GWD-R.125bis DRAFT
CAOPS-WG

125bis version ~~6~~7

David L. Groep, Nikhef*
Mike Jones, University of Manchester*
Michael Helm, LBNL/ESNet
Jens Jensen, RAL/STFC
Milan Sova, CESNET
Scott Rea, DigiCert Inc.
Reimer Karlsen-Masur, DFN
Ursula Epting, KIT

~~October 2012~~June 2013

Grid Certificate Profile

Status of This Document

This document provides **recommendations** to the Grid community. Distribution is unlimited.

Obsoletes

This document obsoletes GFD-C.125.

Copyright Notice

Abstract

This document provides guidance for the use of directory names, attributes, and extensions in X.509 certificates, such that they are usable by the majority of the grid infrastructures today. The intended audience for this document includes issuers of X.509 certificates for use in grid infrastructures, and implementers of X.509 validation software for grid purposes.

Interoperability for X.509 identity certificates between the issuers of certificates and the software that interprets them is increasingly more important as the number of participants in grids that rely on a X.509 certificates grows. It is difficult to predict which particular software will be used by the parties relying on the certificate, and how this software interprets specific name forms, attributes, and extensions. This document gives guidance and defines explicit restrictions on the certificate profile to ensure the certificate is interpreted by the relying party in the way the issuer intended. It specifies and further restricts the certificate format as defined in RFC5280 and the X.509 standard.

This document extends the guidance in GFD.125 by specifying additional constraints and providing further clarification.

Contents

1. Scope of this document

This document provides guidance for the use of attributes and extensions in X.509 [X509] certificates such that they are usable by the majority of the grid infrastructures today. This guidance must be interpreted in the context of RFC 5280 [RFC5280], *i.e.*, all certificates must be compliant to RFC 5280 in addition to any limitations imposed by the guidelines 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~~meaning the extension or attribute is either ~~useful or~~ harmless or useful; 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 or 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 [RFC2119].

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.

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. The current most secure hash function that is supported by the entire target audience of the CA SHOULD be used. In particular SHA-2 or better SHOULD be used and at least as strong as SHA-1 MUST be used[1].

2.2 Serial Number

The serial number of each CA certificate SHOULD be unique among all certificates representing that CA[2].

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. The current most secure hash function that is supported by the entire target audience of the CA SHOULD be used. In particular SHA-2 or better SHOULD be used and at least as strong as SHA-1 MUST be used[3].~~

2.3 Issuer and Subject names

Only a limited number of attribute types are well supported by 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*~~*countryName* (C), ~~*state*~~*stateOrProvinceName* (ST), ~~*locality*~~*localityName* (L), ~~*organization*~~*organizationName* (O), ~~*organizationalUnit*~~*organizationalUnitName* (OU) and *commonName* (CN). Use of other

---

[1] Note that modern hashes, such as SHA-256, are supported in recent versions of the majority of software, (such as OpenSSL in use,version 0.9.8 and above) so SHA-1 is no longer the only available value as ofat the time of writing.

[2] If a root or intermediate CA certificate is re-issued with the same serial number – for example in the case that only the lifetime is extended but the key pair remains the same – web browsers using the Mozilla NSS code base will issue a user warning and the import will fail (tested in Spring 2007), but if installation of the new certificate is attempted in Microsoft Internet Explorer it will overwrite the old one (tested in versions up to and including version 6). For Internet Explorer 7~~,~~ and later (and verified up till IE version 9), both certificates will be in the trust store, but the most recently imported certificate will always be used. Thus, for NSS-based browsers the old certificate has to be removed from the certificate store first,and for IE7+ that is advised.

If the serial number is changed, the process of importing the new root certificate into Microsoft Internet Explorer will result in both certificates being retained in the certificate store, and the original one is not overwritten (tested in versions up to and including version 6). For version 7, this behaviour will also occur when the same serial number is used.

~~[3] Note that modern hashes, such as SHA-256, are supported in recent versions of the majority of software, (such as OpenSSL in use,version 0.9.8 and above) so SHA-1 is no longer the only available value as ofat the time of writing.~~

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 ASN.1 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*~~*organizationName* 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*~~*organizationName* (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 much of the known grid middleware uses the entire DN for subject identification. If no O or OU attributes appear in the DN, browsers[4] might not use other components to show affiliation.

