

Internet Engineering Task Force	P. M. Hallam-Baker
Internet-Draft	Comodo Group Inc.
Intended status: Standards Track	R. N. Stradling
Expires: December 04, 2011	Comodo CA Ltd.
	B. Laurie
	Google Inc.
	June 02, 2011

DNS Certification Authority Authorization (CAA) Resource Record
draft-ietf-pkix-caa-00

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 December 04, 2011.

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

[Table of Contents](#)

- *1. [Definitions](#)
 - *1.1. [Requirements Language](#)
 - *1.2. [Defined Terms](#)
- *2. [Introduction](#)
 - *2.1. [The CAA RR type](#)
 - *2.1.1. [Examples of Use.](#)
 - *2.2. [Certification Authority Processing](#)
 - *2.2.1. [Canonical Domain Name](#)
 - *2.2.2. [Use of DNS Security](#)
 - *2.2.3. [Archive](#)
 - *2.3. [Relying Party Application Processing](#)
- *3. [Mechanism](#)
 - *3.1. [Syntax](#)
 - *3.1.1. [Canonical Presentation Format](#)
 - *3.1.1.1. [Policy OID Encoding Options](#)
 - *3.1.2. [policy Property value](#)
 - *3.1.3. [path Property value](#)
- *4. [Security Considerations](#)
 - *4.1. [Mis-Issue by Authorized Certification Authority](#)
 - *4.2. [Suppression or spoofing of CAA records](#)
 - *4.2.1. [Applications](#)
 - *4.2.2. [Certification Authorities](#)
 - *4.3. [Denial of Service](#)
 - *4.4. [Abuse of the Critical Flag](#)
- *5. [IANA Considerations](#)

- *5.1. [Registration of the CAA Resource Record Type](#)
- *5.2. [Certification Authority Authorization Properties](#)
- *6. [References](#)
 - *6.1. [Normative References](#)
 - *6.2. [Non Normative References](#)
- *Appendix A. [Object Digest Identifier Calculation](#)
 - *Appendix A.1. [Example: CA Certificate A](#)
 - *Appendix A.2. [Example: CA Certificate A](#)
- *Appendix B. [Example Certificates](#)
 - *Appendix B.1. [CA Certificate A](#)
- *Appendix C. [ASN.1 Values \(Non-Normative\)](#)
 - *Appendix C.1. [DER Sequence Encoding](#)
 - *Appendix C.2. [Object Identifiers for Certificate Types](#)
 - *Appendix C.3. [Object Identifiers for Digest Algorithms](#)
 - *Appendix C.4. [DER Data Encoding Prefixes](#)
- *[Authors' Addresses](#)

[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 Specifies that the corresponding Property Entry is to be used by Relying Party Applications and forms part of the Relying Party Authorization Set for the domain.

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 MDIGA1UEJQYJYIZIAWUDBAIBBCAXzJgPaoT7Fe
      XaPzKv6mI2D0yilif+7WhzmhMGLLe/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 MDIGA1UEJQYJYIZIAWUDBAIBBCAXzJgPaoT7Fe
          XaPzKv6mI2D0yilif+7WhzmhMGLe/oBA==
```

Enforcement by Relying Party Applications follows the same general principles. A Relying Party Application MUST NOT enforce CAA records unless at least one Property Entry has the Relying Party flag set, that is the Relyin Party Authorization Set is not empty. In the following example, certificates must not be issued except under the Default Deny Security 'Example 1' Certificate Policy and Relying Party Applications MAY reject certificates presented that do not comply with this requirement:

```
$ORIGIN example.com
.      CAA 3 policy 1.3.6.1.4.1.35405.666.1
```

In the ordinary course of business a Domain administrator may withdraw authorization for issue of new certificates before the previously issued certificates expire.

In the following example, Relying Party Applications are informed that certificates issued under either the policy are to be considered to be authorized but new certificates can only be issued under the first.

