6tisch Working Group
Internet-Draft
Intended status: Informational
Expires: May 3, 2018

M. Richardson
Sandelman Software Works
B. Damm
Silver Spring Networks

October 30, 2017

6tisch Zero-Touch Secure Join protocol
draft-ietf-6tisch-dtsecurity-zerotouch-join-01

Abstract

This document describes a zero-touch mechanism to enroll a new device (the "pledge") into a IEEE802.15.4 TSCH network using the 6tisch signaling mechanisms. The resulting device will obtain a domain specific credential that can be used with either 802.15.9 per-host pair keying protocols, or to obtain the network-wide key from a coordinator. The mechanism describe her is an augmentation to the one-touch mechanism described in [I-D.ietf-6tisch-minimal-security].

Status of This Memo

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

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

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

This Internet-Draft will expire on May 3, 2018.

Table of Contents

1. Introduction

Enrollment of new nodes into LLNs present unique challenges. The constrained nodes has no user interfaces, and even if they did, configuring thousands of such nodes manually is undesireable from a

human resources issue, as well as the difficulty in getting consistent results.

This document is about a standard way to introduce new nodes into a 6tisch network that does not involve any direct manipulation of the nodes themselves. This act has been called "zero-touch" provisioning, and it does not occur by chance, but requires coordination between the manufacturer of the node, the service operator running the LLN, and the installers actually taking the devices out of the shipping boxes.

This document is a constrained profile of [I-D.ietf-anima-bootstrapping-keyinfra]. The above document/protocol is referred by by it's acronym: BRSKI. The pronounciation of which is "brew-ski", a common north american slang for beer with a pseudo-polish ending.

This document follows the same structure as it's parent in order to emphasize the similarities, but specializes a number of things to constrained networks of constrained devices. Like ANIMA's BRSKI, the networks which are in scope for this protocol are deployed by a professional operator. The deterministic mechanisms which have been designed into 6tisch have been created to satisfy the operational needs of industrial settings.

This document builds upon the "one-touch" provisioning described in [I-D.ietf-6tisch-minimal-security], reusing the OSCOAP Join Request mechanism when appropriate. In addition, it uses the CoAP adaption of EST defined in [I-D.vanderstok-ace-coap-est] in an identical way.

Otherwise, this document follows BRSKI with the following high-level changes:

- HTTP is replaced with CoAP.
- TLS (HTTPS) is replaced with either DTLS+CoAP, or EDHOC/OSCOAP+CoAP
- the domain-registrar anchor certificate is replaced with a Raw Public Key (RPK) using [RFC7250].
- the PKCS7 signed JSON voucher format is replaced with CWT
- the GRASP discovery mechanism for the Proxy is replaced with an announcement in the Enhanced Beacon [I-D.richardson-6tisch-join-enhanced-beacon]

- the TCP circuit proxy mechanism is not used. The IPIP mechanism if mandatory to implement when deployed with DTLS, while the CoAP based stateless proxy mechanism is used for OSCOAP/EDHOC.

- real time clocks are assumed to be impossible, so expiry dates in ownership vouchers are never used

- nonce-full vouchers are encouraged, but off-line nonce-less operation is also supported

802.1AR Client certificates are retained, but optionally are specified by reference rather than value.

It is expected that the back-end network operator infrastructure would be able to bootstrap ANIMA BRSKI-type devices over ethernet, while also being able bootstrap 6tisch devices over 802.15.4 with few changes.

1.1. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119] and indicate requirement levels for compliant STuPiD implementations.

The reader is expected to be familiar with the terms and concepts defined in [I-D.ietf-6tisch-terminology], [RFC7252], [I-D.ietf-core-object-security], and [I-D.ietf-anima-bootstrapping-keyinfra]. The following terms are imported: drop ship, imprint, enrollment, pledge, join proxy, ownership voucher, join registrar/coordinator. The following terms are repeated here for readability, but this document is not authoritative for their definition:

pledge the prospective device, which has the identity provided to at the factory. Neither the device nor the network knows if the device yet knows if this device belongs with this network.