All Relative Distinguished Name (RDN) components in distinguished names MUST be compliant with [RFC4630] and in addition SHOULD be encoded as PrintableString. If an UTF8String is used for encoding, the RDN MUST NOT contain characters that cannot be expressed in printable 7-bit ASCII, as these characters have inconsistent representations.

Issuer and authority subject name RDN ~~components~~component recommendations	
Required	commonName
Advised to use	domainComponent, ~~Organization~~organizationName
Optional	~~Country, State, Locality, OrganizationalUnit~~countryName, stateOrProvinceName, localityName, organizationalUnitName
~~Not to~~Should not be used	serialNumber, userID, uniqueIdentifier, emailAddress

2.3.1 *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~~prominently 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~~.~~[5][6]. In addition the use of the "O" entry is encouraged.

2.3.2 *DomainComponent, ~~country, organization, organizationalUnit~~countryName, organizationName, organizationalUnitName*

The distinguished name is usually made up of a combination of the attribute types "DC", "C", "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

[4] In particular this applies to browsers based on the Mozilla NSS code base.

[5] ~~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.~~

[6] 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.

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*~~*organizationName* O attribute in the DN is encouraged.

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

2.3.3 serialNumber

The attribute type *serialNumber* {2.5.4.5} MUST NOT be used in any Name[7].

2.3.4 emailAddress

The attribute type *emailAddress* MUST NOT be used in DNs. It has been ~~depricated~~deprecated in RFC 5280, in favour of having an ~~*rfc822EmailAddress*~~*rfc822Name* in the *subjectAlternativeName* X.509v3 extension, and many recent mail clients can deal with *subjectAlternativeName*.[8]

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.

2.3.5 userID or uid

The attribute type *userID* or *uid* {0.9.2342.19200300.100.1.1} MUST NOT be used in Names. The attribute *uniqueIdentifier* {2.5.4.45} MUST NOT be used in Names. Additionally, it is not relevant for CA certificates of any kind.[9]

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.

---

[7] The *serialNumber* attribute was originally intended to describe the serial number of a device [X.520]. There have been discussions 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 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.

[8] 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.

[9] 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 4514 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 4514. The *uniqueIdentifier* attribute, with OID 2.5.4.45, has been string encoded in OpenSSL as “uid”, also colliding with the *userID* attribute name.

Issuer and authority subject extensions and attribute recommendations ~~**Required**~~	~~basicConstraints~~
Required	basicConstraints, keyUsage
Advised to use	AuthorityKeyIdentifier, SubjectKeyIdentifier
Optional	for all CAs: cRLDistributionPoints, for subordinate CAs: certificatePolicies, authorityInformationAccess,
~~**Not to**~~**Should not be used**	extendedKeyUsage, nsPolicyURL, nsRevocationURL, nsComment, nsCertType, nameConstraints (for grid-only CAs)

Deleted Cells

2.4.1 basicConstraints

The *basicConstraints* extension MUST be included in CA certificates, and it MUST be set to "CA: TRUE". This extension SHOULD be marked as critical[10].

2.4.2 keyUsage

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

For a CA certificate, *keyCertSign* MUST be set, and *crlSign* MUST be set if the CA certificate is used to directly sign issued CRLs[11].

It is RECOMMENDED to set no more than these two values[12]. For proper operation it is not required to have more than *keyCertSign* and *cRLSign* in the CA certificate and adding additional values may convey an impression to relying parties that the CA certificate is used for purposes other than signing and issuing certificates and related signing services. The CA thus ensures that the permitted use of public keys is minimal and relevant to the goals of its PKI, particularly for its own public key (in the CA certificate) [13].

---

[10] While the criticality is intended for a CA to make the use of its certificates more robust, not all verification systems currently in use (specifically those outside of the grid context proper) do not factor the criticality of many of these extensions. Especially for CAs that serve a wider community, marking *basicConstraints* as critical may break other applications, which is the reason it is not marked as such in a sizeable fraction of the CA certificates preinstalled in browsers (as of September 2007, the root store in Microsoft Windows XP contained 85 out of 200 CAs that were not compliant [Netrust2007]). For new CAs that do not face known incompatibilities, it is strongly recommended to set *basicConstraints* and mark it critical.

