ANIMA Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 14, 2017

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

March 13, 2017

Voucher Profile for Bootstrapping Protocols
draft-ietf-anima-voucher-01

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".

The voucher artifact is a YANG-defined JSON document that has been signed using a PKCS#7 structure. The voucher artifact is generated by the pledge's manufacture or delegate (i.e. the MASA).

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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 or delegate (i.e. the MASA). This artifact is known as the voucher.

The voucher artifact is a JSON document, conforming to a data model described by YANG [RFC7950], that has been signed using a PKCS#7 structure.

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 be ephemeral, whereas in others the vouchers may be potentially long-lived. In order to support the second category of vouchers, this document recommends using short-life vouchers with programatic renewal, enabling the MASA to communicate the ongoing validity of vouchers.

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

The following terms are defined for clarity:

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].

Pledge: The prospective device attempting to find and join a secure remote key infrastructure. When shipped it only trusts authorized representatives of the manufacturer.

Voucher: A signed statement from the MASA service that indicates to a Pledge the cryptographic identity of the Registrar it should trust. There are different types of vouchers depending on how that trust asserted. This document describes vouchers in detail.

Domain: The set of entities that trust a common key infrastructure trust anchor. This includes the Proxy, Registrar, Domain

Certificate Authority, Management components and any existing entity that is already a member of the domain.

Domain CA: The domain Certification Authority (CA) provides certification functionalities to the domain. At a minimum it provides certification functionalities to a Registrar and stores the trust anchor that defines the domain. Optionally, it certifies all elements.

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". The term JRC is used in common with other bootstrap mechanisms.

MASA Service: A third-party Manufacturer Authorized Signing Authority (MASA) service on the global Internet. The MASA signs vouchers. It also provides a repository for audit log information of privacy protected bootstrapping events. It does not track ownership. It is trusted by the Pledge.

TOFU: Trust on First Use. Used similarly to [RFC7435]. This is where a Pledge device makes no security decisions but rather simply trusts the first Registrar it is contacted by. This is also known as the "resurrecting duckling" model.

3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in the sections below are to be interpreted as described in RFC 2119 [RFC2119].

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 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 Registrar gaining control over the Pledge device. The following combinations are "types" of vouchers:

	Assertion		Registrar ID		Validity	
Voucher Name	Log-ged	Veri-fied	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		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 direct 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 Voucher's 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: An Ownership ID Voucher is named after inclusion of the Pledge's CN-ID or DNS-ID within the voucher. An example Ownership Voucher is defined in [I-D.ietf-netconf-zerotouch]. The MASA service mitigates a MiTM Registrar by identifying the specific Registrar authorized to own the Pledge. [DISCUSS: still needed?]

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

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

The voucher is signed a PKCS#7 SignedData structure, as specified by Section 9.1 of [RFC2315], encoded using ASN.1 distinguished encoding rules (DER), as specified in ITU-T X.690.

The PKCS#7 structure MUST contain JSON-encoded content conforming to the YANG module specified in Section 5.3.

The PKCS#7 structure MUST also contain a 'signerInfo' structure, as described in Section 9.1 of [RFC2315], containing the signature generated over the content using the MASA's private key.

The PKCS#7 structure SHOULD also contain all of the certificates leading up to and including the MASA's trust anchor certificate known to the pledges.

5.1. Tree Diagram

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

```
module: ietf-voucher
  +--ro voucher
     +--ro authority-key-identifier?         binary
     +--ro created-on                        yang:date-and-time
     +--ro expires-on?                       yang:date-and-time
     +--ro assertion                         enumeration
     +--ro device-identifier                 string
     +--ro trusted-ca-certificate            binary
     +--ro domain-certificate-identifier
     |  +--ro subject?   binary
     |  +--ro cn-id?     string
     |  +--ro dns-id?    string
     +--ro assert-certificate-revocations?   boolean
     +--ro nonce?                            binary
     +--ro last-renewal-date?                yang:date-and-time
```

5.2. Examples

This section provides a couple Voucher examples for illustration purposes.

The following example illustrates an ephemeral voucher (uses a nonce) encoded in JSON. As is expected with a dynamically-generated voucher, only a single pledge (device-identifier) is specified. 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": {
       "assertion": "logged",
       "trusted-ca-certificate": "base64-encoded X.509 DER",
       "device-identifier": "JADA123456789",
       "created-on": "2016-10-07T19:31:42Z",
       "nonce": "base64-encoded octet string"
     }
   }
```

The following illustrates a long-lived voucher (no nonce), encoded in XML. This particular voucher applies to more than one pledge (unique-id), which might relate to, for instance, they were all issued as part of the same purchase order. This voucher includes both a trust anchor certificate (trusted-ca-certificate) as well as some additional information (cn-id and dns-id) that can be used to identify a specific domain certificate issued, perhaps indirectly, by the trust anchor CA.

