DNS Certification Authority Authorization (CAA) Resource Record
draft-ietf-pkix-caa-01

Abstract

The Certification Authority Authorization (CAA) DNS Resource Record allows a DNS domain name holder to specify the certificate signing certificate(s) authorized to issue certificates for that domain. CAA resource records allow a public Certification Authority to implement additional controls to reduce the risk of unintended certificate mis-issue.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 8, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
1. Definitions

1.1. Requirements Language

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 [RFC2119].

1.2. Defined Terms

The following terms are used in this document:

Abstract Syntax Notation One (ASN.1) - A notation for describing abstract types and values, as specified in X.680 [X.680].

Authorization Entry - An authorization assertion that grants or denies a specific set of permissions to a specific group of entities.

Canonical Domain Name - A Domain Name that is not an alias.

Canonical Domain Name Value - The value of a Canonical Domain Name. The value resulting from applying alias transformations to a Domain Name that is not canonical.

Certificate - An X.509 Certificate, as specified in RFC 5280 [RFC5280].

Certification Policy (CP) - Specifies the criteria that a Certification Authority undertakes to meet in its issue of certificates.

Certification Practices Statement (CPS) - Specifies the means by which the criteria of the Certification Policy are met. In most cases this will be the document against which the operations of the Certification Authority are audited.

Certification Authority (CA) - An entity that issues Certificates in accordance with a specified Certification Policy.

Distinguished Encoding Rules (DER) - A set of rules for encoding ASN.1 objects, as specified in X.690 [X.690].

Domain - The set of resources associated with a DNS Domain Name.

Domain Name - A DNS Domain name as specified in RFC 1035 [RFC1035] and revisions.
Domain Name System (DNS)  The Internet naming system specified in RFC 1035 [RFC1035] and revisions.

DNS Security (DNSSEC)  Extensions to the DNS that provide authentication services as specified in RFC 4033 [RFC4033] and revisions.

Extended Issuer Authorization Set  The most specific Issuer Authorization Set that is active for a domain. This is either the Issuer Authorization Set for the domain itself, or if that is empty, the Issuer Authorization Set for the corresponding Public Delegation Point.

Issuer Authorization Set  The set of Authorization Entries for a domain name that are flagged for use by Issuers. Analogous to an Access Control List but with no ordering specified.

Public Delegation Point  A Domain Name that is obtained from a public DNS registry as defined by a Certification Policy.

Public Key Infrastructure X.509 (PKIX)  Standards and specifications issued by the IETF that apply the X.509 [X.509] certificate standards specified by the ITU to Internet applications as specified in RFC 5280 [RFC5280] and related documents.

Resource Record (RR)  A set of attributes bound to a Domain Name.

Relying Party  A party that makes use of an application whose operation depends on use of a Certificate for making a security decision.

Relying Application  An application whose operation depends on use of a Certificate for making a security decision.

Relying Party Authorization Set  The set of Authorization Entries for a domain name that are flagged for use by Relying Party Applications. Analogous to an Access Control List but with no ordering specified.

2. Introduction

The Certification Authority Authorization (CAA) DNS Resource Record allows a DNS domain name holder to specify the Certification Authorities authorized to issue certificates for that domain. Publication of CAA resource records allow a public Certification Authority (CA) to implement additional controls to reduce the risk of unintended certificate mis-issue.
Conformance with a published CAA record is a necessary but not sufficient condition for issue of a certificate. Before issuing a certificate, a PKIX CA is required to validate the request according to the policies set out in its Certificate Policy Statement. In the case of a public CA that validates certificate requests as a third party, the certificate will be typically issued under a public root certificate embedded in one or more relevant reliant applications.

Criteria for inclusion of embedded root certificates in applications are outside the scope of this document but typically require the CA to publish a Certificate Practices Statement (CPS) that specifies how the requirements of the Certificate Policy (CP) are achieved and provide an annual audit statement of their performance against their CPS performed by an independent third party auditor.

It is the intention of the authors to propose the CAA record defined in this document as the basis for CA validation requirements to be proposed in organizations that publish validation requirements.

