRLs[9].

It is RECOMMENDED to set no more than these two attributes. For proper operation it is not required to have more than *keyCertSign* and *cRLSign* in the CA certificate and adding additional attributes may convey an incorrect impression to relying parties.

2.4.3 extendedKeyUsage

The *extendedKeyUsage* extension SHOULD NOT be included in CA certificates[10]. It MUST NOT be marked critical.

2.4.4 nsCertType, nsComment, nsPolicyURL, nsRevocationURL

The *ns** attributes are deprecated and SHOULD NOT be included in any new CA certificates. If they are included, though, these extensions MUST NOT be marked critical[11].

2.4.5 cRLDistributionPoints

The *cRLDistributionPoints* extension need not be in a self-signed root CA certificate, but MUST be included in end-entity certificates and SHOULD be included in any intermediate CA certificates[12].

For subordinate CAs, where a CDP is present, it MUST contain at least one http URL[13].

2.4.6 Authority and Subject Key Identifier

A *subjectKeyIdentifier* extension MAY be included in CA certificates to aid in validation path construction. An *authorityKeyIdentifier* MAY be included in all CA certificates. For a self-signed root certificate, the *authorityKeyIdentifier* and *subjectKeyIdentifier* MUST be the same.

If either of these extensions is included, it SHOULD include only the *keyIdentifier* attribute and no other attributes.

---

[9] There may be CAs that either do not issue CRLs at all, since their end-entity certificates have a short life time, or that use indirect CRLs. The use of indirect CRLs has not been extensively tested, but it is not supported at all by openssl, and it is probably not tested well or supported well in other software, unfortunately. It can't really be tested because nobody seems to be able to create either a client or a "signer". For instance there is no direct path to create such an end-entity certificate in the Sun One/Iplanet CMS product, although direct generation of the ASN.1 is always a possibility. But grid middleware today cannot use it.

[10] *extendedKeyUsage* should not be included not only because the values of this attribute are not normally relevant for CA certificates, but also it will make the certificate unsuitable for use with Microsoft Internet Explorer version up to and including version 6, and unsuitable for use with any version of Microsoft Outlook, as these products will make a logical 'and' between *keyUsage* and *extendedKeyUsage* extensions for potentially unrelated usages.

[11] If adding explicit text to the certificate, such as was possible using the *nsComment* extension, is desired, the new attribute to put such text is the *certificatePolicies.userNotice.explicitText* (encoded as an IA5String). Note that RFC3280 RECOMMENDS that only an OID is used in the *certificatePolicies* extension. Also, compliant RFC3280 implementations SHOULD actually display each and every user notice to the user.

[12] Client software can use the *cRLDistributionPoints* extension to retrieve CRLs on-demand, although no known grid software implementations today actually support that.

Note that by putting a CRL distribution URL in any CA certificate the authority implies that the URL will not change during the lifetime of the root or subordinate CA certificate, so, if included here, one SHOULD make sure the URL will be stable over the life time of the certificate.

[13] The URL should be a plain HTTP URL, and thus not an *https* URL. There are recursive bootstrap issues in validating the download of the https URL, and the CRL returned is signed and integrity protected anyway. The *cRLDistributionPoints* extension MAY contain other URIs.

3. End-entity certificates

3.1 General provisions

All end-entity certificates MUST be in X.509 version 3 format, i.e. the version number MUST be set to the value "2", as the use of specific extensions, such as *basicConstraints* and *keyUsage*, is required.

The serial number of each issued certificate MUST be unique amongst all certificates issued by the same issuer.

For the message digest that protects the certificate integrity, known-weak signatures or hash functions (such as MD5) MUST NOT be used in new certificates. Note that modern hashes, such as SHA-256, are not supported by the majority of OpenSSL versions in use, so SHA1 is currently the only RECOMMENDED value.

3.2 Subject distinguished names

The same general considerations mentioned for CA certificate subject names also apply to subject names in end-entity certificates.