Joined Node the prospective device, after having completing the join process, often just called a Node.

Join Proxy (JP): a stateless relay that provides connectivity between the pledge and the join registrar/coordinator.

Join Registrar/Coordinator (JRC): central entity responsible for authentication and authorization of joining nodes.

Audit Token A signed token from the manufacturer authorized signing authority indicating that the bootstrapping event has been successfully logged. This has been referred to as an "authorization token" indicating that it authorizes bootstrapping to proceed.

Ownership Voucher A signed voucher from the vendor vouching that a specific domain "owns" the new entity as defined in [I-D.ietf-anima-voucher].

MIC manufacturer installed certificate. An [ieee802-1AR] identity. Not to be confused with a (cryptographic) "Message Integrity Check"

1.2. Scope of solution

The solution described in this document is appropriate to enrolling between hundreds to hundreds of thousands of diverse devices into a network without any prior contact with the devices. The devices could be shipped by the manufacturer directly to the customer site without ever being seen by the operator of the network. As described in BRSKI, in the audit-mode of operation the device will be claimed by the first network that sees it. In the tracked owner mode of operation, sales channel integration provides a strong connection that the operator of the network is the legitimate owner of the device.

BRSKI describes a more general, more flexible approach for bootstrapping devices into an ISP or Enterprise network.

[I-D.ietf-6tisch-minimal-security] provides an extremely streamlined approach to enrolling from hundreds to thousands of devices into a network, provided that a unique secret key can be installed in each device.

1.3. Leveraging the new key infrastructure / next steps

In constrained networks, it is unlikely that an ACP be formed. This document does not preclude such a thing, but it is not mandated.

The resulting secure channel MAY be used just to distribute network-wide keys using a protocol such as [I-D.ietf-6tisch-minimal-security]. (XXX - do we need to signal this somehow?)

The resulting secure channel MAY be instead used to do an enrollment of an LDevID as in BRSKI, but the resulting certificate is used to do per-pair keying such as described by {{ieee802159}.

1.3.1. Key Distribution Process

In addition to being used for the initial enrollment process, the secure channel may be kept open (and reversed) to use for network rekeying. Such a process is out of scope of this document, please see future work such as [I-D.richardson-6tisch-minimal-rekey].