[11] There may be CAs that either do not issue CRLs at all, since their end-entity certificates have a short ~~life time~~lifetime, or that use indirect CRLs. Note that indirect CRLs have not been extensively tested, and are not currently supported by OpenSSL. There is also no direct way to create such an end-entity certificate in ~~the many~~some CA products, such as the Sun One/Iplanet CMS, although direct generation of the ASN.1 is always a possibility. Grid middleware currently cannot use indirect CRLs.

[12] If OCSP responses are directly signed by the CA certificate, then *DigitalSignature* MAY be added to the *keyUsage* extension, since future discussions in the IETF PKIX group may lead to this *keyUsage* being required to validate the OCSP responses.

[13] A CA can limit permitted use by defining acceptable and unacceptable uses in the policy statements, 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,

2.4.3 *extendedKeyUsage*

The *extendedKeyUsage* extension SHOULD NOT be included in CA certificates[14]. If present, it MUST NOT be marked critical.

2.4.4 *authorityInfoAccess*

The *authorityInfoAccess* (AIA) extension for subordinate CAs MAY include OCSP information[15] and issuing CA location.nsCertType, nsComment, nsPolicyURL, nsRevocationURL

The *ns** attributes are deprecated and MUST NOT be included in any new CA certificates.[16].

2.4.5 *certificatePolicies*

The presence of a *certificatePolicies* extension is not harmful, but adding this extension in self-signed ~~roots~~root CA certificates permanently binds this CA certificate to the particular instance of the policies referenced and is thus not advisable[17]. The *certificatePolicies* extension MAY be set for subordinate CAs and if set MUST include only policy OIDs.

If present, it SHOULD NOT be marked critical.

2.4.6 *cRLDistributionPoints*

The *cRLDistributionPoints* (CDP) extension ~~need not~~MAY be present in a self-signed root CA certificate, but MUST be included in end-entity certificates and SHOULD be included in any intermediate CA certificates[18] that issues CRLs.

For subordinate CAs, where a CDP is present, it MUST contain at least one http URI[19].

---

it can adversely affect the CA's name, reputation, or operations, and, ultimately, the most precious thing it has - trust.

[14] *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 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.

[15] Running an OCSP responder, according to current best practices, is recommended and it should be run as a highly-available service on a 24x7 basis. If such a production OCSP responder is available, its access information SHOULD be included in the AIA extension. If no highly-available OCSP service is present, there SHOULD NOT be an OCSP end-point included in the AIA extension.

[16] If adding explicit text to the certificate is desired, 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~~RFC5280 RECOMMENDS that only an OID is used in the *certificatePolicies* extension. Also, compliant ~~RFC3280~~RFC5280 implementations SHOULD actually display each and every user notice to the user.

[17] Any change in the policy requires re-issuing the CA certificate with an updated extension, and re-issuing and re-distributing a CA certificate is a complicated operation. It is therefore advisable to put only long-term stable extensions in a CA certificate.

[18] 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.

[19] The URI should be plain *http*, and in particular not an *https*. Although the https connection in theory does not need to be validated, many client tools do this by default and will fail in absence of proper certificate, especially if the web site is not secured with a certificate issued

2.4.7 Authority and Subject Key Identifier

A *subjectKeyIdentifier* extension MUST be included in CA certificates to aid in validation path construction ~~and~~. An *authorityKeyIdentifier* MUST be included in all CA certificates~~,~~ unless the certificate is self-signed[20]. If included for a self-signed root certificate~~,~~ the *authorityKeyIdentifier*'s *subjectKeyIdentifier* and *subjectKeyIdentifier* MUST be the same.

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

2.4.8 nameConstraints

The extension *nameConstraints* (OID 2.5.29.30) is not relevant for grid purposes today and its use is NOT RECOMMENDED[21].

by the CA itself. The CRL returned is signed and integrity protected anyway. The *cRLDistributionPoints* extension MAY contain other URIs.

[20] Not including the *subject*- or *authorityKeyIdentifier* is not known to break any grid software.

[21] The interpretation of the *nameConstraints* extension varies significantly between implementations and therefore SHOULD be avoided in CA certificates, and is not relevant for end-entity certificates.

Note that this applies to CA-defined namespace constraints, and this is completely independent of any constraints on the subject signing namespace to be defined by the relying party, and which is to be independently enforced by software (for example via 'dot-*signing_policy*' files in the Globus Toolkit software).

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.

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. The current most secure hash function that is supported by the entire target audience of the CA SHOULD be used. In particular SHA-2 or better SHOULD be used and at least as strong as SHA-1 MUST be used.