Other RDN attribute types than "DC", "C", "ST", "L", "O", "OU", and "CN" SHOULD NOT be used.

To ensure uniqueness and proper delegated ownership of the certificate subject name space, the use of *domainComponent* RDN components corresponding to a duly registered DNS name [RFC1591] of the authority at the start of the distinguished name is strongly encouraged. Thus, the ASN.1 SEQUENCE MUST begin with the *domainComponent* attribute corresponding to the top-level domain (e.g. "org", or "eu"), and then be followed by the subordinate domain name components.

3.2.1 String encoding of the RDN components

Relative DN components in distinguished names SHOULD be encoded as PrintableString, contrary to any requirements stated in RFC 3280. A RDN MUST NOT contain characters that cannot be expressed in 7-bit ASCII, as these characters have inconsistent representations[14].

3.2.2 PrintableString encoding recommendations

RFC2252 defines PrintableString as consisting of 'a'-'z', 'A'-'Z', '0'-'9', and the characters '"', '(', ')', '+', ',', '-', '.', '/', ':', '?', ' ', that is, upper and lower case alphanumeric, double quote, left and right parentheses, plus, comma, minus/hyphen, dot (period), forward slash, colon, question mark, and space. This set is almost consistent with the PrintableString definition of RFC1778, differing only in allowing ''' (single quote), instead of '"' (double quote).

Of the allowable PrintableString characters, the comma SHOULD NOT be used[15]. The double quote MUST NOT be used and single quote SHOULD NOT be used[16] [17].

---

[14] Non-7-bit ASCII characters have different string representations in different pieces of software, and cannot easily be passed around between locales, or be read from log files. Use of such characters will result in undefined or inconsistent behaviour, e.g. in subsequent authorization.

[15] The comma should not be used since in the string representation of X.500 naming and RFC2253, the RDNs components are comma separated.

[16] The quote characters must not be used because OpenSSL follows RFC1778's definition of PrintableString

[17] OpenSSL uses forward slash ("/") in the one-line string representation to separate RDNs, making the use of the forward potentially confusing. But since there is always an equal sign (=) after the name of a RDN component in this representation, a proper parser should be able to parse this correctly and the equal sign is not part of the allowed character set.

The CA MUST ensure that case or consecutive spaces are not used to distinguish between users (e.g. users with the same name)[18].

3.2.3 commonName

A *commonName* attribute MUST be used in the subject DN of an end-entity certificate.

If the commonName is not encoded as printableString, it SHOULD be encoded as UTF8String.

To prevent name collisions between different entities, mainly in issuing personal certificates, a number or other allowed distinguishing characters can be added to the CN to ensure uniqueness[19]. It is usually allowed for an entity to have more than one subject DN assigned[20].

For certificates issued to networked entities, typically the (primary) FQDN of the server is included in the *commonName*. For regular network entity certificates, there must not be any additional characters in the *commonName*[21].

Some grid middleware, in particular any version of the Globus Toolkit, contains a design flaw that allows implicit wildcard matching of the domainname in the commonName attribute, where the first component of the domainname containing a dash ("-") is stripped of all characters from the dash onwards, and then matched to the FQDN in the commonName[22].

Note that for name-based virtual hosting, additional FQDNs can be asserted in the *subjectAlternativeName* extension in multiple *dNSName* attributes[23].

It should be noted that past versions of the FreeRadius [FR] uses only the *commonName* for its authorization decision. No grid middleware is known to act in this manner. Many browsers use only the *commonName* to label certificates in their certificate stores.

3.2.4 serialNumber

The AttributeType "serialNumber" (i.e. {2.5.4.5}) MUST NOT be used in any Name[24].

---

[18] While printableString encodings are supposed to be case insensitive [RFC3280], in practice most grid software uses case sensitive comparisons. A related problem is found with consecutive spaces which are supposed to be collapsed to a single space.

[19] Adding qualifiers to the CN is preferred over adding other attributes to the subject DN, such as the uid's or serialNumber attributes that MUST NOT be used.