CAA records only describe the current state of Certification Authority certificate issue authority. Since a certificate is typically valid for at least a year, it is possible that a certificate that is not conformant with the CAA records currently published was conformant with the CAA records published at the time that it was issued. Thus Relying Applications MUST NOT use failure to conform to currently published CAA records as a rejection criteria for certificates unless the published records are flagged as being intended for that use.

2.1. The CAA RR type

A CAA RR publishes a CAA property entry that corresponds to the specified domain name. Multiple property entries MAY be associated with the same domain name by publishing multiple CAA RRs at that domain name. Each property entry MAY be tagged with one or more of the following flag values:

Critical  If set, indicates that the corresponding property entry tag MUST be understood if the semantics of the CAA record are to be correctly understood by the specified audience.

Issuers MUST NOT issue certificates for a domain if the Extended Issuer Authorization Set contains unknown property entry tags that are flagged as critical.
Relying Parties MUST NOT attempt to enforce CAA records if the
Relying Party Authorization Set contains unknown property entry
tags that are flagged as critical

Must be Zero  This bit is reserved for future use.

Issuers MUST NOT issue certificates for a domain if the Extended
Issuer Authorization Set contains property entries with the Must
Be Zero Tag Set.

Relying Parties MUST NOT attempt to enforce CAA records if the
Relying Party Authorization Set contains property entries with the
Must Be Zero Tag Set.

Relying Party  This bit is reserved to specify that the corresponding
Property Entry MAY be used by Relying Party Applications.

Relying Parties MUST NOT attempt to enforce properties specified
in CAA records with the Relying Party bit set clear.

Issuer  Specifies that the corresponding Property Entry is to be used
by Issuers and forms part of the Issuer Authorization Set for the
domain.

The following properties are defined:

policy <Certificate Policy OID>  The policy property entry declares
an authorization entry granting authorization to issue under the
specified Certificate Policy.

path <Object Digest Identifier>  The path property entry declares an
authorization entry granting authorization to issue end entity
certificates under a trust path that includes the specified
signing credential.

An Object Digest Identifier (ODI) is a means of specifying a
reference to an object instance by means of a cryptographic digest
function. A CAA path property may use an ODI to specify a
certificate trust path by means of:

A Certificate Signing Certificate

A Public Signing Key

In either case a path Authorization Entry authorizes an issuer to
issue an End Entity certificate to the corresponding domain if and
only if it is possible to form a valid certificate path to it from
the referenced certificate or key.
2.1.1. Examples of Use.

For convenience the examples are presented in the text format suggested in section Section 3.1.1

The following example informs CAs that certificates must not be issued except under the Default Deny Security 'Example 1' Certificate Policy (1.3.6.1.4.1.35405.666.1). Since the policy is published at the Public Delegation Point, the policy applies to all subordinate domains under example.com.

$ORIGIN example.com
.       CAA 1 policy 1.3.6.1.4.1.35405.666.1

The following example informs CAs that certificates must not be issued except under the Certificate Authority Root certificate specified in Appendix B.

$ORIGIN example.com
.       CAA 1 path MDIGA1UEJQYJYYI2IAWUDBAIBBCAXzJgPaoT7Fe
       XaPzKv6mI200y1i1f+7WhzmhMGLe/oBA==

A domain MAY authorize multiple CAs to issue certificates at the same time. The following example allows issue under the Default Deny Security certification policy 'Example 1' or 'Example 2':

$ORIGIN example.com
.       CAA 1 policy 1.3.6.1.4.1.35405.666.1
.       CAA 1 policy 1.3.6.1.4.1.35405.666.2

If Authorization Entries using the path and policy properties are present at a given Domain, compatibility with either is sufficient to authorize the request.

Future versions of this specification MAY use the critical flag to introduce new semantics that MUST be understood for correct processing of the record, preventing Certification Authorities that do not recognize the record from issuing certificates.

In the following example, the property 'tbs' is flagged as critical. The Default Deny Security CA is not authorized to issue under either policy unless the processing rules for the 'tbs' property tag are understood.