3.2 Serial number

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

~~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 supported by the majority of OpenSSL versions in use, so SHA-2 is currently RECOMMENDED if the software in the entire community supports it. At least a SHA-1 hash or stronger MUST be used.~~

~~*3.2*~~*3.3 Subject distinguished names*

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

Relative Distinguished Name (RDN) attribute types other 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*~~*3.3.1 String encoding of the RDN components*

All Relative Distinguished Name (RDN) components in distinguished names MUST be compliant with [RFC4630] and in addition SHOULD be encoded as PrintableString. If an at-sign ("@") is included, IA5String SHOULD be used for encoding. If an UTF8String is used for encoding, the RDN MUST NOT contain characters that cannot be expressed in printable 7-bit ASCII, as these characters have inconsistent representations[22].

~~*3.2.2*~~*3.3.2 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[23], colon,

[22] 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.

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

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).

The double quote MUST NOT be used.

The single quote SHOULD NOT be used[24]. The colon (“:”) SHOULD NOT be used[25].

The CA MUST ensure that case or consecutive spaces are not used to distinguish between users (e.g. users with the same name)[26].

***End-entity* subject name RDN ~~components~~component recommendations**	
Required	commonName
Advised to use	domainComponent, ~~Organization~~organizationName
Optional	~~Country, State, Locality, OrganizationalUnit~~countryName, stateOrProvinceName, localityName, organizationalUnitName
~~Not to~~Should not be used	serialNumber, emailAddress, userID (*also known as ‘uid’*), uniqueIdentifier (*also known as ‘uid’*)

~~3.2.3~~3.3.3 *commonName*

A *commonName* attribute MUST be used in the subject DN of an end-entity certificate[27].

If the *commonName* is not encoded as printableString, it SHOULD be encoded as IA5String or 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[28]. It is usually allowed for an entity to have more than one subject DN assigned[~~29~~].

[24] ~~The quote characters must not be used because~~ OpenSSL follows RFC1778’s definition of PrintableString.

[25] The COLON (“:”) character is used as a field separator in ‘htpassword’ files with FakeBasicAuth as used in Apache mod_ssl and cannot be escaped in that format. Subjects with a colon in their DN will not be listable in this file format.

[26] 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.

[27] Many browsers use only the *commonName* to label certificates in their certificate stores. It should be noted that past versions of the FreeRadius (http://www.freeradius.org/) uses only the *commonName* for its authorization decision. No grid middleware is known to act in this manner.

[28] 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.

[~~29~~] ~~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.~~

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*[30].

Some grid middleware[31] contains a design flaw that allows implicit wildcard matching of the ~~domainname~~domain name in the *commonName* attribute, where the first component of the ~~domainname~~domain name containing a dash ("-") is stripped of all characters from the dash onwards, and then matched to the FQDN in the *commonName*[32].

Note that for name-based virtual hosting, additional FQDNs can be asserted in the *subjectAlternativeName* extension in multiple *dNSName* attributes[33].

~~3.2.4~~3.3.4 *domainComponent (DC), ~~country~~countryName (C), ~~State~~stateOrProvinceName (ST), ~~Locality~~localityName (L), ~~Organization~~organizationName (O),* and ~~*OrganizationalUnit*~~*organizationalUnitName (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 customary to encode the *domainComponent* as an IA5String[34]. Since all known software correctly parses all incoming encodings, all of PrintableString, IA5String and UTF8String MAY be used to encode *domainComponent*, with IA5String being preferred, and the characters 0-9, a-z, A-Z, '-' (hyphen) and '_' (underscore) allowed.

If the ~~*Country*~~*countryName* attribute is used, the value of this attribute SHOULD contain the two-letter ISO3166 encoding of the country's name[35, 36]. The

---

[30] 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.

[31] This refers in particular to the Globus Toolkit, at least up to and including version ~~4.2~~5.

[32] 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.

[33] 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.

[34] The latest OpenSSL and the RedHat Certificate System versions encode the *domainComponent* attribute as an IA5String, OpenSSL 0.9.7c and older 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.

[35] The designation UK is an well-known exception, mainly for historical reasons – GB is the official ISO 3166-1 representation for the United Kingdom of Great Britain and Northern Ireland, although in many contexts the designation "UK" is used for the same. Both GB and UK MAY be used as designations. Note that the Ukraine MUST be encoded as UA.

