CoRE Working Group
Internet-Draft
Intended status: Informational
Expires: January 7, 2020

H. Birkholz
Fraunhofer SIT
C. Bormann
Universitaet Bremen TZI
M. Pritikin
Cisco
R. Moskowitz
Huawei
July 06, 2019

Concise Identities
draft-birkholz-core-coid-02

Abstract

There is an increased demand of trustworthy claim sets -- a set of system entity characteristics tied to an entity via signatures -- in order to provide information. Claim sets represented via CBOR Web Tokens (CWT) can compose a variety of evidence suitable for constrained-node networks and to support secure device automation. This document focuses on sets of identifiers and attributes that are tied to a system entity and are typically used to compose identities appropriate for Constrained RESTful Environment (CoRE) authentication needs.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

X.509 certificates [RFC5280] and Secure Device Identifier [IEEE-802.1AR] are ASN.1 encoded Identity Documents and intended to be tied to a system entity uniquely identified via these Identity Documents. An Identity Document - in general, a public-key certificate - can be conveyed to other system entities in order to prove or authenticate the identity of the owner of the Identity Document. Trust in the proof can be established by mutual trust of the provider and assessor of the identity in a third party verification (TVP) provided, for example, by a certificate authority (CA) or its subsidiaries (sub CA).

The evidence a certificate comprises is typically composed of a set of claims that is signed using secret keys issued by a (sub) CA. The core set of claims included in a certificate - its attributes - are well defined in the X.509v3 specifications and IEEE 802.1AR.

This document summarizes the core set of attributes and provides a corresponding list of claims using concise integer labels to be used in claim sets for CBOR Web Tokens (CWT) [RFC8392]. A resulting

Concise Identity (CoID) is able to represent a signed set of claims that composes an Identity as defined in [RFC4949].

The objective of using CWT as a basis for the signed claim sets defined in this document is to gain more flexibility and at the same time more rigorously defined semantics for the signed claim sets. In addition, the benefits of using CBOR, COSE, and the corresponding CWT structure accrue, including more compact encoding and a simpler implementation in contrast to classical ASN.1 (DER/BER/PEM) structures and the X.509 complexity and uncertainty that has accreted since X.509 was released 29 years ago. One area where both the compactness and the definiteness are highly desirable is in Constrained-Node Networks [RFC7228], which may also make use of the Constrained Application Protocol (CoAP, [RFC7252]); however, the area of application of Concise Identities is not limited to constrained-node networks.

The present version of this document is a strawman that attempts to indicate the direction the work is intended to take. Not all inspirations this version takes from X.509 maybe need to be taken.

1.1. Terminology

This document uses terminology from [RFC8392] and therefore also [RFC7519], as well as from [RFC8152]. Specifically, we note:

Assertion: A statement made by an entity without accompanying evidence of its validity [X.1252].

Claim: A piece of information asserted about a subject. A claim is represented as a name/value pair consisting of a Claim Name and a Claim Value.

Claims are grouped into claims sets (represented here by a CWT), which need to be interpreted as a whole. Note that this usage is a bit different from idiomatic English usage, where a claim would stand on its own.

(Note that the current version of this draft is not very explicit about the relationship of identities and identifiers. To be done in next version.)

1.2. Requirements Language

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

BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Claims in a Concise Identity

A Concise Identity (CoID) is a CBOR Web Token [RFC8392] with certain claims present. It can be signed in a number of ways, including a COSE_Sign1 data object [RFC8152].

2.1. iss: CWT issuer

Optional: identifies the principal that is the claimant for the claims in the CoID ([RFC8392] Section 3.1.1, cf. Section 4.1.1 in [RFC7519]).

- Note that this is a StringOrURI (if it contains a ":" it needs to be a URI)

- For the "string" case (no ":"), there is no way to extract meaningful components from the string

- Make it a URI if it needs to be structured (not for routine retrieval, unless specified so by an application)

- If this URI looks like an HTTP or HTTPS URI then something retrievable by humans should exist there.

- Alternatively, some arithmetic can be applied to the URI (extract origin, add /.well-known/...) to find relevant information.

2.2. sub: CWT subject

Optional: identifies the principal that is the subject for the claims in the CoID ([RFC8392] Section 3.1.2, cf. Section 4.1.2 in [RFC7519]).