$ORIGIN example.com
.       CAA 1 policy 1.3.6.1.4.1.35405.666.1
.       CAA 1 policy 1.3.6.1.4.1.35405.666.2
.       CAA 129 tbs MDIGA1UEJQYJYYI2IAWUDBAIBBCAXzJgPaoT7Fe
Note that the above restrictions only apply to issue of certificates. Since the validity of an end entity certificate is typically a year or more it is quite possible that the CAA records published at a domain will change between the issue of the certificate and verification by a relying party.

2.2. Certification Authority Processing

Before issue of a certificate, a compliant CA MUST check for publication of a relevant CAA Resource Record(s) and if such record(s) are published, that the certificate requested is consistent with them. If the certificate requested is not consistent with the relevant CAA RRs, the CA MUST NOT issue the certificate.

The Issuer Authorization Set for a domain name consists of the set of all CAA Authorization Entries declared for the canonical form of the specified domain.

The Extended Issuer Authorization Set for a domain name consists of the Issuer Authorization Set for that domain name if it is non-empty. Otherwise the Extended Issuer Authorization Set for a domain name consists of the Issuer Authorization Set for the corresponding Public Delegation Point for that domain name.

If the Extended Issuer Authorization Set for a domain name is not empty, a Certification Authority MUST NOT issue a certificate unless it conforms to at least one authorization entry in the Extended Issuer Authorization Set.

Note that while it MUST be possible to form a certificate validation path that contains at least one certificate that is so specified, it MAY also be possible to form valid certificate paths that are not.

For example, a CA that has updated its root certificate to extend the expiry date is entitled to issue certificates for domains where the CAA record only specifies the older root certificate provided that the older root certificate has not actually expired and it is thus possible to form a valid certificate path.

2.2.1. Canonical Domain Name

The DNS defines the CNAME and DNAME mechanisms for specifying domain name aliases. The canonical name of a DNS name is the name that results from performing all DNS alias operations.

A Certification Authority MUST perform CNAME and DNAME processing as
defined in the DNS specifications 1035 [RFC1035].

2.2.2. Use of DNS Security

Use of DNSSEC to authenticate CAA RRs is strongly recommended but not required. A CA MUST NOT issue certificates if doing so would conflict with the corresponding extended issuer authorization set whether the corresponding DNS records are signed or not.

Use of DNSSEC allows a CA to acquire and archive a non-repudiable proof that they were authorized to issue certificates for the domain.

2.2.3. Archive

A compliant CA SHOULD maintain an archive of the DNS transactions used to verify CAA eligibility.

In particular a CA SHOULD ensure that where DNSSEC data is available that the corresponding signature and NSEC/NSEC3 records are preserved so as to enable later compliance audits.

3. Mechanism

3.1. Syntax

A CAA RR contains a single property entry consisting of a tag value pair. Each tag represents a property of the CAA record. The value of a CAA property is that specified in the corresponding value field.

A domain name MAY have multiple CAA RRs associated with it and a given property MAY be specified more than once.

The CAA data field contains one property entry. A property entry consists of the following data fields:

```
+0-1-2-3-4-5-6-7-| 0-1-2-3-4-5-6-7-|
| Flags           | Tag Length = n |
+----------------+----------------+...+---------------+
| Tag char 0     | Tag Char 1     |...| Tag Char n-1  |
+----------------+----------------+...+---------------+
+----------------+----------------+.....+---------------+
| Data byte 0    | Data byte 1    |.....| Data byte m-1 |
+----------------+----------------+.....+---------------+
```

Where n is the length specified in the tag length field and m is the remaining octets in the data field (m = d - n - 2) where d is the length of the data section.
The data fields are defined as follows:

**Flags**  One octet containing the following fields:

Bit 0: Critical Flag  If the value is set (1), the critical flag is asserted and the property MUST be understood if the CAA record is to be correctly processed.

A Certification Authority MUST NOT issue certificates for any Domain that contains a CAA critical property for an unknown or unsupported property type.

