Category: technical information
Status: DRAFT
Document: eugridpma-certprofile-20060814-0-8.doc
Editor: David Groep
Last updated: Wed, 16 August 2006
Total number of pages: 16

Grid Certificate Profile

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 IS IN DRAFT – COMMENTS ARE VERY WELCOME

Table of Contents

1 Scope of this document

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 describes the possibilities and limitations for attributes and extensions in X.509 certificates that are usable by the majority of the grid infrastructures today. These possibilities and limitations must be interpreted in the context of RFC 3280, 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.

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

Issuer and end-entity certificates of all IGTF accredited authorities that issue X.509 certificates must comply with the restrictions mentioned in this document.

2 Self-signed and subordinate CA certificates

2.1 General provisions

All CA certificates MUST be in X.509 version 3, i.e. the version number MUST be set to “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. If a root certificate is re-issued with the same serial number – i.e. in case only the lifetime is extended but the key pair remains the same – Mozilla NSS-based browsers will issue a user warning. In this case, if the new certificate is downloaded with IE, it will overwrite the old one; for NSS-based browsers, the old certificate must be removed from the certificate store first. If the serial number is changed, the import of the new root certificate in Internet Explorer will result in both certificates being retained in the certificate store, and the original one is not overwritten.

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 advised value.

2.3 Issuer and Subject names

Not all attribute types are equally suited to being part of the Issuer or Subject Distinguished Name. Only the following attribute types SHOULD be used, as they are considered "safe": DC, C, O, OU, ST, CN, and L. If you venture outside of this space, odd results may happen in specific installations or with specific client libraries.
To ensure uniqueness and reproducibility of the string renderings of these DNs, which are typically used in subsequent authorization steps, the ASN.1 *SEQUENCE* MUST contain *SETs* of length 1 only. Other *SET* lengths MUST NOT be used.

Contrary to the guidance derived from X.521, multiple instances of the “Organization” attribute MAY be used, as it it has been confirmed that known Grid software today correctly handles this case, and will collate the attributes in the proper order. Also, multiple instances of the “commonName” MAY be used.

The rendering of a multi-“O”, or multi-“CN” name 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, as all grid middleware, as well as the latest versions of FreeRadius, use the entire DN for subject identification.

2.3.1 serialNumber

The AttributeType "serialNumber" {id-at 5, i.e. 2.5.4.5} MUST NOT be used in any Name.

It 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.

There is a second reason not to touch serialNumber: there are versions of OpenSSL out there (up to and including versions 0.9.6) that have a non-standard string representation "SN" of this attribute type. This string representation squarely collides with the recognised abbreviated representation of "surname". It has been changed in OpenSSL 0.9.7+, so

depending on the OpenSSL version used the string representations of DNs with the "serialNumber" RDN component will differ, and this leads to problems in authorization.

2.3.2 emailAddress

The attribute type "emailAddress" SHOULD NOT be used in Names.

It has been obsoleted in the recent RFCs (in favour of having an rfc822EmailAddress in the subjectAlternativeName), and many recent mail clients can deal with subjectAltName. The issues with this attribute type are caused by OpenSSL (again), where versions up to and including 0.9.6 used the non-standard string representation "Email" for this attribute type.
In all cases, the CA itself is not to send email, so mailer client support need not be an issue.

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. Also, it is not relevant for CA certificates of any kind.

The string representation of this attribute is entirely fuzzy. OpenSSL versions 0.9.6 and lower have no string representation for this, and then some versions of the Globus Toolkit that use this OpenSSL version forcibly re-code this to "USERID". Recent OpenSSL versions stringify it to the standard representation "UID", so again there is a clash in the representation. Both "uid" and "userid" are valid string representation of OID 0.9.2342.19200300.100.1.1, with "userid" defined in RFC1274 and “uid” in 2253.

2.3.4 DomainComponent, country, organization, organizationalUnit, etc.

The distinguished name is usually made up of a combination of the attribute types “DC”, “C”, “O”, “OU”, “ST”, and “L”.

To ensure uniqueness and proper delegation, the use of *domainComponent* corresponding to a – duly registered – DNS name of the authority at the start of the issuer and subject distinguished name is strongly encouraged. In that case, the ASN.1 *SEQUENCE* MUST start with the *domainComponent* representing the top-level domain (e.g. “org”, or “eu”).

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.