2. Architectural Overview

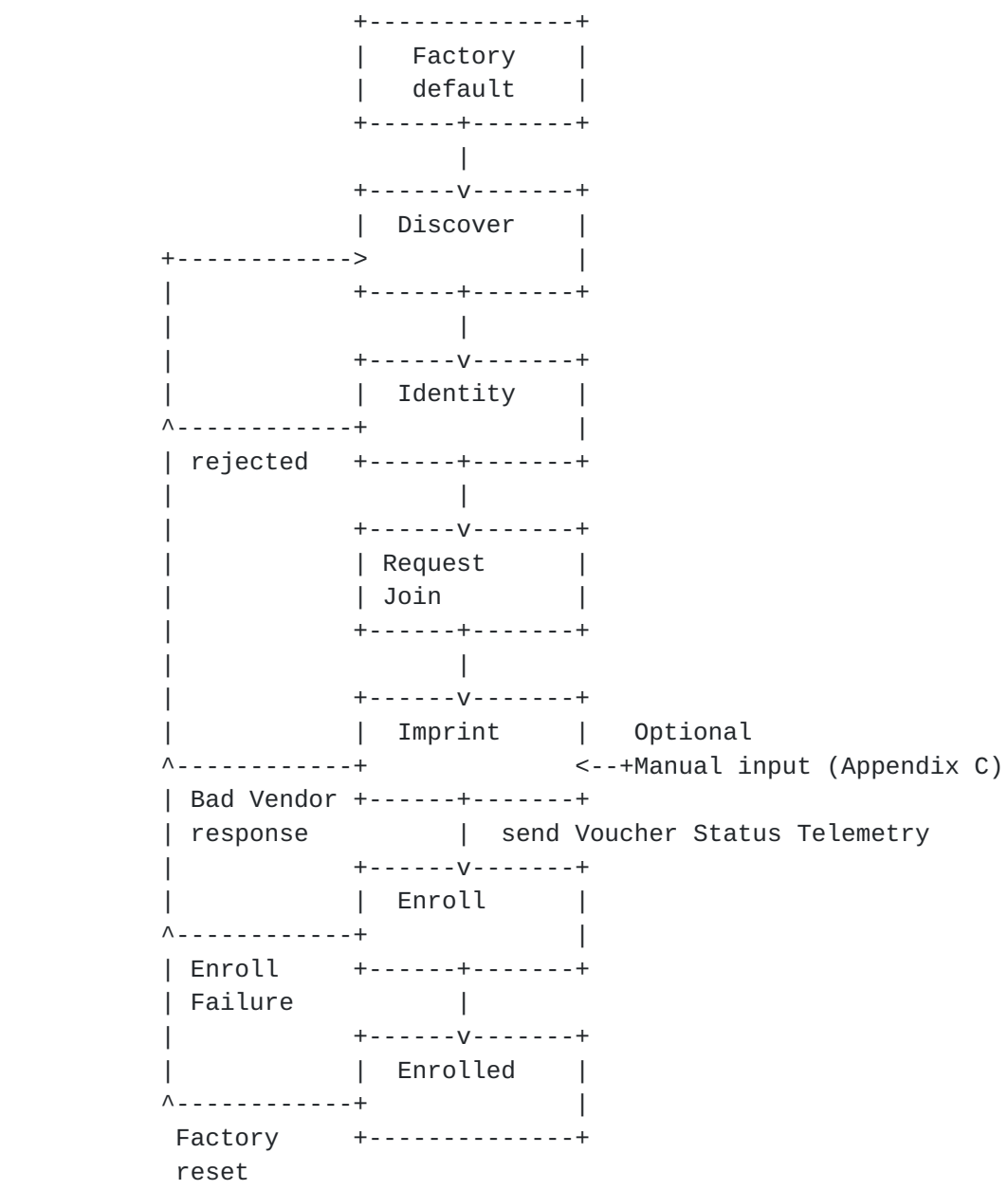
Section 2 of BRSKI has a diagram with all of the components shown together. There are no significant changes to the diagram.

The use of a circuit proxy is not mandated. Instead the IPIP mechanism described in appendix C ("IPIP Join Proxy mechanism") SHOULD be be used instead as it supports both DTLS, EDHOC and OSCOAP protocols.

The CoAP proxy mechanism MAY be implemented instead: the decision depends upon the capabilities of the Registrar and the proxy. A mechanism is included for the Registrar to announce it's capabilities (XXX)

2.1. Behaviour of a Pledge

The pledge goes through a series of steps which are outlined here at a high level.



State descriptions for the pledge are as follows:

1. Discover a communication channel to a Registrar. This is done by listening for beacons as described by [I-D.richardson-anima-6join-discovery]

2. Identify itself. This is done by presenting an X.509 IDevID credential to the discovered Registrar (via the Proxy) in a DTLS or EDHOC handshake. (The Registrar credentials are only provisionally accepted at this time).

The registrar identifies itself using a raw public key, while the the pledge identifies itself to the registrar using an IDevID credential.

3. Requests to Join the discovered Registrar. A unique nonce can be included ensuring that any responses can be associated with this particular bootstrapping attempt.

4. Imprint on the Registrar. This requires verification of the vendor service (MASA) provided voucher. A voucher contains sufficient information for the Pledge to complete authentication of a Registrar. The voucher is signed by the vendor (MASA) using a raw public key, previously installed into the pledge at manufacturing time.

5. Optionally Enroll. By accepting the domain specific information from a Registrar, and by obtaining a domain certificate from a Registrar using a standard enrollment protocol, e.g. Enrollment over Secure Transport (EST) [RFC7030].

6. The Pledge is now a member of, and can be managed by, the domain and will only repeat the discovery aspects of bootstrapping if it is returned to factory default settings.