2.3. aud: CWT audience

Optional: identifies the recipients that the CoID is intended for ([RFC8392] Section 3.1.4, cf. Section 4.1.4 in [RFC7519]).

2.4. exp: CWT expiration time

Optional: the time on or after which the CoID must no longer be accepted for processing ([RFC8392] Section 3.1.4, cf. Section 4.1.4 in [RFC7519]).

2.5. nbf: CWT start of validity

Optional: the time before which the CoID must not be accepted for processing ([RFC8392] Section 3.1.5, cf. Section 4.1.5 in [RFC7519]).

2.6. iat: CWT time of issue

Optional: the creation time of the CoID ([RFC8392] Section 3.1.6, cf. Section 4.1.6 in [RFC7519]).

2.7. cti: CWT ID

The "cti" (CWT ID) claim provides a unique identifier for the CoID ([RFC8392] Section 3.1.7, cf. "jti" in Section 4.1.7 in [RFC7519]).

CWT IDs are intended to be unique within an application, so they need to be either coordinated between issuers or based on sufficient randomness (e.g., 112 bits or more).

2.8. cnf: CWT proof-of-possession key claim

The "cnf" claim identifies the key that can be used by the subject for proof-of-possession and provides parameters to identify the CWT Confirmation Method ([I-D.ietf-ace-cwt-proof-of-possession] Section 3.1).

3. The Essential Qualities of the Subject Claim

As highlighted above, the base definition of the representation of the "sub claim" is already covered by [RFC8392] and [RFC7519].

If claim sets need to be made about multiple subjects, the favored approach in CoID is to create multiple CoIDs, one each per subject.

In certain cases, the subject of a CoID needs to be an X.500 Distinguished Name in its full glory. These are sequences of relative names, where each relative name has a relative name type and a (text string) value.

```
dn-subject = [*(relativenametype, relativenamevalue)]
```

(RFC 5280 does not appear to specify how many DN components must be in a DN, so this uses a zero or more quantity.)

Any RelativeDistinguishedName values that are SETs of more than one AttributeTypeAndValue are translated into a sequence of pairs of the nametype used and each of the namevalues, lexicographically sorted.

To be able to map these to CBOR, we define labels for the relative name types listed in Section 4.1.2.4 of [RFC5280]:

(Note that unusual relative name types could be represented as OIDs; this would probably best be done by reviving the currently dormant [I-D.bormann-cbor-tags-oid].)

```
relativenametype = &(
  country: 1
  organization: 2
  organizational-unit: 3
  distinguished-name-qualifier: 4
  state-or-province-name: 5
  common-name: 6
  serial-number: 7
  locality: 8
  title: 9
  surname: 10
  given-name: 11
  initials: 12
  pseudonym: 13
  generation-qualifier: 14
)
```

The relative name values for these types are always expressed as CBOR text strings, i.e., in UTF-8:

```
relativenamevalue = text
```

4. Signature Envelope

The signature envelope [TBD: need not actually be envelope, may be detached, too] carries additional information, e.g., the signature, as well as the identification of the signature algorithm employed (COSE: alg). Additional information may pertain to the signature (as opposed to the claims being signed), e.g., a key id (COSE: kid) may be given in the header of the signature.

5. Processing Rules

(TBD: This should contain some discussion of the processing rules that apply for CoIDs. Some of this will just be pointers to [I-D.ietf-oauth-jwt-bcp].)

6. IANA Considerations

This document makes no requests of IANA

7. Security Considerations

8. References

8.1. Normative References

[I-D.ietf-ace-cwt-proof-of-possession]
Jones, M., Seitz, L., Selander, G., Erdtman, S., and H. Tschofenig, "Proof-of-Possession Key Semantics for CBOR Web Tokens (CWTs)", draft-ietf-ace-cwt-proof-of-possession-06 (work in progress), February 2019.

[I-D.ietf-oauth-jwt-bcp]
Sheffer, Y., Hardt, D., and M. Jones, "JSON Web Token Best Current Practices", draft-ietf-oauth-jwt-bcp-06 (work in progress), June 2019.

[I-D.ietf-rats-eat]
Mandyam, G., Lundblade, L., Ballesteros, M., and J. O'Donoghue, "The Entity Attestation Token (EAT)", draft-ietf-rats-eat-01 (work in progress), July 2019.