```
$ORIGIN example.com
.      CAA 3 policy 1.3.6.1.4.1.35405.666.1
.      CAA 2 policy 1.3.6.1.4.1.35405.666.2
```

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.

2.3. Relying Party Application Processing

Relying Party Applications **MAY** enforce CAA issue restrictions at their option, provided that the Relying Party Authorization set is not empty. The consequences of determining that a certificate is not compatible with the specified CAA relying party restrictions are outside the scope of this document.

Domains that opt to flag records for use by Relying Party Applications **SHOULD** be aware that the Property Entries supported in this version of the specification are only designed to support the requirements of enforcing issuer restrictions. While these Property Entries **MAY** be sufficient to enable enforcement by Relying Party Applications in some circumstances, they are not intended to provide complete requirements coverage for this purpose.

Domains containing CAA issue restrictions intended for use by Relying Party Applications SHOULD be authenticated using DNSSEC or other equivalent means.

If DNSSEC is deployed in a domain Relying Party Applications MUST treat failure to authenticate signatures of CAA records or absence of CAA records whose presence is indicated as being equivalent to an inconsistent CAA record.

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:

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>.

	Value and meaning	Reference
-----	-----	-----
CAA	TBD1 Certification Authority Restriction	[RFCXXXX]

5.2. Certification Authority Authorization Properties

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

	Meaning	Reference
-----	-----	-----
path	Authorization Entry by Signature Path	[RFCXXXX]
policy	Authorization Entry by Certificate Policy	[RFCXXXX]

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

6. References

6.1. Normative References

[RFC1035]	Mockapetris, P., " Domain names - implementation and specification ", STD 13, RFC 1035, November 1987.
[RFC2119]	Bradner, S., " Key words for use in RFCs to Indicate Requirement Levels ", BCP 14, RFC 2119, March 1997.
[RFC4033]	Arends, R., Austein, R., Larson, M., Massey, D. and S. Rose, " DNS Security Introduction and Requirements ", RFC 4033, March 2005.
[RFC4055]	Schaad, J., Kaliski, B. and R. Housley, " Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile ", RFC 4055, June 2005.
[RFC5280]	Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. and W. Polk, " Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile ", RFC 5280, May 2008.
[RFC5395]	Eastlake, D., " Domain Name System (DNS) IANA Considerations ", RFC 5395, November 2008.
[X.509]	International Telecommunication Union , "ITU-T Recommendation X.509 (11/2008): Information technology - Open systems interconnection - The Directory: Public-key and attribute certificate frameworks ", ITU-T Recommendation X.509, November 2008.
[X.680]	International Telecommunication Union , "ITU-T Recommendation X.680 (11/2008): Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation ", ITU-T Recommendation X.680, November 2008.