2.4 Extensions in CA certificates

For proper operation as a CA certificate, only “basicConstraints” and “keyUsage” need to be present in the (root or subordinate) certificate. There is no a priori requirement by (grid) software to add any other extension to 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 MUST be marked as critical. For a CA certificate, *keyCertSign* and *cRLSign* MUST be set.

Setting only these two attributes is highly preferred. For proper operation, it is not required to have more than these two in the CA certificate, and adding additional attributes conveys the wrong message to relying parties. For a detailed description of the possible values, see Chapter 5.

2.4.3 extendedKeyUsage

The extendedKeyUsage SHOULD NOT be included in a CA certificate, as there is no use for the values of this attribute. It MUST NOT be marked critical.

2.4.4 nsCertType, nsComment , nsPolicyURL, nsRevocationURL

All these attributes are deprecated and SHOULD NOT be included in any new CA certificates. If they are included, though, these extensions MUST NOT be marked as critical.

If you really want to add some explicit text to the certificate, the only place to do that apart from *nsComment* is actually in the *certificatePolicies.userNotice.explicitText* (which must be encoded as an IA5String), but then you are sure to break software that only expects OIDs there. So this form of cetificatePolicies SHOULD NOT be used.

2.4.5 cRLDistributionPoints

This extensions need not be in the CA certificate (but must be in the end-entity certificates). Clients could use this to retrieve the CRL on-demand – but no (grid) software today actually supports that. Putting the CRL distribution URL in the CA certificates implies that it will never change during the lifetime of the CA certificate, so if you do include it here, make sure the URL will be stable over the next 5–20 years.

3 End-entity certificates

3.1 General provisions

All end-entity certificates MUST be in X.509 version 3, i.e. the version number MUST be set to "2", as the use of specific extensions (such as basicConstraints and keyUsage) is required. The serial number of each certificate MUST be unique.

For the message digest that protects the certificate integrity, known-weak signatures or hash functions (such as MD5) MUST NOT be used in end-entity 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 advised value.

3.2 Subject names

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

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

To ensure uniqueness and proper delegation, the use of domainComponent corresponding to a – duly registered – DNS name of the authority at the start of the issuer and subject distinguished name is strongly encouraged. In that case, the ASN.1 SEQUENCE MUST start with the domainComponent representing the top-level domain (e.g. "org", or "eu").

3.2.1 commonName

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

Preferably, this RDN component (but also all others), SHOULD be encoded as PrintableStrings, but certainly not contain characters that cannot be expressed in 7-bit ASCII, as these characters have inconsistent representations in different pieces of software, and cannot easily be passed around between locales, or be read from log files.

PrintableString encoding

Note that 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. RFC1778 has almost the same definition of PrintableString, differing only in allowing ''' (single quote), instead of '"' (double quote).

Of these, comma SHOULD NOT be used (since in X.500 naming, the RDNs are comma separated). Double quote MUST NOT be used and single quote SHOULD NOT be used, because OpenSSL follows RFC1778's definition of PrintableString.

Case: While printableString encodings are supposed to be case insensitive (see e.g., RFC3280), in practice most Grid middleware uses case sensitive comparison. A related problem is found with consecutive spaces which are supposed to be collapsed to a single space (ibidem). The CA MUST ensure that case and consecutive spaces is not used to distinguish between users (e.g. users with the same name).

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

For personal certificates, the CN SHOULD contain a reasonable representation of the person's name, possibly with characters added to ensure uniqueness or some distinguishing characters to allow a person to have more than one DN assigned[1].

For host certificates, typically the (primary) FQDN of the server is included here. For "normal" certificates, there must not be any additional characters in the CN. Some selected components of some grid middleware recognize a Kerberos-style "service" name in the CN as well, which looks 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.

Note that for name-based virtual hosting, additional FQDNs can be listed in the subjectAltName extension as multiple dNSNames; many modern browsers, such as IE6 or Firefox 1.5+, will recognize these names in the subjectAltName and not put up a warning box to the user in that case.

Note that (old) versions of FreeRadius, but possible other software as well, uses only the commonName for authorization. No grid middleware is known to be thus limited. Many browsers use the commonName to label certificates in their certificate stores.

3.2.1 serialNumber

The AttributeType "serialNumber" {id-at 5, i.e. 2.5.4.5} MUST NOT be used in any Name.

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.