2.2. Secure Imprinting using Vouchers

As in BRSKI, the format and cryptographic mechansim of vouchers is described in detail in [I-D.ietf-anima-voucher]. As described in section YYY, the physical format for vouchers in this document differs from that of BRSKI, in that it uses [I-D.ietf-ace-cbor-web-token] to encode the voucher and to sign it.

2.3. Initial Device Identifier

The essential component of the zero-touch operation is that the pledge is provisioned with an 802.1AR (PKIX) certificate installed during the manufacturing process.

It is expected that constrained devices will use a signature algorithm corresponding to the hardware acceleration that they have, if they have any. The anticipated algorithms are the ECDSA P-256 (secp256p1) as SHOULD-, while newer devices SHOLD+ begin to appear using EdDSA curves using the 25519 curves. (EDNOTE details here)

There are a number of simplications detailed later on in this document designed to eliminate the need for an ASN.1 parser in the pledge.

The pledge should consider it's 802.1AR certificate to be an opaque blob of bytes, to be inserted into protocols at appropriate places. The pledge SHOULD have access to it's public and private keys in the most useable native format for computation.

The pledge MUST have the public key of the MASA built in a manufacturer time. This is a seemingly identical requirement as for BRSKI, but rather than being an abstract trust anchor that can be augmented with a certificate chain, the pledge MUST be provided with the Raw Public Key that the MASA will use to sign vouchers for that pledge.

There are a number of security concerns with use of a single MASA signing key, and section Section 9.1 addresses some of them with some operational suggestions.

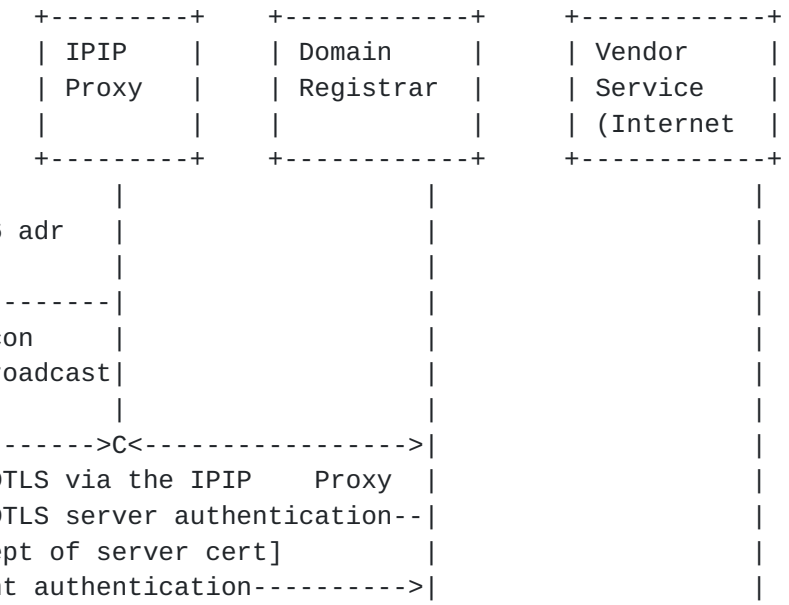
BRSKI places some clear requirements upon the contents of the IDevID, but leaves the exact origin of the voucher serial-number open. This document restricts the process to being the hwSerialNum OCTET STRING. As CWT can handle binary formats, no base64 encoding is necessary.