[20] Having more than one DN (and thus also more than one certificate) per person is needed for some grid middleware for a person to be a member of more than one community. Although this certainly is an authorization issue, it is advisable for CAs to allow a single person to hold more than one certificate – and limiting that to such special cases by policy.

[21] Some components of some grid middleware also recognize Kerberos-style "service" names in the CN as well that look like "*servicename*/*fqdn*". In the majority of the cases, a "normal" server certificate without the "*servicename*/"-qualifier can be used as well – although the documentation of the middleware will not always state that clearly. It is recommended to phase out the "*servicename*/"-qualifiers where possible.

[22] For example: a certificate issued to "CN=grid.example.org" can be used for successfully proving the identity of "grid-ce.example.org" as well as "grid-se.example.org" and "grid.example.org" itself.

[23] Many modern browsers, such as Microsoft Internet Explorer version 6 and higher, or Mozilla Firefox versions 1.5 and higher, will recognize these additional *dNSNames* in the *subjectAlternativeName* and recognise it as valid alternate names for the virtual web site.

[24] See footnote to section 2.3.1 for the argumentation.

Specifically, the serialNumber attribute MUST NOT be used to re-encode the certificate serial number in the subject name[25].

3.2.5 emailAddress

The attribute *pkcs9email* ("emailAddress") SHOULD NOT be used in subject names[26].

If used, by RFC3280 email addresses MUST be encoded in RFC822 "addr-spec" format (section 6.1) and they MUST be encoded as IA5String.

3.2.6 userID or uid

The attribute type "userID" or "uid" (i.e. {0.9.2342.19200300.100.1.1}) MUST NOT be used in Names[27].

3.2.7 domainComponent (DC), country (C), State (ST), Locality (L), Organization (O), and *OrganizationalUnit (OU)*

To ensure subject name uniqueness and proper namespace delegation, the use of *domainComponent* (DC) naming corresponding to a registered DNS name owned by the authority at the beginning of the issuer and subject name RDN sequence is strongly encouraged. In that case, the ASN.1 *SEQUENCE* MUST start with the *domainComponent* representing the top-level domain, for example "DC=org" or "DC=eu".

It is RECOMMENDED to encode the *domainComponent* as an IA5String[28]. Since all known software correctly parses all incoming encodings, all of PrintableString, IA5String and UTF8String MAY be used to encode *domainComponent*, where IA5String is preferred.

If the *Country* attribute is used, the value of this attribute MUST contain the two-letter ISO3166 encoding of the country's name[29, 30]. The *country*, if used, MUST be used at most once. Any of the *State (ST), Locality (L), Organization (O),* and *OrganizationalUnit (OU)* attributes MAY be used and have their usual meaning.

The use of at least one descriptive *organization* O attribute in the DN is encouraged.

---

[25] Not only is such use of *serialNumber* redundant, but it also makes renewals impossible.

[26] The *emailAddress* attribute in the subject DN has been declared obsolete in recent RFCs [RFC3280], in favour of having an *rfc822EmailAddress* in the *subjectAlternativeName* extension. Many recent mail clients are able to deal with the *subjectAlternativeName* (Lotus Notes and Web-Mailer Communigate are known exceptions). Parsing issues with this attribute are caused by OpenSSL, which in versions up to and including 0.9.6 used the non-standard string representation "Email" for this attribute type.

[27] See footnote to section 2.3.3 for the argumentation.

[28] The latest OpenSSL and the RedHat Certificate System versions encode the *domainComponent* attribute as an IA5String. OpenSSL versions 0.9.7c or older version encodes it as PrintableString.

Since PrintableString is really a subset of IA5String, one could modify incoming requests with a PrintableString encoding such that IA5String encodings are used in the issued certificates.

[29] Note the UK is an (in)famous exception, mainly for historical reasons – GB is Great Britain, and UK is "the United Kingdom of Great Britain and Northern Ireland". Ukraine MUST be encoded as UA.

