

Workgroup: ANIMA Working Group

Internet-Draft: draft-ietf-anima-rfc8366bis-05

Published: 25 January 2023

Intended Status: Standards Track

Expires: 29 July 2023

Authors: K. Watsen	M. Richardson	M. Pritikin
Watsen Networks	Sandelman Software	Cisco Systems
T. Eckert	Q. Ma	
Huawei	Huawei	

A Voucher Artifact for Bootstrapping Protocols

Abstract

This document defines a strategy to securely assign a pledge to an owner using an artifact signed, directly or indirectly, by the pledge's manufacturer. This artifact is known as a "voucher".

This document defines an artifact format as a YANG-defined JSON or CBOR document that has been signed using a variety of cryptographic systems.

The voucher artifact is normally generated by the pledge's manufacturer (i.e., the Manufacturer Authorized Signing Authority (MASA)).

This document updates RFC8366, merging a number of extensions into the YANG. The RFC8995 voucher request is also merged into this document.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-anima-rfc8366bis/>.

Discussion of this document takes place on the anima Working Group mailing list (<mailto:anima@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/anima/>. Subscribe at <https://www.ietf.org/mailman/listinfo/anima/>.

Source for this draft and an issue tracker can be found at <https://github.com/anima-wg/voucher>.

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 <https://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 29 July 2023.

Copyright Notice

Copyright (c) 2023 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Terminology](#)
- [3. Requirements Language](#)
- [4. Survey of Voucher Types](#)
- [5. Changes since RFC8366](#)
- [6. Voucher Artifact](#)
 - [6.1. Tree Diagram](#)
 - [6.2. Examples](#)
 - [6.3. YANG Module](#)
 - [6.4. ietf-voucher SID values](#)
 - [6.5. CMS Format Voucher Artifact](#)
- [7. Voucher Request Artifact](#)
 - [7.1. Tree Diagram](#)
 - [7.2. "ietf-voucher-request" Module](#)
 - [7.3. ietf-voucher-request SID values](#)
- [8. Design Considerations](#)
 - [8.1. Renewals Instead of Revocations](#)
 - [8.2. Voucher Per Pledge](#)
- [9. Security Considerations](#)
 - [9.1. Clock Sensitivity](#)

- [9.2. Protect Voucher PKI in HSM](#)
- [9.3. Test Domain Certificate Validity When Signing](#)
- [9.4. YANG Module Security Considerations](#)
- [10. IANA Considerations](#)
 - [10.1. The IETF XML Registry](#)
 - [10.2. The YANG Module Names Registry](#)
 - [10.3. The Media Types Registry](#)
 - [10.4. The SMI Security for S/MIME CMS Content Type Registry](#)
- [11. References](#)
 - [11.1. Normative References](#)
 - [11.2. Informative References](#)
- [Acknowledgements](#)
- [Authors' Addresses](#)

1. Introduction

This document defines a strategy to securely assign a candidate device (pledge) to an owner using an artifact signed, directly or indirectly, by the pledge's manufacturer, i.e., the Manufacturer Authorized Signing Authority (MASA). This artifact is known as the "voucher".

The voucher artifact is a JSON [[RFC8259](#)] document that conforms with a data model described by YANG [[RFC7950](#)]. It may also be serialized to CBOR [[CBOR](#)]. It is encoded using the rules defined in [[RFC8259](#)], and is signed using (by default) a CMS structure [[RFC5652](#)].

The primary purpose of a voucher is to securely convey a certificate, the "pinned-domain-cert" (and constrained variations), that a pledge can use to authenticate subsequent interactions. A voucher may be useful in several contexts, but the driving motivation herein is to support secure onboarding mechanisms. Assigning ownership is important to device onboarding mechanisms so that the pledge can authenticate the network that is trying to take control of it.

The lifetimes of vouchers may vary. In some onboarding protocols, the vouchers may include a nonce restricting them to a single use, whereas the vouchers in other onboarding protocols may have an indicated lifetime. In order to support long lifetimes, this document recommends using short lifetimes with programmatic renewal, see [Section 8.1](#).

This document only defines the voucher artifact, leaving it to other documents to describe specialized protocols for accessing it. Some onboarding protocols using the voucher artifact defined in this document include: [[ZERO-TOUCH](#)], [[SECUREJOIN](#)], and [[BRSKI](#)].

2. Terminology

This document uses the following terms:

Artifact: Used throughout to represent the voucher as instantiated in the form of a signed structure.

Bootstrapping: See Onboarding.

Domain: The set of entities or infrastructure under common administrative control. The goal of the onboarding protocol is to enable a pledge to discover and join a domain.

Imprint: The process where a device obtains the cryptographic key material to identify and trust future interactions with a network. This term is taken from Konrad Lorenz's work in biology with new ducklings: "during a critical period, the duckling would assume that anything that looks like a mother duck is in fact their mother" [[Stajano99theresurrecting](#)]. An equivalent for a device is to obtain the fingerprint of the network's root certification authority certificate. A device that imprints on an attacker suffers a similar fate to a duckling that imprints on a hungry wolf. Imprinting is a term from psychology and ethology, as described in [[imprinting](#)].

Join Registrar (and Coordinator): A representative of the domain that is configured, perhaps autonomically, to decide whether a new device is allowed to join the domain. The administrator of the domain interfaces with a join registrar (and Coordinator) to control this process. Typically, a join registrar is "inside" its domain. For simplicity, this document often refers to this as just "registrar".

MASA (Manufacturer Authorized Signing Authority): The entity that, for the purpose of this document, signs the vouchers for a manufacturer's pledges. In some onboarding protocols, the MASA may have an Internet presence and be integral to the onboarding process, whereas in other protocols the MASA may be an offline service that has no active role in the onboarding process.

Onboarding: In previous documents the term "bootstrapping" has been used to describe mechanisms such as [[BRSKI](#)]. The industry has

however, converged upon the term "onboarding", and this document uses that term throughout.

Owner: The entity that controls the private key of the "pinned-domain-cert" certificate conveyed by the voucher.

Pledge: The prospective device attempting to find and securely join a domain. When shipped, it only trusts authorized representatives of the manufacturer.

Registrar: See join registrar.

TOFU (Trust on First Use): Where a pledge device makes no security decisions but rather simply trusts the first domain entity it is contacted by. Used similarly to [\[RFC7435\]](#). This is also known as the "resurrecting duckling" model.