[I-D.tschofenig-rats-psa-token]
Tschofenig, H., Frost, S., Brossard, M., and A. Shaw, "Arm's Platform Security Architecture (PSA) Attestation Token", draft-tschofenig-rats-psa-token-01 (work in progress), April 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[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, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5755] Farrell, S., Housley, R., and S. Turner, "An Internet Attribute Certificate Profile for Authorization", RFC 5755, DOI 10.17487/RFC5755, January 2010, <https://www.rfc-editor.org/info/rfc5755>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

[X.1252] "ITU-T X.1252 (04/2010)", n.d..

8.2. Informative References

[I-D.bormann-cbor-tags-oid] Bormann, C. and S. Leonard, "Concise Binary Object Representation (CBOR) Tags and Techniques for Object Identifiers, UUIDs, Enumerations, Binary Entities, Regular Expressions, and Sets", draft-bormann-cbor-tags-oid-06 (work in progress), March 2017.

[IEEE-802.1AR] "ISO/IEC/IEEE International Standard for Information technology -- Telecommunications and information exchange between systems -- Local and metropolitan area networks -- Part 1AR: Secure device identity", IEEE standard, DOI 10.1109/ieeestd.2014.6739984, n.d..

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <https://www.rfc-editor.org/info/rfc4949>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

Appendix A. Common Terminology on Identity Documents

To illustrate the purpose and intent of Identity Documents, typically, terms, such as certificates, certificate chains/paths and trust anchors, are used. To provide more context and for the convenience of the reader, three sources of definitions are highlighted in this section.

A.1. Terms Specified in IEEE 802.1AR

1. a certificate is "a digitally signed object that binds information identifying an entity that possesses a secret private key to the corresponding public key."

2. a certificate chain is "an ordered list of intermediate certificates that links an end entity certificate ([...] a DevID certificate) to a trust anchor."

3. a trust anchor is "a Certificate Authority that is trusted and for which the trusting party holds information, usually in the form of a self-signed certificate issued by the trust anchor".

A.2. Terms Specified in RFC 4949

1. a public-key certificate is "a digital certificate that binds a system entity's identifier to a public key value, and possibly to additional, secondary data items; i.e., a digitally signed data structure that attests to the ownership of a public key".

2. a certification path is "a linked sequence of one or more public-key certificates [...] that enables a certificate user to verify the signature on the last certificate in the path, and thus enables the user to obtain (from that last certificate) a certified public key, or certified attributes, of the system entity that is the subject of that last certificate".

3. a trust anchor is "a CA that is the subject of a trust anchor certificate or otherwise establishes a trust anchor key". Correspondingly, a trust anchor has a trust anchor certificate that "is a public-key certificate that is used to provide the first public key in a certification path".

A.3. Terms specified in ISO/IEC 9594-8:2017

1. a public-key certificate is "the public key of an entity, together with some other information, rendered unforgeable by digital signature with the private key of the certification authority (CA) that issued it".

2. a certification path is "an ordered list of one or more public-key certificates, starting with a public-key certificate signed by the trust anchor, and ending with the end-entity public-key certificate to be validated. All intermediate public-key certificates, if any, are certification authority (CA) certificates in which the subject of the preceding public-key certificate is the issuer of the following public-key certificate".

3. a trust anchor is "an entity that is trusted by a relying party and used for validating public-key certificates".

Appendix B. Concise Identities and Trust Relationships

Following the terminology highlighted above, Concise Identities are signed CBOR Web Tokens that compose public-key Identity Documents based on asymmetric key pairs, potentially including additional assertions: claims that are secondary data items.

In the context of certification paths, the "last certificate" in the certification path is the Identity Document that resides on the system component, which presents its Identity Document to relying partyies in order to be authenticated. The "first certificate" in the certification path resides on the trust anchor.

In order to be able to rely on the trust put into the Identity Document presented to relying parties, these have to put trust into two assumptions first:

- the corresponding trust anchor (certificate) is trusted. In consequence, the consumer of the Identity Document requires a basis for decision whether to rely on the trust put in the trust anchor certificate, or not (e.g. via policies or a known certification paths).

- the secret key included in the system component that is presenting its Identity Document is protected. In consequence, the secret key has to be stored in a shielded location. Type and quality of the protection or shielding or even its location are assertions that can be included as secondary data items in the Identity Document.