[36] In case the ~~*country*~~*countryName* (C) is used as part of the varying part of the subject distinguished name (i.e., it is not part of the constant DN prefix that defines the issuing namespace), the ~~*country*~~*countryName* (C) asserted in the subject DN of an end-entity certificate SHOULD correspond the home country of the end-entity, and thus does not

~~*country*~~*countryName*, if used, MUST be used at most once. Any of the ~~*State*~~*stateOrProvinceName* *(ST),* ~~*Locality*~~*localityName* *(L),* ~~*Organization*~~*organizationName* *(O),* and ~~*OrganizationalUnit*~~*organizationalUnitName* *(OU)* attributes MAY be used and have their usual meaning.

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

~~3.2.5~~3.3.5 *serialNumber*

The AttributeType "*serialNumber*" (i.e. {2.5.4.5}) MUST NOT be used in any Name[37].

Specifically, the *serialNumber* attribute MUST NOT be used to re-encode the certificate serial number in the subject name[38].

~~3.2.6~~3.3.6 *emailAddress*

The attribute *pkcs9email* (“emailAddress”) MUST NOT be used in subject names[39].

If used, ~~by~~ RFC5280 email addresses MUST be encoded in RFC822 “addr-spec” format (section 6.1) and they MUST be encoded as IA5String.

~~3.2.7~~3.3.7 *userID* and *uniqueIdentifier*

The attribute type “*userID*” (i.e. OID {0.9.2342.19200300.100.1.1}) and *uniqueIdentifier* (i.e. OID {2.5.4.45}) MUST NOT be used in Names[40]. Both attribute types are also known as *uid*.

~~3.3~~3.4 *Extensions in end-entity certificates*

For use of an end-entity certificate with grid software, at least either of the *extendedKeyUsage* or *nsCertType*[41] 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 *subjectAlternativeName* extensions with a *dNSName* attribute is RECOMMENDED. For end-entity certificates that include an rfc822-style email address, the *subjectAltName* extension SHOULD be used, and the email address included in the *rfc822Name* attribute.

End-entity certificates MUST include the *keyUsage* extension and it is RECOMMENDED that an end-entity certificate also includes ~~also~~ the extensions *certificatePolicies*, and *cRLDistributionPoints*.

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

---

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*~~*countryName* attribute should not be part of a unique subject DN naming prefix.

[37] See footnote 5 to section 2.3.3 for ~~the argumentation~~clarification.

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

[39] The *emailAddress* attribute in the subject DN has been deprecated in RFC5280 [RFC5280], in favour of having an ~~*rfc822EmailAddress*~~*rfc822Name* 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, whereas other ~~products reder~~software renders it as “E”, or as the numeric OID.

[40] See footnote 7 to section 2.3.5 for ~~the argumentation~~clarification.

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

***End-entity subject extensions and attribute recommendations*~~Required~~**	~~keyUsage~~
Required	keyUsage, extendedKeyUsage
Advised to use	basicConstraints, cRLDistributionPoints, certificatePolicies, subjectAlternativeName*, authorityInfoAccess
Optional	~~,~~ authorityKeyIdentifier, subjectKeyIdentifier, ~~,~~ issuerAlternativeName
~~**Not to**~~**Should not be used**	nsCertType

Deleted Cells

~~3.3.1~~3.4.1 *basicConstraints*

The *basicConstraints* extension is RECOMMENDED to be included in end-entity certificates[42]. The *pathLenConstraint* attribute MUST NOT be present[43].

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~~3.4.2 *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 *nonRepudiation* (*contentCommitment*) 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. It SHOULD NOT be set in other end-entity certificates either, as the claims made by this *keyUsage* are ill-defined or non-verifiable, and its interpretation by clients unclear. If it is set regardless, its assertion in personal end-entity certificates SHOULD be limited to special purposes.
>
> The *dataEncipherment* value is RECOMMENDED in order to enable use of the certificates with specific implementations of message-level security mechanisms where messages are to be encrypted[44].

---

[42] 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.

[43] Note that RFC5280 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[45].

~~3.3.3~~3.4.3 *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 the 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*[46]~~.~~ [47].

OCSP responder certificates MUST have *oCSPResponder* asserted.

~~3.3.4~~3.4.4 *nsCertType*

This extension is deprecated ~~and~~. It MUST NOT be used in new certificates~~, and~~; the appropriate equivalent values SHOULD be ~~included~~expressed in the *extendedKeyUsage* extension[48].