Voucher: A signed statement from the MASA service that indicates to a pledge the cryptographic identity of the domain it should trust.

3. 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.

4. Survey of Voucher Types

A voucher is a cryptographically protected statement to the pledge device authorizing a zero-touch "imprint" on the join registrar of the domain. The specific information a voucher provides is influenced by the onboarding use case.

The voucher can impart the following information to the join registrar and pledge:

Assertion Basis: Indicates the method that protects the imprint (this is distinct from the voucher signature that protects the voucher itself). This might include manufacturer-asserted ownership verification, assured logging operations, or reliance on pledge endpoint behavior such as secure root of trust of measurement. The join registrar might use this information. Only some methods are normatively defined in this document. Other methods are left for future work.

Authentication of Join Registrar: Indicates how the pledge can authenticate the join registrar. This document defines a

mechanism to pin the domain certificate. Pinning a symmetric key, a raw key, or "CN-ID" or "DNS-ID" information (as defined in [RFC6125]) is left for future work.

Anti-Replay Protections: Time- or nonce-based information to constrain the voucher to time periods or bootstrap attempts.

A number of onboarding scenarios can be met using differing combinations of this information. All scenarios address the primary threat of a Man-in-The-Middle (MiTM) registrar gaining control over the pledge device. The following combinations are "types" of vouchers:

	Assertion		Registrar ID		Validity	
Voucher Type	Logged	Verified	Trust Anchor	CN-ID or DNS-ID	RTC	Nonce
Audit	X		X			X
Nonceless Audit	X		X		X	
Owner Audit	X	X	X		X	X
Owner ID		X	X	X	X	
Bearer out-of-scope	X		wildcard	wildcard	optional	opt

Table 1

NOTE: All voucher types include a 'pledge ID serial-number' (not shown here for space reasons).

Audit Voucher: An Audit Voucher is named after the logging assertion mechanisms that the registrar then "audits" to enforce local policy. The registrar mitigates a MiTM registrar by auditing that an unknown MiTM registrar does not appear in the log entries. This does not directly prevent the MiTM but provides a response mechanism that ensures the MiTM is unsuccessful. The advantage is that actual ownership knowledge is not required on the MASA service.

Nonceless Audit Voucher: An Audit Voucher without a validity period statement. Fundamentally, it is the same as an Audit Voucher except that it can be issued in advance to support network partitions or to provide a permanent voucher for remote deployments.

Ownership Audit Voucher: An Audit Voucher where the MASA service has verified the registrar as the authorized owner. The MASA service mitigates a MiTM registrar by refusing to generate Audit

Vouchers for unauthorized registrars. The registrar uses audit techniques to supplement the MASA. This provides an ideal sharing of policy decisions and enforcement between the vendor and the owner.

Ownership ID Voucher: Named after inclusion of the pledge's CN-ID or DNS-ID within the voucher. The MASA service mitigates a MiTM registrar by identifying the specific registrar (via WebPKI) authorized to own the pledge.

Bearer Voucher: A Bearer Voucher is named after the inclusion of a registrar ID wildcard. Because the registrar identity is not indicated, this voucher type must be treated as a secret and protected from exposure as any 'bearer' of the voucher can claim the pledge device. Publishing a nonceless bearer voucher effectively turns the specified pledge into a "TOFU" device with minimal mitigation against MiTM registrars. Bearer vouchers are out of scope.

5. Changes since RFC8366

[[RFC8366](#)] was published in 2018 during the development of [[BRSKI](#)], [[ZERO-TOUCH](#)] and other work-in-progress efforts. Since then the industry has matured significantly, and the in-the-field activity which this document supports has become known as *onboarding* rather than *bootstrapping*.

The focus of [[BRSKI](#)] was onboarding of ISP and Enterprise owned wired routing and switching equipment, with IoT devices being a less important aspect. [[ZERO-TOUCH](#)] has focused upon onboarding of CPE equipment like cable modems and other larger IoT devices, again with smaller IoT devices being of less import.

Since [[BRSKI](#)] was published there is now a mature effort to do application-level onboarding of constrained IoT devices defined by The Thread and Fairhair (now OCF) consortia. The [[cBRSKI](#)] document has defined a version of [[BRSKI](#)] that is useable over constrained 802.15.4 networks using CoAP and DTLS, while [[I-D.selander-ace-ake-authz](#)] provides for using CoAP and EDHOC on even more constrained devices with very constrained networks.

[[PRM](#)] has created a new methodology for onboarding that does not depend upon a synchronous connection between the Pledge and the Registrar. This mechanism uses a mobile Registrar Agent that works to collect and transfer signed artifacts via physical travel from one network to another.

Both [[cBRSKI](#)] and [[PRM](#)] require extensions to the Voucher Request and the resulting Voucher. The new attributes are required to carry the additional attributes and describe the extended semantics. In

addition [cBRSKI] uses the serialization mechanism described in [YANGCBOR] to produce significantly more compact artifacts.

When the process to define [cBRSKI] and [PRM] was started, there was a belief that the appropriate process was to use the [RFC8040] *augment* mechanism to further extend both the voucher request [BRSKI] and voucher [RFC8366] artifacts. However, [PRM] needs to extend an enumerated type with additional values and *augment* can not do this, so that was initially the impetus for this document.

An attempt was then made to determine what would happen if one wanted to have a constrained version of the [PRM] voucher artifact. The result was invalid YANG, with multiple definitions of the core attributes from the [RFC8366] voucher artifact. After some discussion, it was determined that the *augment* mechanism did not work, nor did it work better when [RFC8040] yang-data was replaced with the [RFC8971] structure mechanisms.

After significant discussion the decision was made to simply roll all of the needed extensions up into this document as "RFC8366bis".

This document therefore represents a merge of YANG definitions from [RFC8366], the voucher-request from [BRSKI], and then extensions to each of these from [cBRSKI] and [PRM]. There are some difficulties with this approach: this document does not attempt to establish rigorous semantic definitions for how some attributes are to be used, referring normatively instead to the other relevant documents.

6. Voucher Artifact

The voucher's primary purpose is to securely assign a pledge to an owner. The voucher informs the pledge which entity it should consider to be its owner.

This document defines a voucher that is a JSON-encoded or CBOR-encoded instance of the YANG module defined in [Section 6.3](#) that has been, by default, CMS signed. [cBRSKI] defines how to encode with CBOR and sign the voucher with [COSE], while [jBRSKI] explains how to use [JWS] to do JSON signatures.