Bit 5: Must Be Zero  Bit 5 is reserved and MUST be set to zero. Processors that encounter a CAA record containing a property with this bit set MUST treat the record set as if the critical property was asserted for an unknown record.

Bit 6: Relying Application Use  If set, the property entry contains an Authorization Entry that forms part of the Relying Application Authorization Set for the corresponding domain.

Bit 7: Issuer Use  If set, the property entry contains an Authorization Entry that forms part of the Issuer Application Authorization Set for the corresponding domain.

Note that according to the conventions set out in RFC 1035 [RFC1035] Bit 0 is the Most Significant Bit and Bit 7 is the Least Significant. Thus a flags value of 0x51 indicates a tag length of 5 octets and that the property entry is not critical and is not to be used for relying party processing.

**Tag Length**  A single octet containing an unsigned integer specifying the tag length in octets. The tag length MUST be at least 1 and SHOULD be no more than 15.

**Tag**  The property identifier, a sequence of ASCII characters.

Tag values MAY contain ASCII characters a through z and the numbers 0 through 9. Tag values MUST NOT contain any other characters. Matching of tag values is case insensitive.

**Value**  A sequence of octets representing the property value. Property values are encoded as binary values and MAY employ sub-formats.
The length of the value field is specified implicitly as the remaining length of the enclosing Resource Record data field.

3.1.1. Canonical Presentation Format

The canonical presentation format of the CAA record is as follows:

\[
\text{CAA <flags> <tag> <data>}
\]

Where:

- **flags** is an unsigned integer between 0 and 15.
- **tag** is a non-zero sequence of ASCII letter and numbers in lower case.
- **data** is the Base64 Encoding [RFC4648] of the value field.

3.1.1.1. Policy OID Encoding Options

For convenience of administration, implementations MAY support ASN.1 Policy OID encoding at their option.

The Base64 encoding of data never contains the period character '.', while the encoding of ASN.1 OID values specified in IETF GSER encoding [RFC3642] will always incorporate at least one period character.

It follows that a data decoder MAY unambiguously interpret data specified in the Base64 or GSER format without the need for additional disambiguation.

Implementations MAY choose to allow use of both formats in both file and presentation formats.

3.1.2. policy Property value

The policy property value specifies an Authorization Entry by means of an ASN.1 OID specifying a Certification Policy. A Certification Authority is authorized to issue Certificates under a policy Authorization Entry if and only if

- The Certification Authority has the right to issue certificates under the specified policy, AND
- The certificate request is compliant with the requirements of the specified policy, AND
The certificate request meets all the criteria under the Certification Policy under which the certificate is to be issued.

Each policy property specifies a single ASN.1 OID value consisting of the ASN.1 type, length specifier and OID data.

The policy property applies to the specified policy OID and all policy OIDs that fall within the same OID arc. If the OID arc 1.3.6.1.4.1.35405.666 is specified, then the policy OIDs 1.3.6.1.4.1.35405.666, 1.3.6.1.4.1.35405.666.1, 1.3.6.1.4.1.35405.666.2 etc. are all authorized.

The Certificate that is issued MAY incorporate the specified policy OID itself but is not required to provided that the issue of the certificate is consistent with the requirements of the specified policy.

For example, a CA that offers two levels of Certification Policy such that the higher level of assurance included all the requirements of the lower one MAY rely on a policy property specifying the lower assurance policy as authorization for issue under the higher assurance policy but not vice-versa.

3.1.3. path Property value

The path property value specifies an Authorization Entry by means of a Certificate Signer Certificate or a Certificate Signing key. A Certification Authority is authorized to issue Certificates under a path Authorization Entry if and only if

A valid PKIX trust path can be formed from the specified Certificate Signer Certificate or a Certificate Signing key to the certificate that is to be issued, AND

The certificate request meets all the criteria under the Certification Policy under which the certificate is to be issued.

4. Security Considerations

CAA Records provide an accountability control. They are intended to deter rather than prevent undesired behavior.