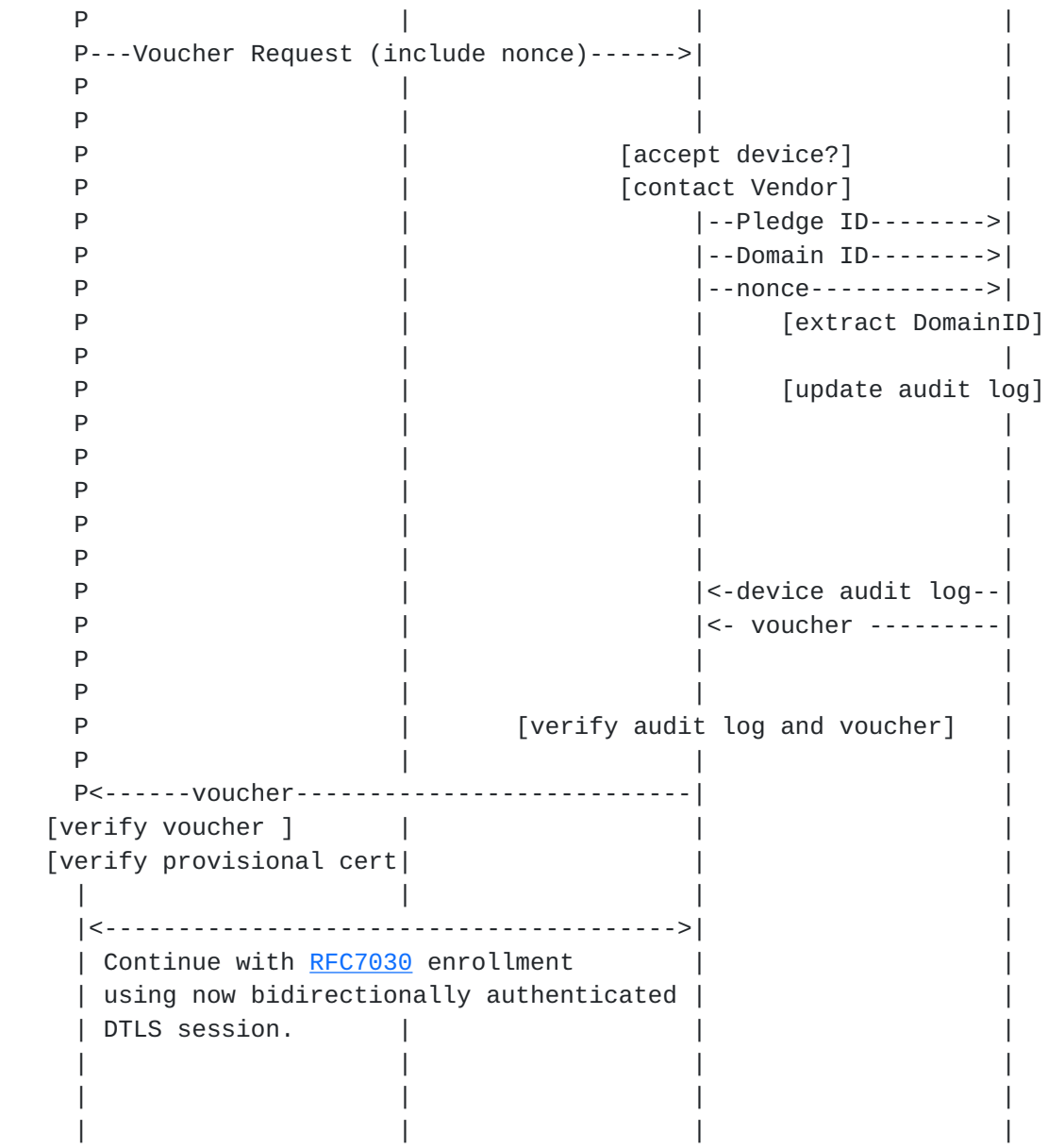
The use of the MASA-URL extension is encouraged if the certificate is sent at all.

EDNOTE: here belongs text about sending only a reference to the IDevID rather than the entire certificate

2.4. Protocol Flow

The diagram from BRSKI is reproduced with some edits:





Noteable changes are:

1. no IPv4 support/options.

2. no mDNS steps, 6tisch only uses Enhanced Beacon

3. nonce-full option is always recommended

2.4.1. Architectural component: Pledge

The Pledge is the device which is attempting to join. Until the pledge completes the enrollment process, it has network connectivity only to the Proxy.

2.4.2. Architectural component: Stateless IPIP Proxy

The stateless CoAP or DTLS Proxy provides CoAP or DTLS connectivity (respectively) between the pledge and the registrar. The stateless CoAP proxy mechanism is described in [I-D.ietf-6tisch-minimal-security] section 5.1. The stateless DTLS mechanism is not yet described (XXX).