This format is described here as a practical basis for some uses (such as in NETCONF), but more to clearly indicate what vouchers look like in practice. This description also serves to validate the YANG data model.

[RFC8366] defined a media type and a filename extension for the CMS-encoded JSON type. Which type of voucher is expected is signaled (where possible) in the form of a MIME Content-Type, an HTTP Accept: header, or more mundane methods like use of a filename extension when a voucher is transferred on a USB key.

6.1. Tree Diagram

The following tree diagram illustrates a high-level view of a voucher document. The notation used in this diagram is described in [\[RFC8340\]](#). Each node in the diagram is fully described by the YANG module in [Section 6.3](#). Please review the YANG module for a detailed description of the voucher format.

module: ietf-voucher

structure voucher:

```
+-- voucher
  +-- created-on?          yang:date-and-time
  +-- expires-on?         yang:date-and-time
  +-- assertion?          enumeration
  +-- serial-number        string
  +-- idevid-issuer?       binary
  +-- pinned-domain-cert?  binary
  +-- domain-cert-revocation-checks? boolean
  +-- nonce?              binary
  +-- pinned-domain-pubk?  binary
  +-- pinned-domain-pubk-sha256? binary
  +-- last-renewal-date?   yang:date-and-time
```

6.2. Examples

This section provides voucher examples for illustration purposes. These examples conform to the encoding rules defined in [\[RFC8259\]](#).

The following example illustrates an ephemeral voucher (uses a nonce). The MASA generated this voucher using the 'logged' assertion type, knowing that it would be suitable for the pledge making the request.

```
{
  "ietf-voucher:voucher": {
    "created-on": "2016-10-07T19:31:42Z",
    "assertion": "logged",
    "serial-number": "JADA123456789",
    "idevid-issuer": "base64encodedvalue==",
    "pinned-domain-cert": "base64encodedvalue==",
    "nonce": "base64encodedvalue=="
  }
}
```

The following example illustrates a non-ephemeral voucher (no nonce). While the voucher itself expires after two weeks, it presumably can be renewed for up to a year. The MASA generated this voucher using the 'verified' assertion type, which should satisfy all pledges.

```
{
  "ietf-voucher:voucher": {
    "created-on": "2016-10-07T19:31:42Z",
    "expires-on": "2016-10-21T19:31:42Z",
    "assertion": "verified",
    "serial-number": "JADA123456789",
    "idevid-issuer": "base64encodedvalue==",
    "pinned-domain-cert": "base64encodedvalue==",
    "domain-cert-revocation-checks": "true",
    "last-renewal-date": "2017-10-07T19:31:42Z"
  }
}
```

6.3. YANG Module

<CODE BEGINS>

```
module ietf-voucher {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-voucher";
  prefix vch;

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-yang-structure-ext {
    prefix sx;
  }

  organization
    "IETF ANIMA Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/anima/>
    WG List:  <mailto:anima@ietf.org>
    Author:   Kent Watsen
              <mailto:kwatsen@juniper.net>
    Author:   Max Pritikin
              <mailto:pritikin@cisco.com>
    Author:   Michael Richardson
              <mailto:mcr+ietf@sandelman.ca>
    Author:   Toerless Eckert
              <mailto:tte+ietf@cs.fau.de>";
  description
    "This module defines the format for a voucher, which is
    produced by a pledge's manufacturer or delegate (MASA)
    to securely assign a pledge to an 'owner', so that the
    pledge may establish a secure connection to the owner's
    network infrastructure.
```

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 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.

Copyright (c) 2023 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC 8366; see the RFC itself for full legal notices.";

```
revision 2023-01-10 {
  description
    "updated to support new assertion enumerated type";
  reference
    "RFC ZZZZ Voucher Profile for Bootstrapping Protocols";
}

// Top-level statement
sx:structure voucher {
  uses voucher-artifact-grouping;
}

// Grouping defined for future augmentations

grouping voucher-artifact-grouping {
  description
    "Grouping to allow reuse/extensions in future work.";
  container voucher {
    description
      "A voucher assigns a pledge to an owner using
        the (pinned-domain-cert) value.";
    leaf created-on {
      type yang:date-and-time;
      mandatory false;
      description
        "A value indicating the date this voucher was created.
          This node is primarily for human consumption and auditing.
          Future work MAY create verification requirements based on
          this node.";
    }
    leaf expires-on {
      type yang:date-and-time;
      must 'not(..nonce)';
      description
        "A value indicating when this voucher expires. The node is
          optional as not all pledges support expirations, such as
          pledges lacking a reliable clock.

          If this field exists, then the pledges MUST ensure that
          the expires-on time has not yet passed. A pledge without
          an accurate clock cannot meet this requirement.

          The expires-on value MUST NOT exceed the expiration date
          of any of the listed 'pinned-domain-cert' certificates.";
```

```

}
leaf assertion {
  type enumeration {
    enum verified {
      value 0;
      description
        "Indicates that the ownership has been positively
        verified by the MASA (e.g., through sales channel
        integration).";
    }
    enum logged {
      value 1;
      description
        "Indicates that the voucher has been issued after
        minimal verification of ownership or control. The
        issuance has been logged for detection of
        potential security issues (e.g., recipients of
        vouchers might verify for themselves that unexpected
        vouchers are not in the log). This is similar to
        unsecured trust-on-first-use principles but with the
        logging providing a basis for detecting unexpected
        events.";
    }
    enum proximity {
      value 2;
      description
        "Indicates that the voucher has been issued after
        the MASA verified a proximity proof provided by the
        device and target domain. The issuance has been
        logged for detection of potential security issues.
        This is stronger than just logging, because it
        requires some verification that the pledge and owner
        are in communication but is still dependent on
        analysis of the logs to detect unexpected events.";
    }
    enum agent-proximity {
      value 3;
      description
        "Indicates that the voucher has been issued
        after the MASA has verified a statement that
        a registrar agent has made contact with the device.
        This type of voucher is weaker than straight
        proximity, but stronger than logged.";
    }
  }
}
leaf serial-number {
  type string;
  mandatory true;
}