In summary, a path of trust relationships between a system component's Identity Document and a trusted authority's Identity Document is required to enable transitive trust in the system component that presents the Identity Document.

Appendix C. Concise Identity (CoID) CDDL Data Definition based on RFC 5280

COSE MUST be used to sign this CoID template flavor.

"signatureAlgorithm" and "signature" are not part of the CoID map but of the COSE envelope.

```
CoID = { version: uint .range 1..3 ; (8)
         issuer: text, ; iss(1)
         subject: text / bytes, ; sub(2)
         notAfter: uint, ; exp(4)
         notBefore: uint ; nbf(5)
         serialNumber: uint,; (7)
         subjectPublicKeyInfo: [ algorithm: COSE-Algorithm-Value,
                                   subjectPublicKey: bytes,
                               ], ;(9)
         ? extensions: [ + [ extension-id: uint / registeredID,
                               extension-value: any,
                               ? criticality: bool,
                             ]
                         ], ;(0)
       }

COSE-Algorithm-Value = uint .size 0..2 / nint .size 0..2
registeredID = [ + uint ] ; OID

extensions = 0
issuer = 1
subject = 2
notAfter = 4
notBefore = 5
serialNumber = 7
version = 8
subjectPublicKeyInfo = 9
```

Appendix D. Concise Secure Device Identifier (CoDeID) based on IEEE 802.1AR-2018

This section illustrates the context and background of Secure Device Identifiers.

D.1. The Intended Use of DevIDs

IEEE 802.1AR Secure Device Identifier are a specific subset of X.509 Identity Documents that are intended to "authenticate a device's identity", where the corresponding Identity Document is "cryptographically bound to that device". In this context, "cryptographically bound" means that the Identity Document is "constructed using cryptographic operations to combine a secret with other arbitrary data objects such that it may be proven that the result could only be created by an entity having knowledge of the secret."

While the intent of using X.509 Identity Documents as Device Identifiers starts to blur the line between authentication and authorization, the specification of IEEE 802.1AR Identity Documents provides a meaningful subset of assertions that can be used to identify one or more system components. The following CDDL data definition maps the semantics of an RFC 5280 Public Key Infrastructure Certificate Profile, which provides the basis for the Secure Device Identifier semantics. Both are mapped to a CWT representation.

D.2. DevID Flavors

In order to provide consistent semantics for the claims as defined below, understanding the distinction of IDevIDs (mandatory representation capabilities) and LDevIDs (recommended representation capabilities) is of the essence.

Both flavors of Secure Device Identifiers share most of their assertion semantics (claim sets).

IDevIDs are the _initially_ Secure Device Identifiers that "are normally created during manufacturing or initial provisioning" and are "installed on the device by the manufacturer". IDevIDs are intended to be globally unique and to be stored in a way that protects it from modification (typically, a shielded location). It is important to note that a potential segregation of a manufacturer into separate supply chain/tree entities is not covered by the 802.1AR specification.

LDevIDs are the _local significant_ Secure Device Identifiers that are intended to be "unique in the local administrative domain in which the device is used". In essence, LDevIDs "can be created at any time [after IDevID provisioning], in accordance with local policies". An "LDevID is bound to the device in a way that makes it infeasible for it to be forged or transferred to a device with a

different IDevID without knowledge of the private key used to effect the cryptographic binding".

D.3. Privacy

The exposition iof IDevID Identity Documents enables global unique identification of a system component. To mitigate the obvious privacy LDevIDs may also be used as the sole identifier (by disabling the IDevID) to assure the privacy of the user of a DevID and the equipment in which it is installed.

D.4. Concise DevID CDDL data definition (sans COSE header)

COSE MUST be used to sign this DevID flavor, if represented via CoID.

"signature" and "signatureValue" are not part of the CoID map but of the COSE envelope.