2.4.3. Architectural component: Domain Registrar

The Domain Registrar (having the formal name Join Registrar/ Coordinator (JRC)), operates as a CMC Registrar, terminating the EST and BRSKI connections. The Registrar is manually configured or distributed with a list of trust anchors necessary to authenticate any Pledge device expected on the network. The Registrar communicates with the Vendor supplied MASA to establish ownership.

The JRC is typically located on the 6LBR/DODAG root, but it may be located elsewhere, provided IP level connectivity can be established. The 6LBR may also provide a proxy or relay function to connect to the actual registrar in addition to the IPIP proxy described above. The existence of such an additional proxy is a private matter, and this documents assumes without loss of generality that the registrar is co-located with the 6LBR.

2.4.4. Architectural component: Vendor Service

The Vendor Service provides two logically seperate functions: the Manufacturer Authorized Signing Authority (MASA), and an ownership tracking/auditing function. This function is identical to that used by BRSKI, except that a different format voucher is used.

2.5. Lack of realtime clock

For the constrained situation it is assumed that devices have no real time clock. These nodes do have access to a monotonically increasing clock that will not go backwards in the form of the Absolute Sequence Number. Synchronization to the ASN is required in order to transmit/ receive data and most nodes will maintain it in hardware.

The heuristic described in BRSKI under this section SHOULD be applied if there are dates in the CWT format voucher.

Voucher requests SHOULD include a nonce. For devices intended for off-line deployment, the vouchers will have been generated in advance and no nonce-ful operation will not be possible.

2.6. Cloud Registrar

In 6tisch, the pledge never has network connectivity until it is enrolled, so no alternate registrar is ever possible.

2.7. Determining the MASA to contact

There are no changes from BRSKI: the IDevID provided by the pledge will contain a MASA URL extension.

3. Voucher-Request artifact

As in BRSKI, a voucher-request artifact is derived from the base voucher definition. The constrained version differs from the non-constrained version in two ways:

1. it does not include the pinned-domain-cert, but rather than pinned-domain-subjet-public-key-info entry. This accomodates the use of a raw public key to identify the registrar.

2. the pledge voucher-request is never signed.

An appendix shows detailed examples of CWT format voucher requests.

3.1. Tree Diagram

```
module: ietf-cwt-voucher-request
  groupings:
  voucher-request-cwt-grouping
      +---- 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
         +---- proximity-registrar-subject-public-key-info?  binary
```

3.2. SID values

SID experimental base 60100 is used for voucher-requests.

dictionary keys are:

60100	ietf-cwt-voucher
60101	assertion
60102	created-on
60103	domain-cert-revocation-checks
60104	expires-on
60105	idevid-issuer
60106	last-renewal-date
60107	nonce
60108	pinned-domain-cert
60109	pinned-domain-subject-public-key-info
60110	prior-signed-voucher
60111	serial-number
60112	proximity-registrar-cert
60113	proximity-registrar-subject-public-key-info
60100	ietf-cwt-voucher-request