```

```

description
    "The serial-number of the hardware.  When processing a
    voucher, a pledge MUST ensure that its serial-number
    matches this value.  If no match occurs, then the
    pledge MUST NOT process this voucher.";
}
leaf idevid-issuer {
    type binary;
    description
        "The Authority Key Identifier OCTET STRING (as defined in
        Section 4.2.1.1 of RFC 5280) from the pledge's IDevID
        certificate.  Optional since some serial-numbers are
        already unique within the scope of a MASA.
        Inclusion of the statistically unique key identifier
        ensures statistically unique identification of the
        hardware.
        When processing a voucher, a pledge MUST ensure that its
        IDevID Authority Key Identifier matches this value.  If no
        match occurs, then the pledge MUST NOT process this
        voucher.
        When issuing a voucher, the MASA MUST ensure that this
        field is populated for serial-numbers that are not
        otherwise unique within the scope of the MASA.";
}
leaf pinned-domain-cert {
    type binary;
    mandatory false;
    description
        "An X.509 v3 certificate structure, as specified by
        RFC 5280, using Distinguished Encoding Rules (DER)
        encoding, as defined in ITU-T X.690.

        This certificate is used by a pledge to trust a Public Key
        Infrastructure in order to verify a domain certificate
        supplied to the pledge separately by the bootstrapping
        protocol.  The domain certificate MUST have this
        certificate somewhere in its chain of certificates.
        This certificate MAY be an end-entity certificate,
        including a self-signed entity.";
    reference
        "RFC 5280:
        Internet X.509 Public Key Infrastructure Certificate
        and Certificate Revocation List (CRL) Profile.
        ITU-T X.690:
        Information technology - ASN.1 encoding rules:
        Specification of Basic Encoding Rules (BER),
        Canonical Encoding Rules (CER) and Distinguished
        Encoding Rules (DER).";
}

```

```

leaf domain-cert-revocation-checks {
  type boolean;
  description
    "A processing instruction to the pledge that it MUST (true)
    or MUST NOT (false) verify the revocation status for the
    pinned domain certificate. If this field is not set, then
    normal PKIX behavior applies to validation of the domain
    certificate.";
}
leaf nonce {
  type binary {
    length "8..32";
  }
  must 'not(..expires-on)';
  description
    "A value that can be used by a pledge in some bootstrapping
    protocols to enable anti-replay protection. This node is
    optional because it is not used by all bootstrapping
    protocols.

    When present, the pledge MUST compare the provided nonce
    value with another value that the pledge randomly
    generated and sent to a bootstrap server in an earlier
    bootstrapping message. If the value is present, but
    the values do not match, then the pledge MUST NOT process
    this voucher.";
}
leaf pinned-domain-pubk {
  type binary;
  description
    "The pinned-domain-pubk may replace the
    pinned-domain-cert in constrained uses of
    the voucher. The pinned-domain-pubk
    is the Raw Public Key of the Registrar.
    This field is encoded as a Subject Public Key Info block
    as specified in RFC7250, in section 3.
    The ECDSA algorithm MUST be supported.
    The EdDSA algorithm as specified in
    draft-ietf-tls-rfc4492bis-17 SHOULD be supported.
    Support for the DSA algorithm is not recommended.
    Support for the RSA algorithm is a MAY.";
}
leaf pinned-domain-pubk-sha256 {
  type binary;
  description
    "The pinned-domain-pubk-sha256 is a second
    alternative to pinned-domain-cert. In many cases the
    public key of the domain has already been transmitted
    during the key agreement process, and it is wasteful

```


to transmit the public key another two times.
The use of a hash of public key info, at 32-bytes for sha256 is a significant savings compared to an RSA public key, but is only a minor savings compared to a 256-bit ECDSA public-key.
Algorithm agility is provided by extensions to this specification which can define a new leaf for another hash type.";

```
    }  
    leaf last-renewal-date {  
        type yang:date-and-time;  
        must '../expires-on';  
        description  
            "The date that the MASA projects to be the last date it  
            will renew a voucher on. This field is merely  
            informative; it is not processed by pledges.  
  
            Circumstances may occur after a voucher is generated that  
            may alter a voucher's validity period. For instance,  
            a vendor may associate validity periods with support  
            contracts, which may be terminated or extended  
            over time."  
    }  
} // end voucher  
} // end voucher-grouping  
  
}
```

<CODE ENDS>

6.4. ietf-voucher SID values

[RFC9148] explains how to serialize YANG into CBOR, and for this a series of SID values are required. While [I-D.ietf-core-sid] defines the management process for these values, due to the immaturity of the tooling around this YANG-SID mechanisms, the following values are considered normative. It is believed, however, that they will not change.

SID Assigned to

```
-----
2451 data /ietf-voucher:voucher/voucher
2452 data /ietf-voucher:voucher/voucher/assertion
2453 data /ietf-voucher:voucher/voucher/created-on
2454 data .../domain-cert-revocation-checks
2455 data /ietf-voucher:voucher/voucher/expires-on
2456 data /ietf-voucher:voucher/voucher/idevid-issuer
2457 data /ietf-voucher:voucher/voucher/last-renewal-date
2458 data /ietf-voucher:voucher/voucher/nonce
2459 data /ietf-voucher:voucher/voucher/pinned-domain-cert
2460 data /ietf-voucher:voucher/voucher/pinned-domain-pubk
2461 data .../pinned-domain-pubk-sha256
2462 data /ietf-voucher:voucher/voucher/serial-number
```

WARNING, obsolete definitions

The "assertion" attribute is an enumerated type [RFC8366], and the current PYANG tooling does not document the valid values for this attribute. In the JSON serialization, the literal strings from the enumerated types are used so there is no ambiguity. In the CBOR serialization, a small integer is used. This following values are documented here, but the YANG module should be considered authoritative. No IANA registry is provided or necessary because the YANG module provides for extensions.

Integer	Assertion Type
0	verified
1	logged
2	proximity
3	agent-proximity

Table 2: CBOR integers for
the "assertion" attribute
enum

6.5. CMS Format Voucher Artifact

The IETF evolution of PKCS#7 is CMS [[RFC5652](#)]. A CMS-signed voucher, the default type, contains a ContentInfo structure with the voucher content. An eContentType of 40 indicates that the content is a JSON-encoded voucher.

The signing structure is a CMS SignedData structure, as specified by Section 5.1 of [[RFC5652](#)], encoded using ASN.1 Distinguished Encoding Rules (DER), as specified in ITU-T X.690 [[ITU-T.X690.2015](#)].