```
   {
     "ietf-voucher:voucher": {
       "assertion": "verified",
       "trusted-ca-certificate": "base64-encoded X.509 DER",
       "domain-certificate-identifier": {
         "subject": "base64-encoded Subject DER"
       },
       "device-identifier": "JADA123456789",
       "created-on": "2016-10-07T19:31:42Z"
     }
   }
```

5.3. YANG Module

```
 <CODE BEGINS> file "ietf-voucher@2017-03-13.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.  The yang-data extension doesn't
          itself have anything to do with RESTCONF, but was placed in the
          that RFC for convenience.  This extension is being tracked to
          be moved to the next version of the YANG modeling language.
          Regardless where or how this extension statement is defined,
          there should not be any impact to a voucher's encoding.";
       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>";

     description
      "This module defines the format for a voucher, which is produced by
       a pledge's manufacturer or delegate (MASA) to securely assign one
       or more pledges to an 'owner', so that the pledges may establish a
       secure connection to the owner's network infrastructure.";

     revision "2017-03-13" {
       description
        "Initial version";
       reference
        "RFC XXXX: Voucher Profile for Bootstrapping Protocols";
     }

     rc:yang-data voucher-artifact {
       uses voucher-grouping;
     }

     grouping voucher-grouping {
       description
         "Grouping only exists for pyang tree output...";

       container voucher {
         config false;
```

```
       description
         "A voucher that can be used to assign one or more
          pledges to an owner.";

       leaf authority-key-identifier {
         type binary;
         description
          "The Subject Key Identifier of the MASA's leaf certificate.
           Enables the pledge a definitively identify the voucher's
           issuer's certificate.  This field is optional as not all
           vouchers will be signed by a private key associated with
           an X.509 certificate.";
       }

       leaf created-on {
         type yang:date-and-time;
         mandatory true;
         description
           "A value indicating the date this voucher was created.  This
            node is optional because its primary purpose is for human
            consumption.  However, when present, pledges that have
            reliable clocks SHOULD ensure that this created-on value
            is not greater than the current time.";
       }

       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 the pledge supports expirations and the expires-on value
            is less then the current time, then the pledge MUST not
            process this voucher.";
       }

       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).";
           }
         }
         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 device-identifier {
         type string;
         mandatory true;
         description
           "A unique identifier (e.g., serial number) within the scope
            of the MASA.

            When processing a vouchers, pledges MUST ensure that their
            unique identifier matches at least one regular expression in
            the list.  If no matching regular expression is found, the
            pledge MUST NOT process this voucher.";
       }

       leaf trusted-ca-certificate {
         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 field is optional because it may not be needed by all
            bootstrapping protocols.

            Note: the expiration date of this certificate effectively
                  imposes an upper limit on the voucher's expiration.";
```

```
         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).";
       }

       // DISCUSS: do we need this anymore, if short-lived vouchers
       // are expected, shouldn't the leaf certificate be pinned, or
       // perhaps just the immediate issuer CA?
       container domain-certificate-identifier {
         must "../trusted-ca-certificate" {
           description
             "A trusted-ca-certificate must be present whenever
              this node is present";
         }
         description
           "This container identifies specific values that a domain
            certificate, provided to the pledge separately by the
            bootstrapping protocol, MUST contain.  This is useful
            when, for instance, the trust anchor is a long-lived
            public CA certificate, while the domain certificate is
            reissued periodically.

            When provided, the pledge MUST perform RFC 6125 style
            validation of the domain certificate against all of
            the provided values.

            This container is optional because it is unneeded when,
            for instance, the trusted CA certificate is owned by the
            domain (i.e.  a private PKI), and hence the trust model
            can be more relaxed.";

         leaf subject {
           type binary;
           description
             "The certificate's entire subject field MUST match
              this value.  This value is the Subject structure, as
              specified by RFC ???? Section ???, encoded using the
              ASN.1 distinguished encoding rules (DER), as specified
              in ITU-T X.690.";
         }
         leaf cn-id {
           type string;
```

```
           description
             "The certificate's subject field's 'common name' value
              MUST match this value.";
         }
         leaf dns-id {
           type string;
           description
             "A subjectAltName entry of type dNSName in the
              certificate MUST match this value.";
         }
       }


       // DISCUSS: does the transition to 'pinning' model mean we can
       // drop this leaf for now? future proofing allows it to be added
       // if needed but its a edge condition?
       //
       // DISCUSS: there must be such future proofing. not clear where
       // to add it in the voucher document. This is probably the most
       // important point of these discusses
       leaf assert-certificate-revocations {
         type boolean;
         must "../expires-on";
         default true;
         description
           "A processing instruction to the device that it should
            verify revocation information for the PKIX certificates
            involved in bootstrapping. This is available only if
            the pledge has a real-time-clock. This is in addition
            to any revocation checks performed by the MASA.";

           // DISCUSS: should this be a boolean or an enum indicating
           // "fail open" vs "fail closed" to make the meaning clearer.
       }

       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 last date that the MASA projects to be the last date it
            will renew a voucher on (assuming the same validity duration
            used in this voucher.  This field is merely infomrative, it
            is not processed by pledges.