There is a second reason not to touch serialNumber: there are versions of OpenSSL out there (up to and including versions 0.9.6) that have a non-standard string representation "SN" of this attribute type. This string representation collides with the well-recognized abbreviated representation of "surname". It has been changed in OpenSSL 0.9.7+, so depending on the OpenSSL version used the string representations of DNs with the "serialNumber" RDN component will differ, and this leads to problems in authorization.

Specifically, the serialNumber attribute MUST NOT be used to re-encode the certificate serial number in the subject name: it is not only redundant information, but it also makes renewals impossible.

3.2.2 emailAddress

The attribute type pkcs9email (emailAddress) SHOULD NOT be used in names.

It is declared obsolete in recent RFCs (in favour of haviong an rfc822EmailAddress in the subjectAlternativeName), and many all recent mail clients are able to deal with subjectAltName (Lotus Notes and Communigate are known exceptions). The issues with this attribute type are caused by OpenSSL (again), which in versions up to and including 0.9.6 used the non-standard string representation "Email" for this attribute type.
In particular, if used, by RFC3280 email addresses MUST be encoded in RFC822 "addr-spec" format (section 6.1) and they MUST be encoded as IA5String.

[1] Having for 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.

3.2.3 userID or uid

The attribute type “userID” or “uid” {0.9.2342.19200300.100.1.1} MUST NOT be used in Names.

The string representation of this attribute is entirely fuzzy; OpenSSL versions 0.9.6 and lower have no string representation for this attribute, and thus some versions of the Globus Toolkit that depend on such OpenSSL versions forcibly re-code this to "USERID". Recent OpenSSL versions stringify it to the RFC2253 standard representation "UID", so there is a clash in the representation between these softwares that results in mismatches in subsequent use during the authorization phase. Since both "uid" and "userid" are valid string representation of OID 0.9.2342.19200300.100.1.1, with "userid" defined in RFC1274 and “uid” in 2253, it is unlikely that this confusion will ever be resolved.

3.2.4 C, O, OU, L, ST

The encoding rules for commonName (section 3.2.1) also apply to these.

Moreover, when used, the C MUST encode the country covered by the CA (as opposed to, say, the country where the user is located). The value of the C attribute SHOULD contain the two-letter ISO3166 encoding of the country’s name[2]. The C, if used, MUST be used at most once.

3.3 Extensions in end-entity certificates

For proper operation as an end-entity certificate, only “basicConstraints”, “keyUsage”, “certificatePolicies”, “cRLDistributionPoints”, and *either* “extendedKeyUsage” *or* “nsCertType” need to be present in the certificate – where the use of nsCertType is depricated. For end-entity certificates issued to SSL Servers, the “subjectAltName” extensions MUST also be present.

There is no a priori requirement by (grid) software to add any other extension to the certificate.

3.3.1 basicConstraints

The basicConstraints extension SHOULD be included in end-entity certificates. According to the ASN.1 encoding rules, a value “CA:FALSE” 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.

If your CA software is capable of generating this extension 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”. It is not known if there is client software that will incorrectly allow signing of subordinate certificates if this extension is absent.

This extension MUST be marked as critical.

3.3.2 keyUsage

The keyUsage extension MUST be included in end-entity certificates, and it MUST be marked as 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!

[2] 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 is UA.

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 MUST NOT be set for server certificates (including "host" and "service" certificates), as it would imply that any use of the key would constitute incontrovertible evidence that the signing was done in a conscious way – something that can never be true for a server certificate. It's use in personal end-entity certificates SHOULD be limited to special-purposes.

The *dataEncipherment* value MAY be set, but is also intended for special purposes.

The *keyCertSign* and *cRLSign* MUST NOT be set in an end-entity certificate.

3.3.3 extendedKeyUsage

The extendedKeyUsage SHOULD be included in end-entity certificates, but it MUST NOT be marked critical. Refer to Chapter 5 for the relevant values to include in end-entity certificates.

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

3.3.3.1 Application interplay between extendedKeyUsage and nsCertType

The *extendedKeyUsage* and *nsCertType* extensions have a particular interplay, as they partially cover the same issues. In OpenSSL derived software, the *nsCertType* will be used to determine the SSL Server or Client purpose of the certificate in the absence of extended KeyUsage.

An OpenLDAP client needs at least one of the two to be present in the OpenLDAP server certificate to properly establish a SSL/TLS connection:

- *nsCertType: server* or
- *extendedKeyUsage: serverAuth*

If both are defined but the purpose in not consistent, the effect is unknown. If neither is defined authentication will fail. Note that OpenLDAP is a necessary component of the Unicore grid middleware.