While a Certification Authority can choose to ignore published CAA records, doing so increases the both the probability that they will mis-issue a certificate and the consequences of doing so. Once it is known that a CA observes CAA records, malicious registration requests will target disproportionately target the negligent CAs that do not,
and so the mis-issue rate amongst the negligent CAs will increase. Since the CA could clearly have avoided the mis-issue by performing CAA processing, the likelihood of sanctions against the negligent CA is increased. Failure to observe CAA issue restrictions provides an objective criteria for excluding issuers from embedded roots of trust.

In contrast, a Certification Authority that processes CAA records correctly can reasonably claim that any residual mis-issue event could have been avoided had the Domain Name holder published appropriate CAA records.

4.1. Mis-Issue by Authorized Certification Authority

Use of CAA records does not provide protection against mis-issue by an authorized Certification Authority.

Domain name holders SHOULD ensure that the CAs they authorize to issue certificates for their domains employ appropriate controls to ensure that certificates are only issued to authorized parties within their organization.

Such controls are most appropriately determined by the domain name holder and the authorized CA(s) directly and are thus out of scope of this document.

4.2. Suppression or spoofing of CAA records

Suppression of the CAA record or insertion of a bogus CAA record could enable an attacker to obtain a certificate from a CA that was not authorized to issue for that domain name.

4.2.1. Applications

Applications performing CAA checking SHOULD mitigate the risk of suppression or spoofing of CAA records by means of DNSSEC validation where present. In cases where DNSSEC validation is not available, CAA checking is of limited security value.

4.2.2. Certification Authorities

Since a certificate issued by a CA can be valid for several years, the consequences of a spoofing or suppression attack are much greater for Certification Authorities and so additional countermeasures are justified.

A CA MUST mitigate this risk by employing DNSSEC verification whenever possible and rejecting certificate requests in any case.
where it is not possible to verify the non-existence or contents of a relevant CAA record.

In cases where DNSSEC is not deployed in a corresponding domain, a CA SHOULD attempt to mitigate this risk by employing appropriate DNS security controls. For example all portions of the DNS lookup process SHOULD be performed against the authoritative name server. Cached data MUST NOT be relied on but MAY be used to support additional anti-spoofing or anti-suppression controls.

4.3. Denial of Service

Introduction of a malformed or malicious CAA RR could in theory enable a Denial of Service attack.

This specific threat is not considered to add significantly to the risk of running an insecure DNS service.

4.4. Abuse of the Critical Flag

A Certification Authority could make use of the critical flag to trick customers into publishing records which prevent competing Certification Authorities from issuing certificates even though the customer intends to authorize multiple providers.

In practice, such an attack would be of minimal effect since any competent competitor that found itself unable to issue certificates due to lack of support for a property marked critical is going to investigate the cause and report the reason to the customer who was deceived. It is thus unlikely that the attack would succeed and the attempt might lay the perpetrator open to civil or criminal sanctions.

5. IANA Considerations

5.1. Registration of the CAA Resource Record Type

IANA has assigned Resource Record Type TBD1 for the CAA Resource Record Type and added the line depicted below to the registry named Resource Record (RR) TYPES and QTYPEs as defined in BCP 42 RFC 5395 [RFC5395] and located at http://www.iana.org/assignments/dns-parameters.

<table>
<thead>
<tr>
<th>Value and meaning</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAA</td>
<td>TBD1 Certification Authority Restriction</td>
</tr>
</tbody>
</table>
5.2. Certification Authority Authorization Properties

IANA has created the Certification Authority Authorization Properties registry with the following initial values:

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization Entry by Signature Path</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>Authorization Entry by Certificate Policy</td>
<td>[RFCXXXX]</td>
</tr>
</tbody>
</table>

Addition of tag identifiers requires a public specification and expert review as set out in RFC5395 [RFC5395]

6. References

6.1. Normative References


6.2. Non Normative References


Appendix A. Object Digest Identifier Calculation

An Object Digest is an ASN.1 structure with three components:

