ANIMA Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2018

K. Watsen
Juniper Networks
M. Richardson
Sandelman Software
M. Pritikin
Cisco Systems
T. Eckert
Huawei

October 25, 2017

Voucher Profile for Bootstrapping Protocols
draft-ietf-anima-voucher-06

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 one artifact format to be a YANG-defined JSON document that has been signed using a CMS structure. Other YANG-derived formats are possible. The voucher artifact is normally generated by the pledge's manufacturer or delegate (i.e. the Manufacturer Authorized Signing Authority).

This document only defines the voucher artifact, leaving it to other documents to describe specialized protocols for accessing it.

Table of Contents

1. Introduction 2
2. Terminology 3
3. Requirements Language 4
4. Survey of Voucher Types 4
5. Voucher artifact 6
5.1. Tree Diagram 7
5.2. Examples 7
5.3. YANG Module 8
5.4. CMS format voucher artifact 13
6. Design Considerations 14
6.1. Renewals instead of Revocations 14
6.2. Voucher Per Pledge 15
7. Security Considerations 15
7.1. Clock Sensitivity 15
7.2. Protect Voucher PKI in HSM 16
7.3. Test Domain Certificate Validity when Signing 16
8. IANA Considerations 16
8.1. The IETF XML Registry 16
8.2. The YANG Module Names Registry 17
8.3. The SMI Security for S/MIME CMS Content Type Registry . 17
9. References 17
9.1. Normative References 17
9.2. Informative References 18
Appendix A. Acknowledgements 19
Authors' Addresses 19

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 or delegate, i.e. the

Manufacturer Authorized Signing Authority (MASA). This artifact is known as the voucher.

The voucher artifact is a JSON [RFC7159] document, conforming to a data model described by YANG [RFC7950], encoded using the rules defined in [RFC7159], and signed using (by default) a CMS structure [RFC5652].

A voucher may be useful in several contexts, but the driving motivation herein is to support secure bootstrapping mechanisms. Assigning ownership is important to bootstrapping mechanisms so that the pledge can authenticate the network that's trying to take control of it.

The lifetimes of vouchers may vary. In some bootstrapping protocols the vouchers may include a nonce restricting them to a single use, whereas in others the vouchers may have an indicated lifetime. In order to support long lifetimes this document recommends using short lifetimes with programmatic renewal, see Section 6.1.

This document only defines the voucher artifact, leaving it to other documents to describe specialized protocols for accessing it. Some bootstrapping protocols using the voucher artifact defined in this draft include: [I-D.ietf-netconf-zerotouch], [I-D.ietf-6tisch-dtsecurity-secure-join], and [I-D.ietf-anima-bootstrapping-keyinfra]).

2. Terminology

This document uses the following terms (sorted by name):

Artifact: The term "artifact" is used throughout to represent the voucher as instantiated in the form of a signed structure.

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." 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. Securely imprinting is a primary focus of this document [imprinting]. The analogy to Lorenz's work was first noted in [Stajano99theresurrecting].

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

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: The Manufacturer Authorized Signing Authority (MASA) service that signs vouchers. In some bootstrapping protocols, the MASA may have Internet presence and be integral to the bootstrapping process, whereas in other protocols the MASA may be an offline service that has no active role in the bootstrapping process. The MAS concept is explained in more detail in [I-D.ietf-anima-bootstrapping-keyinfra]

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. This is 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 bootstrapping 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 might include an indication of the private PKIX (Public Key Infrastructure using X.509) trust anchor used by the Registrar, or an indication of a public PKIX trust anchor and additional CN-ID or DNS-ID information to complete authentication. Symmetric key or other methods are 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 bootstrapping 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:

Voucher Type	Assertion Logged	Assertion Verified	Registrar ID Trust Anchor	Registrar ID CN-ID or DNS-ID	Validity RTC	Validity 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		optional	

NOTE: All voucher types include a 'Pledge ID serial number' (Not shown 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. This 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 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 a ideal sharing of policy decisions and enforcement between the vendor and the owner.

Ownership ID Voucher: An Ownership ID Voucher is 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. 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 instance of the YANG module defined in Section 5.3 that has been, by default, CMS-signed.

This format is described here as a practical basis for some uses (such as in NETCONF), but more to make it clear what vouchers look

like in practice. This description also serves to validate the YANG model.

Future work is expected to define new mappings of the voucher to CBOR (from JSON), and to change the signature container from CMS to JOSE or COSE. XML or ASN.1 formats are also conceivable.

The method of signaling alternative signature methods is out-of-scope for this document. Documents that leverage vouchers can provide this signaling. The signaling could be 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 transfered on a USB key.

5.1. Tree Diagram

The following tree diagram illustrates a high-level view of a voucher document. The notation used in this diagram is described in [I-D.ietf-netmod-yang-tree-diagrams]). Each node in the diagram is fully described by the YANG module in Section 5.3. Please review the YANG module for a detailed description of the voucher format.

```
module: ietf-voucher

  yang-data voucher-artifact:
      +---- 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
         +---- last-renewal-date?               yang:date-and-time
```

5.2. Examples

This section provides voucher examples for illustration purposes. That these examples conform to the encoding rules defined in [RFC7159].

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 later. 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"
  }
}
```

5.3. YANG Module

Following is a YANG [RFC7950] module formally describing the voucher's JSON document structure.

```
<CODE BEGINS> file "ietf-voucher@2017-10-25.yang"
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-restconf {
    prefix rc;
    description
      "This import statement is only present to access
       the yang-data extension defined in RFC 8040.";
    reference "RFC 8040: RESTCONF Protocol";
  }

  organization
   "IETF ANIMA Working Group";

  contact
   "WG Web:   <http://tools.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', 'MAY', and 'OPTIONAL' in
    the module text are to be interpreted as described in RFC 2119.