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 is, however, unclear.

3.3.4 nsCertType

This attribute is deprecated and is not needed in new certificates, if the proper equivalent bits in *extendedKeyUsage* are asserted.

If this extension is included together with extendedKeyUsage, the certificate purpose expressed in both extensions MUST be equivalent for those bits in extendedKeyUsage that express similar purposes.

If *nsCertType* is included, though, the extension MUST NOT be marked as critical.

3.3.5 nsPolicyURL, nsRevocationURL

These attributes are deprecated and are not needed in end-entity certificates. If it is included, though, this extension MUST NOT be marked as critical.

3.3.6 nsComment

This attribute is deprecates and is not needed in end-entity certificates.

If you really want to add some explicit text to the certificate, the only place to do that apart from *nsComment* is actually in the *certificatePolicies.userNotice.explicitText* (which must be encoded as an IA5String), but then you are sure to break software that only expects OIDs there. This form of cetificatePolicies SHOULD NOT be used.

If it is included, though, this extension MUST NOT be marked as critical.

3.3.7 cRLDistributionPoints

This extensions MUST be present in end-entity certificate, and MUST contain at least one http URL (although it may contain other URIs). Clients could use this to retrieve the CRL on-demand – but no (grid) software today actually supports that.

Note that OpenSSL is not able to display the values of the "reasons" and the "CRLissuer" associated with a *DirName* or *URI*.

Some grid software[3] is known not to be able to handle any attributes other than a single URI in this extension.

3.3.8 authorityKeyIdentifier

The *authorityKeyIdentifier* (AKI) is not usually interpreted by the software. It is not known to cause issues with grid software, as it is ignored. The extension MUST NOT be marked critical.

If the AKI contains information that changes when the CA certificate is modified, it will block any "smooth" replacement of CA certificates (i.e. updating a CA certificate to modify the expiry date). Possible attributes in AKI include the directoryName of the authority that issued the issuer certificate (safe as it should not change) plus the serial number (which may or may not change), and/or the keyId 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.

3.3.9 subjectKeyIdentifier

The extension MUST NOT be marked critical.

3.3.10 certificatePolicies

The certificatePolicies extension MUST be present and MUST contain at least one OID. It MAY have more than one OID, i.e. to refer to an Authentication Profile, or one or more one-statement certificate policies (1SCPs). Inclusion of other elements, such as CPS pointers and explicitText is possible, but untested, and as such it is not advisable to include these additional values.

The extension MUST NOT be marked critical.

3.3.11 subjectAlternativeName, issuerAlternativeName

The subjectAlternativeName extension MUST be present for server certificates (and "host" and "service" certificates in the grid context), and MUST contain at least one FQDN (*dNSName*). If the end-entity certificate needs to contain an rfc822 email address, this extension is also the proper place to put this (as an *email* attribute).

For use with web browsers, you can add multiple FQDNs in *dNSName* attributes, allowing name-based virtual hosting for secure web sites – at least for up-to-date browsers such as IE5+ and Firefox 1.5+.

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

The extension MUST NOT be marked critical.

3.3.12 authorityInformationAccess

This is the proper extension to point to any production-quality OCSP service. There is no grid software that uses this extension today, but it also does not do any harm. The extension MUST NOT be filled with values that point to experimental or non-monitored services, as this will break the system as soon as the (OCSP) service is actually implemented in the software.

It MAY also contain the CRL URI, or the location of any superior CA certificates, but note that a CRL URI MUST also be included in the *cRLDistributionPoints* extension.

The extension MUST NOT be marked as critical.

4 General Considerations

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

When you look back at the X.500-series specs and even in the X.509 style guide, there was no explicit guidance on the ordering of the components. It says

```
Name ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET OF AttributeValueAssertion
      AttributeValueAssertion ::= SEQUENCE {
          attributeType OBJECT IDENTIFIER,
          attributeValue ANY
}
```

where 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 (the representation of that in a string is subject to discussion, as long debates between OpenSSL and RFC2253 have shown).