[30] The *country* (C) asserted in the subject DN of an end-entity certificate SHOULD correspond the home country of the end-entity, and thus does not necessarily reflect and is not necessarily the same as the country in which the CA is operating, or the country code in the issuer DN. Therefore, in such cases the *Country* attribute should not be part of a unique subject DN naming prefix.

3.3 Extensions in end-entity certificates

For use of an end-entity certificate certificate with grid software, at least either of the *extendedKeyUsage* or *nsCertType*[31] extensions MUST be present, where the use of the *extendedKeyUsage* extension is preferred. Including *basicConstraints* is RECOMMENDED.

For end-entity certificates issued to networked entities (servers or services), the use of the *subjectAltName* extensions with a *dNSName* attribute is RECOMMENDED. For end-entity certificates that include an rfc822 email address, the *subjectAltName* extension SHOULD be used, and the email address included in the *rfc822Name* attribute.

It is RECOMMENDED that an end-entity certificate includes also the extensions *keyUsage*, *certificatePolicies*, and *cRLDistributionPoints*.

There is no a priori requirement by grid software for any other extension in end entity certificates.

3.3.1 basicConstraints

The *basicConstraints* extension is RECOMMENDED to be included in end-entity certificates[32]. The *pathLenConstraint* attribute MUST NOT be present[33].

If the CA software is capable of generating the *basicConstraints* extension with a *cA* attribute even if its value is “CA:FALSE”, this extension MUST be included in end-entity certificates, and its value MUST be set to “CA:FALSE”.

When present, this extension MUST be marked critical.

3.3.2 keyUsage

The *keyUsage* extension MUST be included in end-entity certificates, and it MUST be marked critical.

For an end-entity certificate, it depends on certificate usage which values need to be set.

> The *digitalSignature* and *keyEncipherment* values MUST be set for authentication in SSL sessions, and thus for typical grid usage, as otherwise grid authentication will not work. These two are the only values that are actually required.
>
> The *keyAgreement*, *encipherOnly*, and *decipherOnly* values primarily apply to DH keys, and need not normally be asserted in an end-entity certificate.
>
> The *nonRepudation* value SHOULD NOT be set for server certificates (including “host” and “service” certificates), as it implies that any use of the key would constitute incontrovertible evidence that the signing was done in a conscious way, which is unlikely for a server certificate. Its assertion in personal end-entity certificates SHOULD be limited to special purposes.
>
> The *dataEncipherment* value MAY be set, but is similarly intended for special purposes.

---

[31] The use of nsCertType is deprecated, see section 3.3.5.

[32] According to the ASN.1 encoding rules, a value “CA:FALSE” for *basicConstraints* is the default and thus should not need to be encoded as an extension, but recent discussion (on RFC3280bis) has made clear that it would be strongly advisable to include it.

It is not known if there is client software that will incorrectly allow signing of subordinate certificates if this extension is absent.

[33] Note that RFC3280 forbids the use of pathLenConstraints in end-entity certificates. If it is included anyway, it MUST allow for an unlimited path length to allow the user to issue proxy certificates [RFC3820].

The *keyCertSign* and *cRLSign* MUST NOT be set in an end-entity certificate, unless the certificate is explicitly intended for use in indirect CRL signing[34].

3.3.3 extendedKeyUsage

The *extendedKeyUsage* (EKU) extension SHOULD be included in end-entity certificates, but MUST NOT be marked critical.

For personal end-entity certificates or automated entities, *clientAuth* should be asserted in EKU. But in the grid context, servers at times do act like clients, and thus for host or service certificates it does make sense to include both *serverAuth* as well as *clientAuth*[35].

If this extension is included together with the *nsCertType* extension, the certificate purpose expressed in both extensions MUST be equivalent[36].