To facilitate interoperability, [Section 10.3](#) in this document registers the media type "application/voucher-cms+json" and the filename extension ".vcj".

The CMS structure **MUST** contain a 'signerInfo' structure, as described in Section 5.1 of [[RFC5652](#)], containing the signature generated over the content using a private key trusted by the recipient. Normally, the recipient is the pledge and the signer is the MASA. Another possible use could be as a "signed voucher request" format originating from the pledge or registrar toward the MASA. Within this document, the signer is assumed to be the MASA.

Note that Section 5.1 of [[RFC5652](#)] includes a discussion about how to validate a CMS object, which is really a PKCS7 object (cmsVersion=1). Intermediate systems (such the Bootstrapping Remote Secure Key Infrastructures [[BRSKI](#)] registrar) that might need to evaluate the voucher in flight **MUST** be prepared for such an older format. No signaling is necessary, as the manufacturer knows the capabilities of the pledge and will use an appropriate format voucher for each pledge.

The CMS structure **SHOULD** also contain all of the certificates leading up to and including the signer's trust anchor certificate known to the recipient. The inclusion of the trust anchor is unusual in many applications, but third parties cannot accurately audit the transaction without it.

The CMS structure **MAY** also contain revocation objects for any intermediate certificate authorities (CAs) between the voucher issuer and the trust anchor known to the recipient. However, the use of CRLs and other validity mechanisms is discouraged, as the pledge is unlikely to be able to perform online checks and is unlikely to have a trusted clock source. As described below, the use of short-lived vouchers and/or a pledge-provided nonce provides a freshness guarantee.

7. Voucher Request Artifact

[[BRSKI](#)], [Section 3](#) defined a Voucher-Request Artifact as an augmented artifact from the Voucher Artifact originally defined in [[RFC8366](#)]. That definition has been moved to this document, and translated from YANG-DATA [[RFC8040](#)] to the SX:STRUCTURE extension [[RFC8971](#)].

7.1. Tree Diagram

The following tree diagram illustrates a high-level view of a voucher request document. The notation used in this diagram is described in [[RFC8340](#)]. Each node in the diagram is fully described by the YANG module in [Section 7.2](#).

module: ietf-voucher-request

```
structure voucher:
  +-- voucher
    +-- created-on?
      |      yang:date-and-time
    +-- expires-on?
      |      yang:date-and-time
    +-- assertion?
                                     enumeration
    +-- serial-number
                                     string
    +-- idevid-issuer?
                                     binary
    +-- pinned-domain-cert?
                                     binary
    +-- domain-cert-revocation-checks?
                                     boolean
    +-- nonce?
                                     binary
    +-- pinned-domain-pubk?
                                     binary
    +-- pinned-domain-pubk-sha256?
                                     binary
    +-- last-renewal-date?
      |      yang:date-and-time
    +-- prior-signed-voucher-request?
                                     binary
    +-- proximity-registrar-cert?
                                     binary
    +-- proximity-registrar-pubk?
                                     binary
    +-- proximity-registrar-pubk-sha256?
                                     binary
    +-- agent-signed-data?
                                     binary
    +-- agent-provided-proximity-registrar-cert?
                                     binary
    +-- agent-sign-cert?
                                     binary
```

7.2. "ietf-voucher-request" Module

The ietf-voucher-request YANG module is derived from the ietf-voucher module.

<CODE BEGINS>

===== NOTE: '\\' line wrapping per RFC 8792 =====

```
module ietf-voucher-request {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-voucher-request";
  prefix vcr;

  import ietf-yang-structure-ext {
    prefix sx;
  }
  import ietf-voucher {
    prefix vch;
    description
      "This module defines the format for a voucher,
       which is produced by a pledge's manufacturer or
       delegate (MASA) to securely assign a pledge to
       an 'owner', so that the pledge may establish a secure
       connection to the owner's network infrastructure";
    reference
      "RFC 8366: Voucher Artifact for
       Bootstrapping Protocols";
  }

  organization
    "IETF ANIMA Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/anima/>
     WG List:  <mailto:anima@ietf.org>
     Author:   Kent Watsen
               <mailto:kent+ietf@watsen.net>
     Author:   Michael H. Behringer
               <mailto:Michael.H.Behringer@gmail.com>
     Author:   Toerless Eckert
               <mailto:tte+ietf@cs.fau.de>
     Author:   Max Pritikin
               <mailto:pritikin@cisco.com>
     Author:   Michael Richardson
               <mailto:mcr+ietf@sandelman.ca>";
  description
    "This module defines the format for a voucher request.
     It is a superset of the voucher itself.
     It provides content to the MASA for consideration
     during a voucher request.

     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 (RFC 2119) (RFC 8174) when, and only when,
```

they appear in all capitals, as shown here.

Copyright (c) 2019 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the \

RFC

itself for full legal notices.";

```
revision 2023-01-10 {
  description
    "Initial version";
  reference
    "RFC XXXX: Bootstrapping Remote Secure Key Infrastructure";
}
```

```
// Top-level statement
sx:structure voucher {
  uses voucher-request-grouping;
}
```

```
// Grouping defined for future usage
```

```
grouping voucher-request-grouping {
  description
    "Grouping to allow reuse/extensions in future work.";
  uses vch:voucher-artifact-grouping {
    refine "voucher/created-on" {
      mandatory false;
    }
    refine "voucher/pinned-domain-cert" {
      mandatory false;
      description
        "A pinned-domain-cert field
        is not valid in a voucher request, and
        any occurrence MUST be ignored";
    }
    refine "voucher/last-renewal-date" {
      description
        "A last-renewal-date field
        is not valid in a voucher request, and
        any occurrence MUST be ignored";
    }
  }
}
```

```

}
refine "voucher/domain-cert-revocation-checks" {
  description
    "The domain-cert-revocation-checks field
    is not valid in a voucher request, and
    any occurrence MUST be ignored";
}
refine "voucher/assertion" {
  mandatory false;
  description
    "Any assertion included in registrar voucher
    requests SHOULD be ignored by the MASA.";
}
augment "voucher" {
  description
    "Adds leaf nodes appropriate for requesting vouchers.";
  leaf prior-signed-voucher-request {
    type binary;
    description
      "If it is necessary to change a voucher, or re-sign and
      forward a voucher that was previously provided along a
      protocol path, then the previously signed voucher SHOULD
      be included in this field.

      For example, a pledge might sign a voucher request
      with a proximity-registrar-cert, and the registrar
      then includes it as the prior-signed-voucher-request
      field. This is a simple mechanism for a chain of
      trusted parties to change a voucher request, while
      maintaining the prior signature information.

      The Registrar and MASA MAY examine the prior signed
      voucher information for the
      purposes of policy decisions. For example this
      information could be useful to a MASA to determine
      that both pledge and registrar agree on proximity
      assertions. The MASA SHOULD remove all
      prior-signed-voucher-request information when
      signing a voucher for imprinting so as to minimize
      the final voucher size.";
  }
  leaf proximity-registrar-cert {
    type binary;
    description
      "An X.509 v3 certificate structure as specified by
      RFC 5280, Section 4 encoded using the ASN.1
      distinguished encoding rules (DER), as specified
      in [ITU.X690.1994].