[X.690]	International Telecommunication Union , "ITU-T Recommendation X.690 (11/2008): Information technology

	- Abstract Syntax Notation One (ASN.1): Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) ", ITU-T Recommendation X.690, November 2008.
--	--

6.2. Non Normative References

[RFC3642]	Legg, S., " Common Elements of Generic String Encoding Rules (GSER) Encodings ", RFC 3642, October 2003.
[RFC4648]	Josefsson, S., " The Base16, Base32, and Base64 Data Encodings ", RFC 4648, October 2006.
[NIST-ALGS]	National Institute of Standards and Technology , "Cryptographic Algorithm Registration", March 2009.

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.

[Appendix 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:

042017cc980f6a84fb15e5da3f32afea62360f4ca29627feed68739a13062defe804

The ODI in BASE64 format is

MDIGA1UEJQYJYIZIAWUDBAIBBCAXzJgPaoT7FeXaPzKv6mI2D0yilif+7WhzmhMGLe/
oBA==.

[Appendix 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'

*SHA-256 Digest Value

[Appendix B.](#) Example Certificates

The following certificates are used in the examples.

[Appendix B.1.](#) CA Certificate A

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

-----BEGIN CERTIFICATE-----

MIIDATCCAeugAwIBAgIBATALBgkqhkiG9w0BAQUwKDERMA8GA1UEChMIQWNtZSBJ
bmMxEzARBgNVBAMTCkV4YW1wbGUgQ0EwHhcNMTAxMTExMTgxMjAzWhcNMjAxMTA4
MTgxMjAzWjAoMREwDwYDVQQKEWhBY211IEluYzETMBEGA1UEAxMKRXhxbXBsZSBD
QTCCAR8wCwYJKoZIhvcNAQEBAA4IBDgAwggEJAoIBALHvos3yEe0ugR6Ae2rPATXA
pBYGK6BMzGTLkXCg6MZA9CZpfleZTZ/EgIKBwRJlIXvWdKwjMZ7GBByT+fdMDZp
7zKx64UZ4+CJm98NRjdugxovl8HhscIBXnhCHERGamp0U/f8Ho5W8eAxYLZ1XcIG
mB7mVknvolan9EqLEmYn+qHexGJPlpWfM4R4NKhVAATE6B1a9z5PCmoOgW9p0Vqic
SJ6CdAHKaa7JZS+sqNQDx57H8Q6R9lh52XXmJVVficxBp2K7C+Wvht45t68FG6f1
sXWuWDRYc6iUmOxZbzDDvIoFU0pAXESTdM0WvXKI8ZUaYBoZ7/YnSSTaseiW86sC
AwEAAAM9MDswDgYDVR0PAQEBAQDAgAEMA8GA1UdEwEBAQQFMAMBAQEwGAYDVR0g
BBEwDzANBgSRBgEEAYKUTYUaATALBgkqhkiG9w0BAQUdggEBAGcNiaQXdyiI9Y5e
Ps+XEYdKiWYvmSnRIfbUZuQwaQpPcj5cHzMe91CUZipGDNJYXwqWhIUtQAAGmtrq
ZGa4F9Yh0cPFAHBXPHXKGeM1hMtAR7Mv9kHu4DFIhb82200n4DdBIit8FNas5t/5
CbM6crDpWB5hjAsD37U+GZGvTJmag059VWjn9v90NcfCQ6YJ6AA5VKnmrV695VnL
dSPa9VS5RN6heJqU9tcbqPkaEP3MuJtd1QxB8Q34f9e1kTYXxc/dBJK1RQ0F4nc
Jc4NbJzakvFq+QcbzEqkhDMiXvjDV0JJt+GkFZrsREi6IgQY4DQHPv650Ivbr3uW
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	f7	0d	01	01	05	30	28	31
0020	11	30	0f	06	03	55	04	0a	13	08	41	63	6d	65	20	49
0030	6e	63	31	13	30	11	06	03	55	04	03	13	0a	45	78	61
0040	6d	70	6c	65	20	43	41	30	1e	17	0d	31	30	31	31	31
0050	31	31	38	31	32	30	33	5a	17	0d	32	30	31	31	30	38
0060	31	38	31	32	30	33	5a	30	28	31	11	30	0f	06	03	55
0070	04	0a	13	08	41	63	6d	65	20	49	6e	63	31	13	30	11
0080	06	03	55	04	03	13	0a	45	78	61	6d	70	6c	65	20	43
0090	41	30	82	01	1f	30	0b	06	09	2a	86	48	86	f7	0d	01
00a0	01	01	03	82	01	0e	00	30	82	01	09	02	82	01	00	b1
00b0	ef	a2	cd	f2	11	ed	2e	81	1e	80	7b	6a	cf	01	35	c0
00c0	a4	16	06	2b	a0	4c	cc	64	cb	91	70	a0	e8	c6	5a	1b
00d0	d0	99	a5	f9	5e	65	36	7f	12	02	0a	07	04	49	94	85