"AlgorithmIdentifier" and corresponding "algorithm" and "parameters" should be part of the COSE envelope.

```
CoDeID = { version: 3, ;(8)
           serialNumber: uint,(7)
           issuer: text, ; iss(1)
           notAfter: uint, ; exp(4)
           notBefore: uint ; nbf(5)
           subject: text / URI, ; sub(2)
           subjectPublicKeyInfo: [ algorithm: COSE-Algorithm-Value,
                                   subjectPublicKey: bytes,
                                 ], ;(9)
           signatureAlgorithm: COSE-Algorithm-Value ; 802.1ar-2018 states
                                                    ; this must be identical
                                                    ; to cose sig-alg (rm?)
           authorityKeyidentifier: bytes, ; all, non-critical,
           ? subjectKeyIdentifier; bytes, ; only intermediates, non-critical
           ? keyUsage : [ bitmask: bytes .size 1,
                          ? criticality: bool,
                        ]
           ? subjectAltName: text / iPAddress / registeredID,
           ? HardwareModuleName: [ hwType: registeredID,
                                   hwSerialNum: bytes,
                                 ],
           ? extensions: [ + [ extension-id: uint,
                               extension-value: any,
                               ? criticality: bool,
                             ],
         }

COSE-Algorithm-Value = uint .size 0..2 / nint .size 0..2
iPAddress = bytes .size 4 / bytes .size 16
registeredID = [ + uint ] ; OID

extensions = 0
issuer = 1
subject = 2
notAfter = 4
notBefore = 5
serialNumber = 7
version = 8
subjectPublicKeyInfo = 9
signatureAlgorithm = 10
authorityKeyidentifier = 11
subjectKeyIdentifier = 12
keyUsage = 13 ; could move to COSE header?
subjectAltName = 14
HardwareModuleName = 15
```

Appendix E. Concise Attribute Documents

Additional well-defined sets of characteristics can be bound to Identity Documents [RFC5280] or Secure Device Identifiers [IEEE-802.1AR]. CDDL specifications to define these can be found in the corresponding appendices above and the Profile for X.509 Internet Attribute Certificates is defined in [RFC5755].

Essentially, various existing CWT specializations, such as the Entity Attestation Tokens [I-D.ietf-rats-eat] and the Platform Security Architecture Tokens [I-D.tschofenig-rats-psa-token] already compose a type of Attribute Certificates today. In order to bridge the gap between these already existing Concise Attribute Documents and binding them to traditional X.509 Identity Documents (pub-key certificates), a sub claim referencing the corresponding Identity Document has to be included in the signed CBOR Web Token (flavor). The mechanics of how to handle the corresponding key material is also defined in [RFC5755] (and this document will elaborate on these in future versions).

With respect to Concise Identity Documents the dn-subject claim should be used. If a Concise Attribute Certificate has to refer to a traditional ASN.1 encoded X.509 Identity document the subject claim should be used. This procedure provides a migration path from ASN.1 encoded Identity documents [RFC5280] to CBOR encoded Concise Identity documents that allows to bind Concise Attribute Documents, such as EAT or PSA Tokens to both kinds of certificates. In an ideal scenario CBOR encoding in the form of [RFC8392] is used both for Concise Identity Documents and Concise Attribute Documents. The alternate uses of subject claims or dn-subject claims addresses the fact that the vast majority of constrained node devices still use an ASN.1 encoding and simplified interoperability between CBOR encoded and ASN.1 encoded documents is still of essence today.

Appendix F. Attic

Notes and previous content that will be pruned in next versions.

F.1. Examples of claims taken from IEEE 802.1AR identifiers

This appendix briefly discusses common fields in a X.509 certificate or an IEEE 802.1AR Secure Device Identifier and relates them to claims in a CoID.

The original purpose of X.509 was only to sign the association between a name and a public key. In principle, if something else needs to be signed as well, CMS [RFC5652] is required. This principle has not been strictly upheld over time; this is

demonstrated by the growth of various extensions to X.509 certificates that might or might not be interpreted to carry various additional claims.

This document details only the claim sets for CBOR Web Tokens that are necessary for authentication. The plausible integration or replacement of ASN.1 formats in enrollment protocols, (D)TLS handshakes and similar are not in scope of this document.

Subsections in this appendix are marked by the ASN.1 Object Identifier (OID) typically used for the X.509 item. [TODO: Make this true; there are still some section numbers.]

F.1.1. 7.2.1 version

The version field is typically not employed usefully in an X.509 certificate, except possibly in legacy applications that accept original (pre-v3) X.509 certificates.

Generally, the point of versioning is to deliberately inhibit interoperability (due to semantic meaning changes). CoIDs do not employ versioning. Where future work requires semantic changes, these will be expressed by making alternate kinds of claims.