```

```

    The first certificate in the Registrar TLS server
    certificate_list sequence (the end-entity TLS
    certificate, see [RFC8446]) presented by the Registrar
    to the Pledge.
    This MUST be populated in a Pledge's voucher request
    when a proximity assertion is requested.";
}
leaf proximity-registrar-pubk {
    type binary;
    description
        "The proximity-registrar-pubk replaces
        the proximity-registrar-cert in constrained uses of
        the voucher-request.
        The proximity-registrar-pubk is the
        Raw Public Key of the Registrar. This field is encoded
        as specified in RFC7250, section 3.
        The ECDSA algorithm MUST be supported.
        The EdDSA algorithm as specified in
        draft-ietf-tls-rfc4492bis-17 SHOULD be supported.
        Support for the DSA algorithm is not recommended.
        Support for the RSA algorithm is a MAY, but due to
        size is discouraged.";
}
leaf proximity-registrar-pubk-sha256 {
    type binary;
    description
        "The proximity-registrar-pubk-sha256
        is an alternative to both
        proximity-registrar-pubk and pinned-domain-cert.
        In many cases the public key of the domain has already
        been transmitted during the key agreement protocol,
        and it is wasteful to transmit the public key another
        two times.
        The use of a hash of public key info, at 32-bytes for
        sha256 is a significant savings compared to an RSA
        public key, but is only a minor savings compared to
        a 256-bit ECDSA public-key.
        Algorithm agility is provided by extensions to this
        specification which may define a new leaf for another
        hash type.";
}
leaf agent-signed-data {
    type binary;
    description
        "The agent-signed-data field contains a JOSE [RFC7515]
        object provided by the Registrar-Agent to the Pledge.

        This artifact is signed by the Registrar-Agent
        and contains a copy of the pledge's serial-number.";
}

```



```

}
leaf agent-provided-proximity-registrar-cert {
    type binary;
    description
        "An X.509 v3 certificate structure, as specified by
        RFC 5280, Section 4, encoded using the ASN.1
        distinguished encoding rules (DER), as specified
        in ITU X.690.
        The first certificate in the registrar TLS server
        certificate_list sequence (the end-entity TLS
        certificate; see RFC 8446) presented by the
        registrar to the registrar-agent and provided to
        the pledge.
        This MUST be populated in a pledge's voucher-request
        when an agent-proximity assertion is requested.";
    reference
        "ITU X.690: Information Technology - ASN.1 encoding
        rules: Specification of Basic Encoding Rules (BER),
        Canonical Encoding Rules (CER) and Distinguished
        Encoding Rules (DER)
        RFC 5280: Internet X.509 Public Key Infrastructure
        Certificate and Certificate Revocation List (CRL)
        Profile
        RFC 8446: The Transport Layer Security (TLS)
        Protocol Version 1.3";
}
leaf agent-sign-cert {
    type binary;
    description
        "An X.509 v3 certificate structure, as specified by
        RFC 5280, Section 4, encoded using the ASN.1
        distinguished encoding rules (DER), as specified
        in ITU X.690.
        This certificate can be used by the pledge,
        the registrar, and the MASA to verify the signature
        of agent-signed-data. It is an optional component
        for the pledge-voucher request.
        This MUST be populated in a registrar's
        voucher-request when an agent-proximity assertion
        is requested.";
    reference
        "ITU X.690: Information Technology - ASN.1 encoding
        rules: Specification of Basic Encoding Rules (BER),
        Canonical Encoding Rules (CER) and Distinguished
        Encoding Rules (DER)
        RFC 5280: Internet X.509 Public Key Infrastructure
        Certificate and Certificate Revocation List (CRL)
        Profile";
}

```

```
}  
}  
}  
}
```

<CODE ENDS>

7.3. ietf-voucher-request SID values

[RFC9148] explains how to serialize YANG into CBOR, and for this a series of SID values are required. While [I-D.ietf-core-sid] defines the management process for these values, due to the immaturity of the tooling around this YANG-SID mechanisms, the following values are considered normative. It is believed, however, that they will not change.

SID Assigned to

2501 data /ietf-voucher-request:voucher/voucher
2515 data .../agent-provided-proximity-registrar-cert
2516 data .../agent-sign-cert
2517 data .../agent-signed-data
2502 data /ietf-voucher-request:voucher/voucher/assertion
2503 data /ietf-voucher-request:voucher/voucher/created-on
2504 data .../domain-cert-revocation-checks
2505 data /ietf-voucher-request:voucher/voucher/expires-on
2506 data .../idevid-issuer
2507 data .../last-renewal-date
2508 data /ietf-voucher-request:voucher/voucher/nonce
2509 data .../pinned-domain-cert
2518 data .../pinned-domain-pubk
2519 data .../pinned-domain-pubk-sha256
2510 data .../prior-signed-voucher-request
2511 data .../proximity-registrar-cert
2513 data .../proximity-registrar-pubk
2512 data .../proximity-registrar-pubk-sha256
2514 data .../serial-number

WARNING, obsolete definitions

The "assertion" attribute is an enumerated type, and has values as defined above in [Table 2](#).

8. Design Considerations

8.1. Renewals Instead of Revocations

The lifetimes of vouchers may vary. In some onboarding protocols, the vouchers may be created and consumed immediately, whereas in other onboarding solutions, there may be a significant time delay between when a voucher is created and when it is consumed. In cases when there is a time delay, there is a need for the pledge to ensure that the assertions made when the voucher was created are still valid.