~~3.3.5~~3.4.5 *nsPolicyURL, nsRevocationURL*

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

~~3.3.6~~3.4.6 *nsComment*

This attribute is deprecated and ~~is not required~~SHOULD NOT be used in end-entity certificates[49]. If it is included, this extension MUST NOT be marked critical.

---

[44] The *dataEncipherment* usage is intended to refer to the direct use of the RSA key in enciphering data, and as such ought to bear no relevance to the encryption of documents with a session key, however some web services stacks to date require this usage to be set in order to use the certificate for use in XML encryption and message-level security. This has been verified for exchanging encrypted messages via GSISecureMessage as implemented in the Globus Toolkit middleware. This includes the receiving entity's certificate that must have the dataEncipherment keyUsage extension set if keyUsage itself is set to be a critical extension.

[45] See also section 2.4.2.

[46] 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.

[47] ~~Refer to Chapter 5 for all values that could be included in certificates.~~ The list of common values as represented in OpenSSL includes “serverAuth”, “clientAuth”, “codeSigning”,”emailProtection”,and ”timeStamping”. In addition, Netscape and Microsoft specific values may be asserted, as well as numerically expressed OIDs.

[48] The *extendedKeyUsage* and *nsCertType* extensions are interrelated and do partially cover the same purposes. Either of these has to be present to ensure correct operation of grid and other software[48], and nsCertType MUST NOT be used. For example for certificates issued to a Unicore NJS service, the *nsCertType* can be set to “*server, client”* but the preferred way to expressing this is by setting eKU to “*serverAuth, clientAuth”*.

[49] 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.

~~3.3.7~~3.4.7 *cRLDistributionPoints*

The *cRLDistributionPoints* extensions MUST be present in end-entity certificates, and MUST contain at least one http URI (i.e., not an *https* URI) although it may contain other URIs[50] [51] [52]. It MUST return the CRL in DER encoded form.

Some software[53] is ~~known not to be able~~unable to handle any values other than a single URI in this extension.

It is RECOMMENDED that the reply returned at the http URI is cacheable[54].

~~3.3.8~~3.4.8 *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 RFC5280 (i.e. is purely derived from the public key value), it will not impair smooth replacement.

~~3.3.9~~3.4.9 *subjectKeyIdentifier*

The *subjectKeyIdentifier* extension MUST NOT be marked critical.

~~3.3.10~~3.4.10 *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.

~~3.3.11~~3.4.11 *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[55].

---

[50] See also footnotes to section 2.4.5.

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

[52] The *cRLDistributionPoints* extension should contain (a list of) locations where the actual CRL data is stored, e.g. `URI:http://www.example.org/ca/cacrl.der`. The data retrieved must be the actual CRL. Preferably it returns a direct answer and not a 302 'HTTP redirect', in order to allow caching of the results.

[53] This defect is only known to apply to VOMS and VOMS-Admin, at least up to and including VOMS version 1.7.

[54] 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~~3.4.12 *authorityInformationAccess*

The *authorityInformationAccess* extension is the proper place to refer to any OCSP service that the issuer recommends validating software to used.

It is RECOMMENDED to include this extension if the issuer operates a production-quality OCSP service. The extension SHOULD NOT be included unless it points to a highly-available service.

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 URI MUST also be included in the *cRLDistributionPoints* extension.

The extension MUST NOT be marked critical.

[55] 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[56].

```
Name ::= SEQUENCE OF RelativeDistinguishedName


RelativeDistinguishedName ::= SET OF AttributeValueAssertion

      AttributeValueAssertion ::= SEQUENCE {

          attributeType OBJECT IDENTIFIER,

          attributeValue ANY

}
```

Since many authorization applications and namespace constraining policies are based on wildcard matching of only the trailing part of an OpenSSL one-line string representation rendering of the *Name*, 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[57].

---

[56] 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 RFC4514.

[57] Discussions around the successor to RFC 3280 have included statements that the SEQUENCE ought to start with the ~~Country~~countryName 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 special 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[58]. As other digital signature and key exchange algorithms are introduced, such as elliptic curve mechanisms, their use should be considered for new certificates provided the entire target audience is capable of dealing with such mechanisms[59].