            Circustances may occur after when a voucher was generated
            that can 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. Design Considerations

6.1. Renewals instead of Revocations

A revocation artifact is generally used to verify the continued validity of an assertion such as a PKIX certificate, web token, or a "voucher". Conceptually revocation allows for issuance of assertions using long lifetimes and thereby avoiding ongoing protocol operations to renew the assertion. In practice the use of revocation artifacts increases the solution complexity. Rather than a single protocol, or operation, to obtain or renew the assertion the resulting solution instead has two or more protocols: one for assertion maintanence and the other(s) for revocation verification.

The PKIX use of CRLs and OCSP responses provides an illustrative example. Relying parties that verify revocation information must obtain and parse the CRL or OCSP information. Each revocation method has its own validity period that effectively shortens the certificate validity period (since without valid revocation checks the certificate would be rejected). In addition to having multiple revocation protocol options the resulting space is further

complicated by inline distribution of the revocation information. The TLS extension "Certificate Status Request" [RFC6066] for when "constrained clients may wish to use a certificate-status protocol" is an example of this. Including revocation information into Cryptographic Message Syntax [RFC5652] is another example.

If vouchers included revocation similar complexities would propagate to all related voucher distribution and assertion protocols. Instead vouchers do not support revocation. Instead of the asserting party, or relying party, obtaining and distributing revocation information the asserting party MUST obtain an up-to-date valid voucher. The protocol and operations infrastructures for this are expected to be the same as the initial methods used to obtain a voucher in the first place, with one important clarification: the MASA services MUST issue updated validity period vouchers to the same Registrar ID with minimal friction. This is similar to how an OCSP revocation system is always willing to confirm that a certificate is not revoked. There is no requirement implied that vouchers be contiguously renewed. For example if a two-week lifetime voucher is not used before it expires there is no requirement that it be still valid when renewed. The domain MAY renew an expired voucher at any time. The MASA always has authoritative control and MAY reject such renewals (such as when requested by domain owner's to "block" renewals or if the device has been successfully claimed by an alternate domain). Allowing non-contiguous lifetimes significantly reduces the operational load on the domain as it is not required to maintain valid vouchers; only to ensure a valid voucher is available during the time window in which it needs to be used.

[[EDNOTE: It might be worth including an indication of maximum lifetime for which such automated renewal is available. If so the language we'd use would be similar to the RFC5280 statement that certificate validity period is "the time interval during which the CA warrants that it will maintain information about the status of the certificate" only here used to inform the Registrar of "the time interval during which the MASA warrants that it will maintain information about the status of the ownership claim". Such a field would be independent of the actual validity period of the voucher and is not intended for consumption by the Pledge. A suggested name for this field would be "last-renewal-date".]]

The communications to the MASA service regarding claiming and blocking of devices is out of scope of this specification. Similarly if revocation methods had been described, the method of reporting a revocation would have been out-of-scope.

The lifetimes of vouchers may vary. In some bootstrapping protocols the vouchers may be ephemeral, whereas in others the vouchers may be

potentially long-lived. For bootstrapping protocols that support ephemeral vouchers, there is no need to support renewal. For bootstrapping protocols that support long-lived vouchers, final protocol complexity is reduced when short lifetime vouchers are easily renewed rather than layering on additional revocation methods. Manufacturers MAY issue long-lived vouchers to customers if required but no revocation method is described.

6.2. Voucher Per Pledge

The solution 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.

7. Security Considerations

7.1. Clock Sensitivity

This document defines artifacts containing time values for voucher 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

This document favors using voucher-renewals over needing to support voucher-revocations (Section 6.1). However, a voucher may be signed by a chain of intermediate CAs leading to the trust anchor known to a pledge. Revocation checking of these certificates is similarly difficult. The current voucher format supports the existing PKIX revocation information distribution within the limits of the current PKI technology; but pledges MAY accept vouchers without checking X.509 certificate revocation. Without revocation checking, a compromized MASA keychain could be used to issue vouchers ad infinitum without recourse. For this reason, MASA implementations SHOULD ensure that all the CA private keys 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 in normal in PKI solutions). Similarly they are expected to contact the MASA to indicate that an outstanding (presumably short lifetime) voucher 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
```

9. References

9.1. Normative References

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

[RFC2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax Version 1.5", RFC 2315, DOI 10.17487/RFC2315, March 1998, <http://www.rfc-editor.org/info/rfc2315>.

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

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

9.2. Informative References

[I-D.bjorklund-netmod-yang-tree-diagrams]
Bjorklund, M. and L. Berger, "YANG Tree Diagrams", 2017.

[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-04 (work in progress), October 2016.

[I-D.ietf-netconf-zerotouch]
Watsen, K. and M. Abrahamsson, "Zero Touch Provisioning for NETCONF or RESTCONF based Management", draft-ietf-netconf-zerotouch-12 (work in progress), January 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, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <http://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):

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

EMail: tte+anima@cs.fau.de