00e0	ef	59	d2	b0	8c	c6	7b	18	10	72	4f	e7	dd	30	36	69
00f0	ef	39	31	eb	85	19	e3	e0	89	9b	df	0d	46	37	6e	83
0100	1a	2f	97	c1	e1	b1	c2	01	5e	78	42	1c	44	60	6a	6a
0110	74	53	f7	fc	1e	8e	56	f1	e0	31	60	b6	75	5d	c2	06
0120	98	1e	e6	56	49	ef	a2	56	8d	f4	4a	a5	12	66	27	fa
0130	a1	de	c4	62	4f	96	95	85	99	1e	0d	2a	15	40	01	31
0140	3a	07	56	bd	cf	93	c2	9a	83	a0	5b	da	74	56	a8	9c
0150	48	9e	82	74	01	ca	69	ae	c9	65	2f	ac	a8	d4	03	c7
0160	9e	c7	f1	0e	91	f6	58	79	d9	75	e6	25	55	5f	89	cc
0170	41	a7	62	bb	0b	e5	af	86	de	39	b7	af	05	1b	a7	f5
0180	b1	75	ae	58	34	58	73	a8	94	98	ec	59	6f	30	c3	bc
0190	8a	05	53	4a	40	5c	44	93	74	c3	96	bd	72	88	f1	95
01a0	1a	60	1a	19	ef	f6	27	49	24	da	b1	e8	96	f3	ab	02
01b0	03	01	00	01	a3	3d	30	3b	30	0e	06	03	55	1d	0f	01
01c0	01	01	04	04	03	02	00	04	30	0f	06	03	55	1d	13	01
01d0	01	01	04	05	30	03	01	01	01	30	18	06	03	55	1d	20
01e0	04	11	30	0f	30	0d	06	0b	2b	06	01	04	01	82	94	4d
01f0	85	1a	01	30	0b	06	09	2a	86	48	86	f7	0d	01	01	05
0200	03	82	01	01	00	67	0d	89	a4	17	77	28	88	f5	8e	5e
0210	3e	cf	97	11	87	4a	89	66	2f	99	29	d1	21	f6	d4	66
0220	e4	16	69	0a	4f	72	3e	5c	1f	33	1e	f7	50	94	66	2a
0230	46	0c	d2	58	5f	0a	96	84	85	2d	40	00	06	9a	da	ea
0240	64	66	b8	17	d6	21	d1	c3	c5	00	70	57	3c	75	ca	19
0250	e3	35	84	cb	40	47	b3	2f	f6	41	ee	e0	31	48	85	bf
0260	36	d8	ed	27	e0	37	41	22	2b	7c	14	d6	ac	e6	df	f9
0270	09	b3	3a	72	b0	e9	58	1e	61	8c	0b	03	df	b5	3e	19
0280	91	af	4c	99	9a	83	4e	7d	55	68	e7	8e	ff	74	35	c7
0290	c2	43	a6	09	e8	00	39	54	a9	e6	ad	5e	bd	e5	59	cb
02a0	75	23	da	37	d5	52	e5	13	7a	85	e2	6a	53	db	5c	6e
02b0	a3	e4	00	43	f7	32	e2	6d	77	54	31	07	c4	37	e1	ff
02c0	5e	d6	44	d8	5f	17	3f	74	12	4a	d5	14	34	17	89	dc
02d0	25	ce	0d	6c	9c	da	92	f1	6a	f9	07	1b	cc	4a	a4	84
02e0	33	22	5e	f8	c3	57	42	49	b7	e1	a4	15	9a	ec	44	48
02f0	ba	22	04	18	e0	34	07	3e	fe	b9	38	8b	db	af	7b	96

0300 df 6f 5d 77 e8

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.

[Appendix 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.

[Appendix 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.

```
id-at OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) ds(5) 4 }

id-at-userCertificate OBJECT IDENTIFIER ::= { id-at 36 }
-- 06 03 55 04 24
    id-at-cACertificate OBJECT IDENTIFIER ::= { id-at 37 }
-- 06 03 55 04 25
TBS-PUBLIC-KEY-VALUE OBJECT IDENTIFIER ::= { ??? }
-- 06 xx xx xx xx
```

[Appendix 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.

```

hashAlgs OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3)
    nistAlgorithm(4) 2 }

id-sha256 OBJECT IDENTIFIER ::= { hashAlgs 1 }
    -- 06 09 60 86 48 01 65 03 04 02 01
id-sha384 OBJECT IDENTIFIER ::= { hashAlgs 2 }
    -- 06 09 60 86 48 01 65 03 04 02 02
id-sha512 OBJECT IDENTIFIER ::= { hashAlgs 3 }
    -- 06 09 60 86 48 01 65 03 04 02 03
id-sha224 OBJECT IDENTIFIER ::= { hashAlgs 4 }
    -- 06 09 60 86 48 01 65 03 04 02 04

```

Appendix 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 Hallam-Baker Comodo Group Inc. EMail:
philliph@comodo.com

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

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