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. Thus, 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. The most secure hash function that is current supported by the entire target audience of the CA SHOULD be used, but at least SHA-1 or stronger MUST be used[60], with SHA-2 being recommended.

4.3 Maximum key lengths

RSA keys longer than 8192 bits have not been evaluated in production deployments. No EC keys have been evaluated in these environments either.

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

[59] At of time of writing, only RSA algorithms are sufficiently well supported in clients. It is thus NOT advisable to select non-RSA algorithms.

[60] Note that modern hashes, such as SHA-256, are not supported by the majority of OpenSSL versions in use, so SHA-1 is the only available value as of time of writing.

5. Directory Names and String Representations

Although comprehensive texts on the creation of certificate authorities and the configuration of particular CA software exist[61], it is considered appropriate to repeat some of this information here. In particular, the ordering of Relative Distinguished Name (RDN) components in a Directory Name and the string representation thereof remains a source of frequent mistakes. An example of the relation between the ASN.1 DN and its various string representations is given below. This section does not contain normative text.

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

RFC4514 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
	  IA5STRING :org
	  SET
	  SEQUENCE
	  OBJECT :domainComponent
	  IA5STRING :example
	  SET
	  SEQUENCE
	  OBJECT :organization
	  PRINTABLESTRING :MyOrg Authorities
	  SET
	  SEQUENCE
	  OBJECT :commonName
	  PRINTABLESTRING :My Authority 1

RFC4514 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

[61] 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:

RFC4514 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
	  IA5STRING :org
	  SET
	  SEQUENCE
	  OBJECT :domainComponent
	  IA5STRING :example
	  SET
	  SEQUENCE
	  OBJECT :commonName
	  PRINTABLESTRING :Juergen Schmidt 90210

RFC4514 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

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.

Contributors

This document captures the collective knowledge of many people, and the editors are grateful for the essential contributions made to this document by the members of the International Grid Trust Federation (IGTF, see http://www.gridpma.org/), the individual certification authorities and their staff, and relying parties that have conducted the experiments and tests, and the contributions from the participants in the CAOPS WG.

For the editors,
David L. Groep (davidg@nikhef.nl)

Intellectual Property Statement

Disclaimer

Full Copyright Notice

languages other than English. OGF, with the approval of its board, may remove this restriction for inclusion of OGF document content for the purpose of producing standards in cooperation with other international standards bodies.

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

References

[Netrust2007] Vivek Kaushik *Digital Certificate Extensions: Should "Basic Constraints" Be Marked Critical?*, Netrust Pte Ltd, 2007. (http://www.netrust.net/BasicConstraints_whitepaper_v1.0.pdf, visited October 18, 2007)

[PG2000] Peter Gutmann, *X.509 Style Guide*, October 2000, (http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt, visited October 2007)

[RFC1591] J. Postel (ed.) *Domain Name System Structure and Delegation*, RFC 1591, 1994

[RFC2119] S. Bradner *Key words for use in RFCs to Indicate Requirement Levels*, RFC 2119, 1997

[RFC2616] R. Fielding et al. *Hypertext Transfer Protocol -- HTTP/1.1*, RFC 2616, 1999

[RFC5280] D. Cooper et al. (ed.) *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 5280, 2008

[RFC3820] S. Tuecke et al. *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*, RFC 3820, 2004

[RFC4514] K. Zeilenga *Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names*, RFC 4514, 2006

[RFC4630] R. Housley et al. *Update to DirectoryString Processing in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 4630, 2006

[RSA03] Burt Kaliski, *TWIRL and RSA key sizes*, RSA Laboratories, May 2003, (http://www.rsasecurity.com/rsalabs/node.asp?id=2004, visited October 2007)

[SP800-57] Elaine Barker et al. *Recommendation for Key Management*, NIST Special Publication 800-57, August 2005

[X501] ISO/IEC JTC 1 and ITU-T *Information Technology -- Open Systems Interconnection -- The Directory: Models*, ISO/IEC International Standard 9594-2, ITU-T Recommendation X.501

[X509] ISO/IEC JTC 1 and ITU-T *Information Technology -- Open Systems Interconnection -- The Directory: Authentication Framework*, ISO/IEC International Standard 9594-8, ITU-T Recommendation X.509

[X521] ISO/IEC JTC 1 and ITU-T *Information Technology -- Open Systems Interconnection -- The Directory: Selected Object Classes*, ISO/IEC International Standard 9594-7, ITU-T Recommendation X.521