An ASN.1 Object Identifier specifying the object type of the referenced object

An ASN.1 Object Identifier specifying the digest algorithm

An ASN.1 DER [X.690] encoded data field containing the digest value of the referenced object processed using the specified digest algorithm.
DNSCAA DEFINITIONS ::= 

BEGIN

ObjectDigestIdentifier ::= SEQUENCE {
  type             OBJECT IDENTIFIER,
  digestAlgorithm  OBJECT IDENTIFIER,
  digest           OCTET STRING
}

END

The Object Digest Identifier construction is designed to facilitate implementation in applications that already require ASN.1 handling mechanisms (i.e. most cryptographic applications) without causing an undue coding burden in cases where ASN.1 code is not already supported. Appendix C provides all the necessary information to create a fully compliant Object Digest Identifier implementation.

A.1. Example: CA Certificate A

The ODI of CA Certificate A (specified in Appendix B.1) is calculated as follows:

ASN.1 Sequence tag: "3032"

ASN.1 OID id-at-cACertificate (2.5.4.37): "0603550425"

ASN.1 OID sha256 (2.16.840.1.101.3.4.2.1): "0609608648016503040201"

SHA-256 Digest Value: "042017cc980f6a84fb15e5da3f32afea62360f4ca29627feed6739a13062defe804"

The ODI in BASE64 format is MDIGA1UEJQYJYIZIAWUDBAIBBCAXzJgPaoT7FeXaPzKv6mI2D0y1llf+7WhzmmGLe/oBA==.

A.2. Example: CA Certificate A

The ODI of the signing key of CA Certificate A (specified in Appendix B.1) is calculated as follows:

ASN.1 Sequence tag

ASN.1 OID ‘CA Signing Key’

ASN.1 OID ‘SHA-256’
Appendix B. Example Certificates

The following certificates are used in the examples.

B.1. CA Certificate A

CA Certificate A is a self signed certificate signed with a 2048 bit RSA key:

-----BEGIN CERTIFICATE-----
MIIDATCCAeugAwIBAgIBATALBgkqhkiG9w0BAQUwKDERMA8GA1UEChMIQWNtZSBJ
bmMxExARBgNVBAMTCmFtZTHwdGFsQ2FsdGluZwMBAAEwggEiAgEAMB0DAQ7BggrBgEE
AYIKoBUABIIBA0GAIwAAYDVR0gBAEEAwIBAgIBADkCAgEDBwIFADBgVzAsMDk6MRKw
EgYIKwYBBQUHAgEBBCwwCgYGCCsGAQUFBzAChGQwDQYJKoZIhvcNAQEBBQADgY0AM
BBBEwD3ANBgsrBgEAYKUTYUaATALBkgkhkiG9w0BAQQDgEBAGcNicQxdi1I9Y5e
Ps+XEYdIKyVWymSnRIfbUzQWaQpPcj5cHze91CUzipGDNJYXwHiUtQAAGmtrq
ZG4a4F9y0cPFABHXPKKGEmIHMrART7MV9kHu4DFIhb82200n4DrBiit8PNa5t/5
Cm6rcDrpWB5hjAsD37U+GZGvTJmag059VWjnjq90NcfCQ6YJ6AA5VKnmrV695VN
lDSPAaN9V55N6heqJU9tcbqPKAE3MuJtd1QxBQ34f9e1kTXYxc/dbJIK1RQ0F4nc
Jc4NbjZakvFq+QcbzEkhDM1xvJdV0JJt+GkFZrsRE16IgQY4DQHPv650Ivbr3U
329dd+g=
-----END CERTIFICATE-----

In binary form, the certificate data is:

```
0000 30 82 03 01 30 82 01 eb a0 03 02 01 02 02 01 01
0010 30 0b 06 09 2a 86 48 86 5a f7 0d 01 01 05 30 28 31
0020 13 08 41 63 65 20 49
0030 13 08 41 63 65 20 49
0040 13 08 41 63 65 20 49
0050 13 08 41 63 65 20 49
0060 13 08 41 63 65 20 49
0070 13 08 41 63 65 20 49
0080 13 08 41 63 65 20 49
0090 13 08 41 63 65 20 49
00a0 13 08 41 63 65 20 49
00b0 13 08 41 63 65 20 49
00c0 13 08 41 63 65 20 49
00d0 13 08 41 63 65 20 49
```