3.3.4 Application interplay between extendedKeyUsage and nsCertType

The *extendedKeyUsage* and *nsCertType* extensions are interrelated and do partially cover the same purposes. In any software based on the OpenSSL code, the *nsCertType* will be used to determine the SSL Server or Client purpose of the certificate in the absence of an *extendedKeyUsage* extension. Either of these MUST be present to ensure correct operation of grid and other software[37]. If both are present, the purposes expressed MUST be consistent.

3.3.5 nsCertType

This attribute is deprecated and it is RECOMMENDED not to use this extension in new certificates, and the appropriate equivalent attributes be included in the *extendedKeyUsage* extension.

If this extension is included together with *extendedKeyUsage*, the purposes expressed in both extensions MUST be consistent, for those attributes in *extendedKeyUsage* that express similar purposes[38].

If the *nsCertType* extension is included it MUST NOT be marked critical.

3.3.6 nsPolicyURL, nsRevocationURL

These attributes are deprecated and are not required in end-entity certificates. If any of these extensions is included, it MUST NOT be marked critical.

---

[34] See also section 2.4.2.

[35] This dual-use of host and service certificates action in both a server and a client role is required for, for example, the Network Job Service (NJS) and the Gateway in the Unicore grid middleware, where one NJS may forward a request to another NJS, and in this interaction the NJS acts as a client.

[36] Refer to Chapter 5 for all values that could be included in certificates.

[37] Either *nsCertType* or *extendedKeyUsage* must be present. For example, the OpenLDAP client needs at least one of "*nsCertType: server*" or "*extendedKeyUsage: serverAuth*" to be present in the LDAP server's server certificate to properly establish a SSL/TLS connection. If neither is present, the SSL server authentication will fail in the OpenLDAP client. Note that many grid operations rely on OpenLDAP in a secure mode.

Web browser clients and automated clients built based on Apache Axis stubs seem less picky about these extensions, and will survive if neither is defined in the server certificate. To what extent this holds for other software is unclear.

[38] So, for example for certificates issued to a Unicore NJS service, *nsCertType* can be set to "*server*, *client*", but it is preferred to set EKU to "*serverAuth*, *clientAuth*" and not to include any *nsCertType*.

3.3.7 nsComment

This attribute is deprecated and is not required in end-entity certificates[39]. If it is included, this extension MUST NOT be marked critical.

3.3.8 cRLDistributionPoints

The *cRLDistributionPoints* extensions MUST be present in end-entity certificates, and MUST contain at least one http URL (i.e., not an *https* URL) although it may contain other URIs[40] [41].

Some software[42] is known not to be able to handle any attributes other than a single URI in this extension.

It is RECOMMENDED that the reply returned at the http URL is cacheable[43].

3.3.9 authorityKeyIdentifier

The *authorityKeyIdentifier* (AKI) is not usually interpreted by the software, and is considered harmless to current known grid software. The AKI extension MUST NOT be marked critical.

If the AKI in an end-entity certificate contains information that changes when the issuer certificate is modified, it may block a 'smooth' replacement of isser certificates (e.g. when updating a CA certificate to modify the expiry date).

Possible attributes in AKI include the *directoryName* of the authority that issued the issuer certificate, which is safe to include as it should not change, as well as the serial number (which may or may not change), or the *keyIdentifier* of the end-entity issuing CA. If the *keyIdentifier* has been generated using one of the two recommended methods from RFC3280 (i.e. is purely derived from the public key value), it will not impair smooth replacement.

3.3.10 subjectKeyIdentifier

The *subjectKeyIdentifier* extension MUST NOT be marked critical.

3.3.11 certificatePolicies

The *certificatePolicies* extension MUST be present and MUST contain at least one policy OID. It MAY contain more than one OID, *e.g.*, to refer to an Authentication Profile, or one or more one-statement certificate policies (1SCPs).

The *certificatePolicies* extension SHOULD NOT be marked critical.

---

[39] If adding explicit text to the certificate, such as was possible using the *nsComment* extension, is desired, the new attribute to put such text is the *certificatePolicies.userNotice.explicitText* (encoded as an IA5String). Note that RFC3280 RECOMMENDS that only an OID is used in the *certificatePolicies* extension. Also, compliant RFC3280 implementations SHOULD actually display each and every user notice to the user.