F.1.2. 7.2.2 serialNumber

Covered by cti claim.

F.1.3. 7.2.3 signature

The signature, as well as the identification of the signature algorithm, are provided by the COSE container (e.g., COSE_Sign1) used to sign the CoID's CWT.

F.1.4. 7.2.4 issuer Name

Covered by iss claim.

F.1.5. 7.2.5 authoritykeyidentifier

Covered by COSE kid in signature, if needed.

F.1.6. 7.2.7.1 notBefore

Covered by nbf claim.

F.1.7. 7.2.7.2 notAfter

Covered by exp claim.

For Secured Device identifiers, this claim is typically left out.

- get a new one whenver you think you need it ("normal path")

- nonced ocsp? might benefit from a more lightweight freshness verification of existing signed assertion - exploration required!

- (first party only verfiable freshness may be cheaper than third-party verifiable?)

F.1.8. 7.2.10 subjectPublicKeyInfo

Covered by cnf claim.

F.1.9. 7.2.11 signatureAlgorithm

In COSE_Sign1 envelope.

F.1.10. 7.2.12 signatureValue

In COSE_Sign1 envelope.

F.2. Examples of claims taken from X.509 certificates

Most claims in X.509 certificates take the form of certificate extensions. This section reviews a few common (and maybe not so common) certificate extensions and assesses their usefulness in signed claim sets.

F.2.1. 2.5.29.35 - Authority Key Identifier

Used in certificate chaining. Can be mapped to COSE "kid" of the issuer.

F.2.2. 2.5.29.14 - Subject Key Identifier

Used in certificate chaining. Can be mapped to COSE "kid" in the "cnf" (see Section 3.4 of [I-D.ietf-ace-cwt-proof-of-possession]).

F.2.3. 2.5.29.15 - Key Usage

Usage information for a key claim that is included in the signed claims. Can be mapped to COSE "key_ops" [TBD: Explain details].

F.2.4. 2.5.29.37 - Extended key usage

Can include additional usage information such as 1.3.6.1.5.5.7.3.1 for TLS server certificates or 1.3.6.1.5.5.7.3.2 for TLS client certificates.

F.2.5. 1.3.6.1.5.5.7.1.1 - Authority Information Access

More information about the signer. May include a pointer to signers higher up in the certificate chain (1.3.6.1.5.5.7.48.2), typically in the form of a URI to their certificate.

F.2.6. 1.3.6.1.4.1.311.20.2 - Certificate Template Name Domain Controller (Microsoft)

This is an example for many ill-defined extensions that are on some arcs of the OID space somewhere.

E.g., the UCS-2 string (ASN.1 BMPString) "IPSECIntermediateOffline"

Appendix G. Graveyard

Items and Content that was already discarded.

G.1. 7.2.9 subjectAltName

(See "sub").

G.2. 7.2.13 extensions

Extensions are handled by adding CWT claims to the CWT.

G.3. 2.5.29.31 - CRL Distribution Points

Usually URIs of places where a CRL germane to the certificate can be obtained. Other forms of validating claim sets may be more appropriate than CRLs for the applications envisaged here.

(Might be replaced by a more general freshness verification approach later. For example one could define a generic "is this valid" request to an authority.)

G.4. 2.5.29.17 - Subject Alternative Name

Additional names for the Subject.

These may be an "OtherName", i.e. a mistery blob "defined by" an ASN.1 OID such as 1.3.6.1.4.1.9.21.2.3, or one out of a few formats

such as URIs (which may, then, turn out not to be really URIs). Naming subjects obviously is a major issue that needs attention.

G.5. 2.5.29.19 - Basic Constraints

Can identify the key claim as that for a CA, and can limit the length of a certificate path. Empty in all the examples analyzed.

Any application space can define new fields / claims as appropriate and use them. There is no need for the underlying structure to define an additional extension method for this. Instead, they can use the registry as defined in Section 9.1 of [RFC8392].>

Acknowledgements

Authors' Addresses

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt 64295
Germany

Email: henk.birkholz@sit.fraunhofer.de

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

Max Pritikin
Cisco

Email: pritikin@cisco.com

Robert Moskowitz
Huawei
Oak Park, MI 48237

Email: rgm@labs.htt-consult.com