3.3. YANG Module

```
/* -*- c -*- */
module ietf-cwt-voucher-request {
  yang-version 1.1;

  namespace
    "urn:ietf:params:xml:ns:yang:ietf-cwt-voucher-request";
  prefix "vcwt";

  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";
  }

  import ietf-voucher {
    prefix "v";
  }

  organization
   "IETF 6tisch Working Group";
```

```
  contact
   "WG Web:   <http://tools.ietf.org/wg/6tisch/>
    WG List:  <mailto:6tisch@ietf.org>
    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.

    This version provides a very restricted subset appropriate
    for very constrained devices.
    In particular, it assumes that nonce-ful operation is
    always required, that expiration dates are rather weak, as no
    clocks can be assumed, and that the Registrar is identified
    by a pinned Raw Public Key.

    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 "YYYY-MM-DD" {
    description
     "Initial version";
    reference
     "RFC XXXX: Voucher Profile for Constrained Devices";
  }

  // Grouping defined for future usage
  grouping voucher-request-cwt-grouping {
    description
      "Grouping to allow reuse/extensions in future work.";

    uses v:voucher-artifact-grouping {
      augment "voucher" {
        description "Base the CWT voucher-request upon the regular one";
        leaf proximity-registrar-subject-public-key-info {
          type binary;
          description
            "The proximity-registrar-subject-public-key-info replaces the
         proximit-registrar-cert in constrained uses of
         the voucher-request.
         The proximity-registrar-subject-public-key-info is the
         Raw Public Key of the Registrar. This field is encoded
         as specified in RFC7250, section 3.
         The ECDSA algorithm MUST be supported.
```

```
          The EdDSA algorithm as specified in
          draft-ietf-tls-rfc4492bis-17 SHOULD be supported.
          Support for the DSA algorithm is not recommended.
          Support for the RSA algorithm is a MAY.";
        }
      }
    }
  }
}
```

This definition, translated via the rules in [I-D.ietf-core-yang-cbor] uses the

4. Proxy details

The role of the Proxy is to facilitate communication. In the constrained situation the proxy needs to be stateless.

4.1. Pledge discovery of Proxy

In BRSKI, the pledge discovers the proxy via use of a GRASP M_FLOOD messages sent by the proxy. In 6tisch-zero-touch, the existence of the proxy is related by the Enhanced Beacon.

4.2. CoAP connection to Registrar

In BRSKI CoAP is future work. This document represents this work.

4.3. HTTPS proxy connection to Registrar

HTTPS connections are not used.

4.4. Proxy discovery of Registrar

In BRSKI, the proxy autonomically discovers the Registrar by listening for GRASP messages. In the constrained network, the proxies are optionally configured with the address of the registrar by the Join Response in [I-D.ietf-6tisch-minimal-security] section XX. The address of the registrar otherwise defaults to be that of the DODAG root.

5. Protocol Details

BRSKI is specified to run over HTTPS. This document respecifies it to run over CoAP with either DTLS or EDHOC-provided OSCOAP security. There is an emerging (hybrid) possibility of DTLS-providing the OSCOAP security, but such a specification does not yet exist.

[I-D.vanderstok-ace-coap-est] specifies that CoAP specifies the use of CoAP Block-Wise Transfer ("Block") [RFC7959] to fragment EST messages at the application layer.

BRSKI introduces the concept of a provisional state for EST. The same situation must also be added to DTLS: a situation where the connection is active but the identity of the Registar has not yet been confirmed. The DTLS MUST validate that the exchange has been signed by the Raw Public Key that is presented by the Server, even though there is as yet no trust in that key. Such a key is often available through APIs that provide for channel binding, such as described in [RFC5056].

As in [I-D.vanderstok-ace-coap-est], support for Observe CoAP options [RFC7641] with BRSKI is not supported in the current BRSKI/EST message flows. Observe options could be used by the server to notify clients about a change in the cacerts or csr attributes (resources) and might be an area of future work.

Redirection as described in [RFC7030] section 3.2.1 is NOT supported.

5.1. BRSKI-EST (D)TLS establishment details

Zerotouch Join does not use TLS. The connection is either CoAP over DTLS, or CoAP with EDHOC security.

5.1.1. BRSKI-EST CoAP/DTLS estasblishment details

The details in the BRSKI document apply directly to use of DTLS.

The registrar SHOULD authenticate itself with a raw public key.

The pledge SHOULD authenticate itself with the built-in IDevID certificate.

5.1.2. BRSKI-EST CoAP/EDHOC estasblishment details

[I-D.ietf-6tisch-minimal-security] section YYY details how to setup EDHOC.

The registrar SHOULD authenticate itself with a raw public key.

The pledge SHOULD authenticate itself with the built-in IDevID certificate.

5.2. Pledge Requests Voucher from the Registrar

The voucher request and response as defined by BRSKI is modified slightly.