[40] See also footnotes to section 2.4.5.

[41] Note that OpenSSL is not able to display the values of the *reasons* and the *CRLissuer* associated with a *DirectoryName* or *URI*.

[42] As of August 11, 2006, this is known to apply only to VOMS and VOMS-Admin. This has been reported and is being addressed.

[43] The http CRL URL will be downloaded extremely frequently. To allow for web caching of the CRL, it is RECOMMENDED that the web server return a 200 response to the HTTP GET request, and not a 302 redirection, since such an answer it is not normally followed by clients or cached by web caches [RFC2616]. It is RECOMMENDED that the CRL be labelled with the correct MIME document type.

3.3.12 subjectAlternativeName, issuerAlternativeName

The *subjectAlternativeName* extension SHOULD be present for server certificates (including "host" and "service" certificates in the grid context), and, if present, MUST contain at least one FQDN in the *dNSName* attribute. If an end-entity certificate needs to contain an rfc822 email address, this rfc822 address SHOULD be included as an *rfc822Name* attribute in this extension only.

For use with web server certificates, multiple FQDNs *dNSName* attributes can be added to allow name-based virtual hosting of secured web sites[44].

3.3.13 authorityInformationAccess

The *authorityInfoAccess* extension is the proper place to refer to any OCSP service that the issuer recommends validating software to used. There is no grid software today that uses this extension, but including it does not interfere with correct operations.

It is RECOMMENDED to include this extension if the issuer operates a production-quality OSCP service. The extension MUST NOT be included if the value points to an experimental or non-monitored service, as this will impair operations as soon as an OCSP client is implemented and enabled in the software.

The extension MAY also contain a CRL URI, as described in RFC4325, or the location of any higher-level CA certificates, but it should be noted that regardless, a CRL http URL MUST also be included in the *cRLDistributionPoints* extension.

The extension MUST NOT be marked critical.

[44] See also footnote to section 3.4.3.

4. General Considerations

4.1 ASN.1 Structure of the DN and ordering of the RDN components

The subject and issuer distinguished Names (DNs) consist of a sequence (an order-preserving list) of Relative DN (RDN) components sets. As stated in the preceding sections, the length of any RDN set MUST be equal to one (1). There has, however, not been definitive guidance on the way the RDN components should be ordered in the DN sequence, neither from the X.500 document series (specifically X.521 [X521]), nor from sources such as the X.509 Style Guide [PG2000].

The definition of the Name in X.501 [X501] defines it as a SEQUENCE OF RelativeDistinguishedName, where the SEQUENCE OF is an ASN.1 construct that in the DER encoding should be written out "as-is" in the order in which it is presented. It should not be re-ordered for interpretation[45].

```
Name ::= SEQUENCE OF RelativeDistinguishedName


RelativeDistinguishedName ::= SET OF AttributeValueAssertion

      AttributeValueAssertion ::= SEQUENCE {

          attributeType OBJECT IDENTIFIER,

          attributeValue ANY

}
```

The *SEQUENCE* of *RelativeDistinguishedNames* SHOULD start with the least-varying component (i.e. the static prefix) of the *distinguishedName* for all issuer and subject names, and MUST start with the least-varying component for any names issued by an issuing authority that issues end-entity certificates, or three or more trusted subordinate authorities[46].

---

[45] This ordering applies for comparisons based on the ASN.1 structure. The representation of that ASN.1 SEQUENCE as a string is subject to many discussions and conflicting solutions, as is testified to by the long debates regarding the represention returned by the OpenSSL X509_one_line function and the string representation defined in RFC2253.

[46] Discussions around the successor to RFC 3280 have included statements that the SEQUENCE ought to start with the Country or a domainComponent (still in draft). Formerly, it could only be deduced from the examples, and the unclear guidance *"In theory it should be a full, proper DN, which traces a path through the X.500 DIT"*, which usually interpreted "trace" as "start at the root of the tree".