What SwissSign (and apparently also Purdue) have done is to start the SEQUENCE with the commonName (use the OpenSSL asn1parse command to see the exact structure). There used to be no definite guidance on this, but the new RFC 3280bis is supposed to have a statement that the SEQUENCE ought to start with Country, or a domainComponent. Before this “bis” edition (which is still in draft) it could only be deduced from the examples. The only reason most of the CAs did put the commonName at the end is because of this line: "In theory it should be a full, proper DN, which traces a path through the X.500 DIT" and most of us interpret "trace" as "start at the root of the tree". The X.509 style guide made clear that for the DN, anything goes, as it has been ill-defined up to now.

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

The ‘reverse’ ordering of the *SEQUENCE* of *RDNs* is theoretically *not* malformed, but causes significant problems with most software. Some previously established CAs 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. Note that SwissSign has, on request of SWITCH, gone through a change process to allow the SWITCH CA to issue end-entity certificates in the "other" ordering for Grid end-entity certificates.

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.

4.2 Keys, key lengths and hashes

According to NIST 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. RSA claims that 1024-bit keys are likely to become crackable between 2006 and 2010 and that 2048-bit keys are sufficient until 2030. 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.[4]

Similar considerations hold for the hash functions used. The MD5 hash function is known to have collisions, and a proof-of-principle collision between two certificates has been published. Also SHA-1 has recently been shown to provide less than 80 bits of security, but as more modern hash functions (such as SHA-256) are not yet widely supported, there is no ready alternative.
See also www.keylength.com for an overview and comparison based on various publications.

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 *cannot handle* 4096 bit keys. Although a workaround is available[5], use of 4096-bit keys might still be less advisable in 2006, but this should be re-evaluated in 2007.

[4] Source: Recommendation for Key Management, NIST Special Publication 800-57 Draft, 08/2005.
[5] http://codelabs.ru/grid/java-4096.txt

5 Extension attribute values and types

The meaning of even the most common extensions is not always immediately clear. Although some good descriptions exist[6], some of the information is repeated here. Only extensions that are a source of confusion or have special application characteristics in (grid) software are mentioned.

5.1 keyUsage

The following table has the list of possible values:

Attribute	Comments
digitalSignature	Ought not to be in a CA cert as it is not supposed to casually sign documents (as opposed to certificates or CRLs, to which this extension in not applicable). Should be in user certificates if they are used to sign document and mail, or to authenticate.
nonRepudiation	Claims that signing with the key pertaining to this certificate is incontrovertible evidence that the signatory has done that consciously. Should not be in CA certs, may be in user certs for signed email.
keyEncipherment	Key is used in key management (mainly DH). Software status support unknown, including it in EE certificates does not harm operations but is not needed.
dataEncipherment	May be in EE certificates encrypting data, but should not be in CA certificates
keyAgreement	To be used in the exchange of keys. Software support status unknown, but including it in EE certificates does not harm operations.
keyCertSign	Must be present in a CA certificate, but never in an EE cert
cRLSign	Must be in a CA certificate in order to sign its CRLs
encipherOnly	Only applicable together with *keyAgreement*
decipherOnly	Only applicable together with *keyAgreement*

5.2 extendedKeyUsage

The following table has the list of possible values for OpenSSL 0.9.7d:

Attribute	Comments
serverAuth	This is an SSL Server
clientAuth	This is an SSL Client
OCSPSigning	This is a trusted OCSP responder
timeStamping	
codeSigning	
emailProtection	For use with S/MIME software
ipsecEndSystem	
ipsecTunnel	
ipsecUser	
DVCS	?

[6] 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

5.3 nsCertType

The following table has the list of possible values (names from OpenSSL):

Attribute	Comments
server	This is an SSL Server
client	This is an SSL Client
email	For use with S/MIME
objsign	
sslCA	
emailCA	
objCA	

5.4 cRLDistributionPoints

This extension should contain a list of locations where the actual CRL data is stored, for example a URI 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
```

5.5 certificatePolicies

It is possible to add some free text to this extension, as a replacement of *nsComment*. This would require a proper parsing of this extension by all client software (**which has not been checked yet**). To do that in OpenSSL, the configuration file would need:

```
certificatePolicies=ia5org,1.2.3.4,1.5.6.7.8,@polsect

[polsect]

policyIdentifier = 1.3.5.8
CPS.1="http://my.host.name/"
CPS.2="http://my.your.name/"
userNotice.1=@notice

[notice]

explicitText="Explicit Text Here"
organization="Organisation Name"
noticeNumbers=1,2,3,4
```

Also, if *organization* is included, also the *noticeNumbers* MUST be included, and vice versa. *ExplicitText* can be used ‘stand alone’.