    Copyright (c) 2017 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 "2017-10-25" {
    description
     "Initial version";
```

```
    reference
     "RFC XXXX: Voucher Profile for Bootstrapping Protocols";
  }

  // Top-level statement
  rc:yang-data voucher-artifact {
    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 (pinned-domain-cert).";

      leaf created-on {
        type yang:date-and-time;
        mandatory true;
        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 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 {
```

```
          description
            "Indicates that the ownership has been positively
             verified by the MASA (e.g., through sales channel
             integration).";
        }
        enum logged {
          description
            "Indicates that this ownership assignment has been
             logged into a database maintained by the MASA, after
             first verifying that there has not been a previous
             claim in the database for the same pledge (voucher
             transparency).";
        }
        enum proximity {
          description
            "Indicates that this assertion is made based on
             the proximity of the signer as determined by
             local network information. This is useful for
             a pledge device to indicate it 'sees' a specific
             registrar on a TLS connection, or for a registrar
             to indicate it 'sees' a pledge.";
        }
      }
      mandatory true;
      description
        "The assertion is a statement from the MASA regarding how
         the owner was verified.   This statement enables pledges
         to support more detailed policy checks.  Pledges MUST
         ensure that the assertion provided is acceptable before
         processing the voucher.";
    }

    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 RFC5280 4.2.1.1 Authority Key Identifier OCTET STRING
         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 true;
        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-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;
        must "../expires-on";
        description
          "A processing instruction to the pledge that it MUST verify
           the revocation status for the domain certificate.  This
           instruction is only available for vouchers that expire. If
           this field is not set, then normal PKIX behaviour 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 values do not match, then the pledge MUST
           NOT process this voucher.";
      }

      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>

5.4. 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 TBD1 indicates 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.

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. A possible other use could be as a "signed voucher request" format originating from 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 BRSKI Registrar) which 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 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 without it third parties can not accurately audit the transaction.

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 pledge provided nonce provides a freshness guarantee.

6. Design Considerations

6.1. Renewals instead of Revocations

The lifetimes of vouchers may vary. In some bootstrapping protocols, the vouchers may be created and consumed immediately whereas, in other bootstrapping 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 short-comings 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 re-issue 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?), re-issuing 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, without any possibility for a pledge to choose 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 a long-lived vouchers, 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.

6.2. 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.

7. Security Considerations

7.1. Clock Sensitivity

An attacker could use an expired voucher to gain control over a device that has no understand of time.

To defend against this there are three things: devices are required to verify that the expires-on field has not yet passed. Devices

without access to time can use nonces to get ephemeral vouchers. Thirdly, 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 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.

7.2. Protect Voucher PKI in HSM

A voucher is signed by a CA, that may itself be signed by a chain of CAs leading to a trust anchor known to a pledge. Revocation checking of the intermediate certificates may be difficult in some scenarios. The voucher format supports the existing PKIX revocation information distribution within the limits of the current PKI technology (a PKCS7 structure can contain revocation objects as well), but pledges MAY accept vouchers without checking X.509 certificate revocation (when 'domain-cert-revocation-checks' is false). Without revocation checking, a compromised MASA keychain could be used to issue vouchers ad infinitum without recourse. For this reason, MASA implementations wanting to support such deployments SHOULD ensure that all the CA private keys used for signing the vouchers are protected by hardware security modules (HSMs).

7.3. 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 any domain identity certificates before automated renewal of vouchers.

8. IANA Considerations

8.1. The IETF XML Registry

This document registers a URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested:

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.

8.2. The YANG Module Names Registry

This document registers a YANG module in the YANG Module Names registry [RFC6020]. Following the format defined in [RFC6020], the the following registration is requested:

```
name:         ietf-voucher
namespace:    urn:ietf:params:xml:ns:yang:ietf-voucher
prefix:       vch
reference:    RFC XXXX
```

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

This document registers an OID in the "SMI Security for S/MIME CMS Content Type" registry (1.2.840.113549.1.9.16.1), with the value:

Decimal	Description	References
TBD1	id-ct-animaJSONVoucher	[ThisRFC]

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 5652, September 2009.

[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/info/rfc6020>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/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/info/rfc8174>.

9.2. Informative References

[I-D.ietf-6tisch-dtsecurity-secure-join]
Richardson, M., "6tisch Secure Join protocol", draft-ietf-6tisch-dtsecurity-secure-join-01 (work in progress), February 2017.

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-08 (work in progress), October 2017.

[I-D.ietf-netconf-zerotouch]
Watsen, K., Abrahamsson, M., and I. Farrer, "Zero Touch Provisioning for NETCONF or RESTCONF based Management", draft-ietf-netconf-zerotouch-19 (work in progress), October 2017.

[I-D.ietf-netmod-yang-tree-diagrams]
Bjorklund, M. and L. Berger, "YANG Tree Diagrams", draft-ietf-netmod-yang-tree-diagrams-02 (work in progress), October 2017.

[imprinting]
Wikipedia, , "Wikipedia article: Imprinting", July 2015, <https://en.wikipedia.org/wiki/Imprinting_(psychology)>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

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

[Stajano99theresurrecting]
Stajano, F. and R. Anderson, "The resurrecting duckling: security issues for ad-hoc wireless networks", 1999, <https://www.cl.cam.ac.uk/~fms27/papers/1999-StajanoAnd-duckling.pdf>.

Appendix A. 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, Sheng Jiang.

Russ Housley provided the upgrade from PKCS7 to CMS(RFC5652), along with the detailed CMS structure diagram.

Authors' Addresses

Kent Watsen
Juniper Networks

EMail: kwatsen@juniper.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
USA

EMail: tte+ietf@cs.fau.de