Starting the sequence with the commonName does create problems in, e.g., wildcard matching in the signing policy file, and other places that do prefix-only matching, or in pattern matching where a wildcard can only appear at the 'end' of a string pattern.

The 'reverse' ordering of the sequence is theoretically not malformed, but causes significant problems with grid software. The 'reverse' ordering starts the sequence with the commonName (as is apparent from the output of the asn1parse OpenSSL command). Some established issuers that do not issue end-entity certificates (e.g. the SwissSign intermediate CAs) may continue to issue 'reversed' names, as they are in wide-spread use and the list of issued subject names is small and can be enumerated. However, no large numbers (three or more) of trusted subordinate CAs can be accommodated by enumeration in the namespace constraints policy files used in grid operations. Note that, in the case of SwissSign, they have changed and now allow the SWITCH CA to issue end-entity certificates in the "other" ordering for grid use.

4.2 Keys, key lengths and hashes

As explained in NIST publication 800-57, 1024-bit RSA keys are equivalent in strength to 80-bit symmetric keys, 2048-bit RSA keys to 112-bit symmetric keys and 3072-bit RSA keys to 128-bit symmetric keys [SP800-57]. RSA claims that 1024-bit keys are likely to become crackable between 2006 and 2010 and that 2048-bit keys are sufficient until 2030 [RSA03]. An RSA key length of 3072 bits should be used if security is required beyond 2030. NIST key management guidelines further suggest that 15360-bit RSA keys are equivalent in strength to 256-bit symmetric keys[47].

Similar considerations hold for the hash functions used, with the MD5 hash function known to have collisions, and SHA-1 having been shown to provide less than 80 bits of security. Since more modern hash functions (such as SHA-256) are not yet widely supported, there is no ready alternative and SHA-1 is recommended.

4.3 Maximum key lengths

Note that key lengths of 4096 bits or more give complications with many applications and libraries. The standard JCE Java crypto libraries provided with SUN Java versions up to and including 1.4.2 cannot handle 4096 bit keys. Although a workaround is available[48], use of 4096-bit keys is NOT RECOMMENDED for use in 2006. This should be re-evaluated in 2007.

[47] See also www.keylength.com for a comprehensive overview.

[48] http://codelabs.ru/grid/java-4096.txt

5. Examples and background information

The meaning of several common attributes used in extensions is not necessarily always clear. Although comprehensive descriptions exist[49], it is considered appropriate to repeat some of this information here. Only extensions that are a common source of confusion or that have special application characteristics in grid software are discussed.

This section does not contain normative text.

5.1 Examples of directory names

A typical issuer distinguished name that is compliant to the guidelines given in this document could be:

RFC2253 string representation	CN=My Authority 1, O=MyOrg Authorities, DC=example, DC=org
OpenSSL oneline representation	/DC=org/DC=example/O=MyOrg Authorities/CN=My Authority 1

ASN.1 sequence	SEQUENCE
	  SET
	  SEQUENCE
	  OBJECT :domainComponent
	  PRINTABLESTRING :org
	  SET
	  SEQUENCE
	  OBJECT :domainComponent
	  PRINTABLESTRING :example
	  SET
	  SEQUENCE
	  OBJECT :organization
	  PRINTABLESTRING :MyOrg Authorities
	  SET
	  SEQUENCE
	  OBJECT :commonName
	  PRINTABLESTRING :My Authority 1

RFC2253 string representation	CN=My Authority 1, O=MyOrg Authorities, C=lu
OpenSSL oneline representation	/C=lu/O=MyOrg Authorities/CN=My Authority 1

ASN.1 sequence	SEQUENCE
	  SET
	  SEQUENCE
	  OBJECT :country
	  PRINTABLESTRING :lu
	  SET
	  SEQUENCE
	  OBJECT :organization
	  PRINTABLESTRING :MyOrg Authorities
	  SET
	  SEQUENCE
	  OBJECT :commonName
	  PRINTABLESTRING :My Authority 1