A revocation artifact is generally used to verify the continued validity of an assertion such as a PKIX certificate, web token, or a "voucher". With this approach, a potentially long-lived assertion is paired with a reasonably fresh revocation status check to ensure that the assertion is still valid. However, this approach increases solution complexity, as it introduces the need for additional protocols and code paths to distribute and process the revocations.

Addressing the shortcomings of revocations, this document recommends instead the use of lightweight renewals of short-lived non-revocable vouchers. That is, rather than issue a long-lived voucher, where the 'expires-on' leaf is set to some distant date, the expectation is for the MASA to instead issue a short-lived voucher, where the 'expires-on' leaf is set to a relatively near date, along with a promise (reflected in the 'last-renewal-date' field) to reissue the voucher again when needed. Importantly, while issuing the initial voucher may incur heavyweight verification checks ("Are you who you say you are?" "Does the pledge actually belong to you?"), reissuing the voucher should be a lightweight process, as it ostensibly only updates the voucher's validity period. With this approach, there is only the one artifact, and only one code path is needed to process it; there is no possibility of a pledge choosing to skip the revocation status check because, for instance, the OCSP Responder is not reachable.

While this document recommends issuing short-lived vouchers, the voucher artifact does not restrict the ability to create long-lived voucher, if required; however, no revocation method is described.

Note that a voucher may be signed by a chain of intermediate CAs leading up to the trust anchor certificate known by the pledge. Even though the voucher itself is not revocable, it may still be revoked, per se, if one of the intermediate CA certificates is revoked.

8.2. Voucher Per Pledge

The solution described herein originally enabled a single voucher to apply to many pledges, using lists of regular expressions to represent ranges of serial-numbers. However, it was determined that blocking the renewal of a voucher that applied to many devices would be excessive when only the ownership for a single pledge needed to be blocked. Thus, the voucher format now only supports a single serial-number to be listed.

9. Security Considerations

9.1. Clock Sensitivity

An attacker could use an expired voucher to gain control over a device that has no understanding of time. The device cannot trust

NTP as a time reference, as an attacker could control the NTP stream.

There are three things to defend against this: 1) devices are required to verify that the expires-on field has not yet passed, 2) devices without access to time can use nonces to get ephemeral vouchers, and 3) vouchers without expiration times may be used, which will appear in the audit log, informing the security decision.

This document defines a voucher format that contains time values for expirations, which require an accurate clock in order to be processed correctly. Vendors planning on issuing vouchers with expiration values must ensure that devices have an accurate clock when shipped from manufacturing facilities and take steps to prevent clock tampering. If it is not possible to ensure clock accuracy, then vouchers with expirations should not be issued.

9.2. Protect Voucher PKI in HSM

Pursuant the recommendation made in Section 6.1 for the MASA to be deployed as an online voucher signing service, it is **RECOMMENDED** that the MASA's private key used for signing vouchers is protected by a hardware security module (HSM).

9.3. Test Domain Certificate Validity When Signing

If a domain certificate is compromised, then any outstanding vouchers for that domain could be used by the attacker. The domain administrator is clearly expected to initiate revocation of any domain identity certificates (as is normal in PKI solutions).

Similarly, they are expected to contact the MASA to indicate that an outstanding (presumably short lifetime) voucher should be blocked from automated renewal. Protocols for voucher distribution are **RECOMMENDED** to check for revocation of domain identity certificates before the signing of vouchers.

9.4. YANG Module Security Considerations

The YANG module specified in this document defines the schema for data that is subsequently encapsulated by a CMS signed-data content type, as described in Section 5 of [[RFC5652](#)]. As such, all of the YANG modeled data is protected from modification.

Implementations should be aware that the signed data is only protected from external modification; the data is still visible. This potential disclosure of information doesn't affect security so much as privacy. In particular, adversaries can glean information such as which devices belong to which organizations and which CRL Distribution Point and/or OCSP Responder URLs are accessed to

validate the vouchers. When privacy is important, the CMS signed-data content type **SHOULD** be encrypted, either by conveying it via a mutually authenticated secure transport protocol (e.g., TLS [[RFC5246](#)]) or by encapsulating the signed-data content type with an enveloped-data content type (Section 6 of [[RFC5652](#)]), though details for how to do this are outside the scope of this document.

The use of YANG to define data structures, via the 'yang-data' statement, is relatively new and distinct from the traditional use of YANG to define an API accessed by network management protocols such as NETCONF [[RFC6241](#)] and RESTCONF [[RFC8040](#)]. For this reason, these guidelines do not follow template described by Section 3.7 of [[YANG-GUIDE](#)].

10. IANA Considerations

10.1. The IETF XML Registry

This document registers two URIs in the "IETF XML Registry" [[RFC3688](#)].

IANA has registered the following:

URI: urn:ietf:params:xml:ns:yang:ietf-voucher
Registrant Contact: The ANIMA WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

10.2. The YANG Module Names Registry

This document registers two YANG module in the "YANG Module Names" registry [[RFC6020](#)].

IANA is asked to registrar the following:

name: ietf-voucher
namespace: urn:ietf:params:xml:ns:yang:ietf-voucher
prefix: vch

reference: :RFC 8366

IANA is asked to register a second YANG module as follows:

name: iana-voucher-assertion-type
namespace: urn:ietf:params:xml:ns:yang:iana-voucher-assertion-type
prefix: ianavat
reference: RFC XXXX

10.3. The Media Types Registry

This document requests IANA to update the following "Media Types" entry to point to the RFC number that will be assigned to this document:

Type name: application

Subtype name: voucher-cms+json

Required parameters: none

Optional parameters: none

Encoding considerations: CMS-signed JSON vouchers are ASN.1/DER encoded.

Security considerations: See [Section 9](#)

Interoperability considerations: The format is designed to be broadly interoperable.

Published specification: RFC 8366

Applications that use this media type: ANIMA, 6tisch, and NETCONF zero-touch imprinting systems.

