ANIMA Working Group   K. Watsen
Internet-Draft   Juniper Networks
Intended status: Standards Track   M. Richardson
Expires: December 9, 2017   Sandelman Software
                                                          M. Pritikin
                                                        Cisco Systems
                                                            T. Eckert
                                                               Huawei
                                                         June 7, 2017

              Voucher Profile for Bootstrapping Protocols
                     draft-ietf-anima-voucher-03

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

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on December 9, 2017.

Copyright Notice

Table of Contents

[1](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

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. i 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.

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", "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:

Voucher Name	Assertion		Registrar ID		Validity	
	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		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. 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

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.

The PKCS#7 structure MAY also contain revocation objects for any
intermediate CAs between the voucher-issuer and the trust anchor
known to the pledge.

[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 created-on                          yang:date-and-time
     +--ro expires-on?                         yang:date-and-time
     +--ro assertion                           enumeration
     +--ro serial-number                       string
     +--ro idevid-issuer?                      binary
     +--ro pinned-domain-cert*                 binary
     +--ro domain-cert-revocation-checks?      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": {
    "created-on": "2016-10-07T19:31:42Z",
    "assertion": "logged",
    "serial-number": "JADA123456789",
    "serial-number-issuer": "some binary identifier",
    "domain-cert-trusted-ca": "base64-encoded X.509 DER",
    "domain-cert-identifier": {
      "subject": "base64-encoded Subject DER"
    },
    "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": {
    "created-on": "2016-10-07T19:31:42Z",
    "expires-on": "2016-10-21T19:31:42Z",
    "assertion": "verified",
    "serial-number": "JADA123456789",
    "serial-number-issuer": "some binary identifier",
    "domain-cert-trusted-ca": "base64-encoded X.509 DER",
    "domain-cert-identifier": {
      "subject": "base64-encoded Subject DER"
    },
    "assert-revocations-on-PKIX-certs": "false",
    "last-renewal-date": "2017-10-07T19:31:42Z"
  }
}
```

5.3. YANG Module

```
<CODE BEGINS> file "ietf-voucher@2017-06-07.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>";

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.

  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.";

revision "2017-06-07" {
  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 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 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).";
        }
      }
      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-list pinned-domain-cert {
      type binary;
      min-elements 1;
      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>

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 delay
   between when a voucher is created and when it is consumed. In cases
   when there is a delay, there is a need for the pledge to ensure that
   the assertions made when the voucher was created are still valid when
   it is consumed.

   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, the
   expectation is for the MASA to instead issue a short-lived voucher
   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 validatity 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 ephermal vouchers. Thirdly, vouchers without expiration times may be used, which will appear in the audit log, informing the security decision.

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

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

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-06 (work in progress), May 2017.

   [I-D.ietf-netconf-zerotouch]
              Watsen, K. and M. Abrahamsson, "Zero Touch Provisioning
              for NETCONF or RESTCONF based Management", draft-ietf-
              netconf-zerotouch-13 (work in progress), March 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
   Futurewei Technologies Inc.
   2330 Central Expy
   Santa Clara 95050
   USA

   EMail: tte+ietf@cs.fau.de