[49] See for instance: *Aufbau und Betrieb einer Zertifizierungsinstantz*, DFN Bericht 79, and especially Chapter 8. http://www.dfn-cert.de/dfn/berichte/db089/

For expressing these in OpenSSL, e.g., http://www.math.ias.edu/doc/openssl-0.9.7a/openssl.txt

While for an end-entity named “Jürgen Schmidt”, the following name forms could be used:

RFC2253 string representation	CN=Juergen Schmidt 90210, DC=example, DC=org
OpenSSL oneline representation	/DC=org/DC=example/CN=Juergen Schmidt 90210

ASN.1 sequence	SEQUENCE
	  SET
	  SEQUENCE
	  OBJECT :domainComponent
	  PRINTABLESTRING :org
	  SET
	  SEQUENCE
	  OBJECT :domainComponent
	  PRINTABLESTRING :example
	  SET
	  SEQUENCE
	  OBJECT :commonName
	  PRINTABLESTRING :Juergen Schmidt 90210

RFC2253 string representation	CN=Juergen Schmidt 90210, O=ExOrg B.V., C=nl
OpenSSL oneline representation	/C=nl/O=ExOrg B.V./CN=Juergen Schmidt 90210

ASN.1 sequence	SEQUENCE
	  SET
	  SEQUENCE
	  OBJECT :country
	  PRINTABLESTRING :nl
	  SET
	  SEQUENCE
	  OBJECT :organization
	  PRINTABLESTRING :ExOrg B.V.
	  SET
	  SEQUENCE
	  OBJECT :commonName
	  PRINTABLESTRING :Juergen Schmidt 90210

5.2 cRLDistributionPoints extension

The *cRLDistributionPoints* extension should contain a list of locations where the actual CRL data is stored, for example a URL with the http location of the CRL itself. These URIs should *not* point to just the index file, but to the actual CRL, like:

```
X509v3 CRL Distribution Points:
      URI:http://www.example.org/ca/cacrl.pem
```

and preferably return a direct answer, and not a 302 HTTP redirect.

6. Security Considerations

The correct and complete interpretation of any and all parts of a certificate is essential to maintain integrity of the system that relies on them. Inconsistencies in name ordering and representation, as well as the use of non-standard attributes and extensions that are not well tested with the validation software and subsequent authorisation systems may leave holes in a deployment of a grid certificates. Where such adverse interactions are known, they have been highlighted in the corresponding sections of this document. However, the absence of any such warnings may not be construed as to mean that no security issues exist.

Acknowledgements

The editors are grateful for the contributions made to this document by the members of the International Grid Trust Federation [IGTF].

Author Information

David L. Groep

davidg@nikhef.nl

Michael Helm

helm@fionn.es.net

Intellectual Property Statement

Full Copyright Notice

by removing the copyright notice or references to the GGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the GGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the GGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

References

[RFC1591] RFC1591: Domain Name System Structure and Delegation, J. Postel, ed. 1994

[RFC3280] RFC3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, R. Housley et al. ed., 2002

[RFC2616] RFC2616: Hypertext Transfer Protocol -- HTTP/1.1, R. Fielding et al., 1999

[RFC3820] RFC3820: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile, S. Tuecke et al., 2004

[FR] FreeRadius, http://www.freeradius.org/

[PG2000] Peter Gutmann, *X.509 Style Guide*, 2000, http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt

[X501] IT-OSI-The Directory: Models, Recommendation X.501

[X521] IT-OSI-The Directory: Selected Object Classes, Recommendation X.521

[SP800-57] Recommendation for Key Management, NIST Special Publication 800-57 (draft), August 2005

[RSA03] Burt Kaliski, TWIRL and RSA key sizes, RSA Laboratories, May 2003, http://www.rsasecurity.com/rsalabs/node.asp?id=2004

[IGTF] International Grid Trust Federation, http://www.gridpma.org/