Fragment identifier considerations: none

Additional information:

Deprecated alias names for this type: none

Magic number(s): None

File extension(s): .vcj

Macintosh file type code(s): none

Person and email address to contact for further information:
IETF ANIMA WG

Intended usage: LIMITED

Restrictions on usage: NONE

Author: ANIMA WG

Change controller: IETF

Provisional registration? (standards tree only): NO

10.4. The SMI Security for S/MIME CMS Content Type Registry

This document requests IANA to update this registered OID in the "SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1)" registry to point to the RFC number to be assigned to this document:

Decimal	Description	References
40	id-ct-animaJSONVoucher	RFC 8366

Table 3

11. References

11.1. Normative References

- [BRSKI] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <<https://www.rfc-editor.org/rfc/rfc8995>>.
- [cBRSKI] Richardson, M., Van der Stok, P., Kampanakis, P., and E. Dijk, "Constrained Bootstrapping Remote Secure Key Infrastructure (BRSKI)", Work in Progress, Internet-Draft, draft-ietf-anima-constrained-voucher-19, 2 January 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-anima-constrained-voucher-19>>.
- [I-D.ietf-core-sid] Veillette, M., Pelov, A., Petrov, I., Bormann, C., and M. Richardson, "YANG Schema Item iDentifier (YANG SID)", Work in Progress, Internet-Draft, draft-ietf-core-sid-19, 26 July 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-core-sid-19>>.
- [ITU-T.X690.2015] International Telecommunication Union, "Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, ISO/IEC 8825-1, August 2015, <<https://www.itu.int/rec/T-REC-X.690/>>.
- [jBRSKI] Werner, T. and M. Richardson, "JWS signed Voucher Artifacts for Bootstrapping Protocols", Work in Progress, Internet-Draft, draft-ietf-anima-jws-voucher-05, 24 October 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-anima-jws-voucher-05>>.
- [PRM] Fries, S., Werner, T., Lear, E., and M. Richardson, "BRSKI with Pledge in Responder Mode (BRSKI-PRM)", Work in Progress, Internet-Draft, draft-ietf-anima-brski-prm-06, 11 January 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-anima-brski-prm-06>>.

[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/rfc/rfc2119>>.

[RFC5652]

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

[RFC6020]

Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/rfc/rfc6020>>.

[RFC7950]

Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/rfc/rfc7950>>.

[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/rfc/rfc8174>>.

[RFC8259]

Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/rfc/rfc8259>>.

[RFC8971]

Pallagatti, S., Ed., Mirsky, G., Ed., Paragiri, S., Govindan, V., and M. Mudigonda, "Bidirectional Forwarding Detection (BFD) for Virtual eXtensible Local Area Network (VXLAN)", RFC 8971, DOI 10.17487/RFC8971, December 2020, <<https://www.rfc-editor.org/rfc/rfc8971>>.

[RFC9148]

van der Stok, P., Kampanakis, P., Richardson, M., and S. Raza, "EST-coaps: Enrollment over Secure Transport with the Secure Constrained Application Protocol", RFC 9148, DOI 10.17487/RFC9148, April 2022, <<https://www.rfc-editor.org/rfc/rfc9148>>.

[ZERO-TOUCH]

Watsen, K., Farrer, I., and M. Abrahamsson, "Secure Zero Touch Provisioning (SZTP)", RFC 8572, DOI 10.17487/RFC8572, April 2019, <<https://www.rfc-editor.org/rfc/rfc8572>>.

11.2. Informative References

[CBOR]

Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, December 2020. <<https://www.rfc-editor.org/info/std94>>

[COSE]

Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, August 2022.
Schaad, J., "CBOR Object Signing and Encryption (COSE): Countersignatures", STD 96, RFC 9338, December 2022.
<<https://www.rfc-editor.org/info/std96>>

[I-D.selander-ace-ake-authz]

Selander, G., Mattsson, J. P., Vučinić, M., Richardson, M., and A. Schellenbaum, "Lightweight Authorization for Authenticated Key Exchange.", Work in Progress, Internet-Draft, draft-selander-ace-ake-authz-05, 18 April 2022, <<https://datatracker.ietf.org/doc/html/draft-selander-ace-ake-authz-05>>.

[imprinting] Wikipedia, "Wikipedia article: Imprinting", February 2018, <[https://en.wikipedia.org/w/index.php?title=Imprinting_\(psychology\)&oldid=825757556](https://en.wikipedia.org/w/index.php?title=Imprinting_(psychology)&oldid=825757556)>.

[JWS]

Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <<https://www.rfc-editor.org/rfc/rfc7515>>.

[RFC3688]

Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/rfc/rfc3688>>.

[RFC5246]

Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/rfc/rfc5246>>.

[RFC6125]

Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <<https://www.rfc-editor.org/rfc/rfc6125>>.

[RFC6241]

Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/rfc/rfc6241>>.

[RFC6838]

Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC

6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/rfc/rfc6838>>.

[RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <<https://www.rfc-editor.org/rfc/rfc7435>>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/rfc/rfc8040>>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/rfc/rfc8340>>.

[RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <<https://www.rfc-editor.org/rfc/rfc8366>>.

[SECUREJOIN] Richardson, M., "6tisch Secure Join protocol", Work in Progress, Internet-Draft, draft-ietf-6tisch-dtsecurity-secure-join-01, 25 February 2017, <<https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-dtsecurity-secure-join-01>>.

[Stajano99theresurrecting] Stajano, F. and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks", 1999, <<https://www.cl.cam.ac.uk/research/dtg/www/files/publications/public/files/tr.1999.2.pdf>>.

[YANG-GUIDE] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <<https://www.rfc-editor.org/rfc/rfc8407>>.

[YANGCBOR] Veillette, M., Ed., Petrov, I., Ed., Pelov, A., Bormann, C., and M. Richardson, "Encoding of Data Modeled with YANG in the Concise Binary Object Representation (CBOR)", RFC 9254, DOI 10.17487/RFC9254, July 2022, <<https://www.rfc-editor.org/rfc/rfc9254>>.

Acknowledgements

The authors would like to thank for following for lively discussions on list and in the halls (ordered by last name): William Atwood, Toerless Eckert, and Sheng Jiang.

Russ Housley provided the upgrade from PKCS7 to CMS (RFC 5652) along with the detailed CMS structure diagram.

Authors' Addresses

Kent Watsen
Watsen Networks

Email: kent+ietf@watsen.net

Michael C. Richardson
Sandelman Software

Email: mcr+ietf@sandelman.ca
URI: <http://www.sandelman.ca/>

Max Pritikin
Cisco Systems

Email: pritikin@cisco.com

Toerless Eckert
Futurewei Technologies Inc.
2330 Central Expy
Santa Clara, 95050
United States of America

Email: tte+ietf@cs.fau.de

Qiufang Ma
Huawei
101 Software Avenue, Yuhua District
Nanjing
210012
China

Email: maqiufang1@huawei.com