The SHA-256 digest of the certificate data is:

17cc980f6a84fb15e5da3f32afea62360f4ca29627feed68739a13062defe804

Appendix C.  ASN.1 Values (Non-Normative)

Although the Object Digest Identifier form employs ASN.1 DER encoding only a small subset of ASN.1 features are used and a full ASN.1 stack is not necessary.

This appendix provides sufficient information to implement an Object
Digest Identifier constructor or parser.

C.1. DER Sequence Encoding

In DER encoding, the enclosing SEQUENCE will always be represented by the type identifier x30 followed by the length specifier. Since the total length of the following data fields will almost certainly be less than 127 bytes, the single byte encoding mechanism in which bit 7 is clear and the length value is encoded in the lower 7 bits will be required.

C.2. Object Identifiers for Certificate Types

OIDs have been defined in connection with the X.500 directory for user certificates, certification authority certificates, revocations of certification authority, and revocations of user certificates. The following table lists the OIDs, their DER encoding, and their type identifier and length-prefixed hex format for use in Object Digest Identifiers.

<table>
<thead>
<tr>
<th>OBJECT IDENTIFIER</th>
<th>DER Encoding</th>
<th>Hex Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>id-at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id-at-userCertificate</td>
<td>-- 06 03 55 04 24</td>
<td>-- 06 03 55 04 24</td>
</tr>
<tr>
<td>id-at-cACertificate</td>
<td>-- 06 03 55 04 25</td>
<td>-- 06 03 55 04 25</td>
</tr>
<tr>
<td>TBS-PUBLIC-KEY-VALUE</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

C.3. Object Identifiers for Digest Algorithms

OIDs have been assigned by NIST for the SHA-2 digest algorithms [NIST-ALGS] [RFC4055] Use of the SHA-1 digest algorithm is not recommended due to concerns for the security of the algorithm.

<table>
<thead>
<tr>
<th>OBJECT IDENTIFIER</th>
<th>DER Encoding</th>
<th>Hex Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>hashAlgs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id-sha256</td>
<td>-- 06 09 60 86 48 01 65 03 04 02 01</td>
<td>-- 06 09 60 86 48 01 65 03 04 02 01</td>
</tr>
<tr>
<td>id-sha384</td>
<td>-- 06 09 60 86 48 01 65 03 04 02 02</td>
<td>-- 06 09 60 86 48 01 65 03 04 02 02</td>
</tr>
<tr>
<td>id-sha512</td>
<td>-- 06 09 60 86 48 01 65 03 04 02 03</td>
<td>-- 06 09 60 86 48 01 65 03 04 02 03</td>
</tr>
<tr>
<td>id-sha224</td>
<td>-- 06 09 60 86 48 01 65 03 04 02 04</td>
<td>-- 06 09 60 86 48 01 65 03 04 02 04</td>
</tr>
</tbody>
</table>
C.4. DER Data Encoding Prefixes

The rules of ASN.1 encoding state that every data value is preceded by a data type identifier and a length identifier. In the case of an Object Digest Identifier the data type identifier is always OCTET STRING (04) and the length for all currently defined digest algorithms will be less than 128 bytes (1024 bits) and thus use the single byte encoding form in which bit 7 is set to 0 and the lower 7 bits specify the length.

The length prefixes for commonly used digest lengths in hexadecimal notation are thus:

- 160 bits  04 14
- 224 bits  04 1C
- 256 bits  04 20
- 384 bits  04 30
- 512 bits  04 40

Authors’ Addresses

Phillip Hallam-Baker  
Comodo Group Inc.  
Email: philliph@comodo.com

Rob Stradling  
Comodo CA Ltd.  
Email: rob.stradling@comodo.com

Ben Laurie  
Google Inc.  
Email: benl@google.com