In order to simplify the pledge, the use of a certificate (and chain) for the Registrar is not supported. Instead the newly defined pinned-domain-subject-public-key-info must contain the (raw) public key info for the Registrar. It MUST be byte for byte identical to that which is transmitted by the Registrar during the TLS ServerCertificate handshake.

BRSKI permits the voucher request to be signed or unsigned. This document defines the voucher request to be unsigned.

5.3. BRSKI-MASA TLS establishment details

There are no changes. The connection from the Registrar to MASA is still HTTPS.

5.4. Registrar Requests Voucher from MASA

There are no change from BRSKI, as this step is between two non-constrained devices. The format of the voucher is CWT, which implies changes to both the Registrar and the MASA, but semantically the content is the same.

The manufacturer will know what algorithms are supported by the pledge, and will issue a 406 (Conflict) error to the Registrar if the Registar's public key format is not supported by the pledge.

5.5. Voucher Response

The format of the voucher is CWT as described in section YYY.

5.5.1. Completing authentication of Provisional TLS connection

The BRSKI process uses the pinned-domain-cert field of the voucher to validate the registrar's ServerCertificate. In the ZeroTouch case, the voucher will contain a pinned-domain-subject-public-key-info field containing the raw public key of the certificate. It should match, byte-to-byte with the raw public key ServerCertificate.

5.6. Voucher Status Telemetry

The voucher status telemetry report is communicated from the pledge to the registrar over CoAP channel. The shortened URL is as described in table QQQ.

5.7. MASA authorization log Request

There are no changes to the MASA audit log request.

5.7.1. MASA authorization log Response

There are no changes to the MASA audit log response.

5.8. EST Integration for PKI bootstrapping

There.

5.8.1. EST Distribution of CA Certificates

TBD.

5.8.2. EST CSR Attributes

TBD.

5.8.3. EST Client Certificate Request

TBD.

5.8.4. Enrollment Status Telemetry

There are no changes to the status telemetry between Registrar and MASA.

5.8.5. EST over CoAP

This document and [I-D.vanderstok-ace-coap-est] detail how to run EST over CoAP.

6. Reduced security operational modes

TBD

6.1. Trust Model

TBD

6.2. Pledge security reductions

TBD

6.3. Registrar security reductions

TBD

6.4. MASA security reductions

TBD

7. IANA Considerations

XXX

7.1. MIME-Type Registry

XXX

8. IANA Considerations

9. Security Considerations

TBD

9.1. Security of MASA voucher signing key(s)

TBD

10. Privacy Considerations

[I-D.ietf-6lo-privacy-considerations] details a number of privacy considerations important in Resource Constrained nodes. There are two networks and three sets of constrained nodes to consider. They are: 1. the production nodes on the production network. 2. the new pledges, which have yet to enroll, and which are on a join network. 3. the production nodes which are also acting as proxy nodes.

10.1. Privacy Considerations for Production network

The details of this are out of scope for this document.

10.2. Privacy Considerations for New Pledges

New Pledges do not yet receive Router Advertisements with PIO options, and so configure link-local addresses only based upon layer-2 addresses using the normal SLAAC mechanisms described in [RFC4191].

These link-local addresses are visible to any on-link eavesdropper (who is synchronized to the same Join Assistant), so regardless of what is chosen they can be seen. This link-layer traffic is encapsulated by the Join Proxy into IPIP packets and carried to the JRC. The traffic SHOULD never leave the operator's network, will be kept confidential by the layer-2 keys inside the LLN. As no outside traffic can enter the join network, to do any ICMP scanning as described in [I-D.ietf-6lo-privacy-considerations].

The join process described herein requires that some identifier meaningful to the network operator be communicated to the JRC. The join request with this object occurs within a secured CoAP channel, although the link-local address configured by the pledge will be visible in either the CoAP stateless proxy option (section 5.1 of [I-D.ietf-6tisch-minimal-security]), or in the equivalent DTLS stateless proxy option (reference TBD).

This need not be a manufacturer created EUI-64 as assigned by IEEE; it could be another value with higher entropy and less interesting vendor/device information. Regardless of what is chosen, it can be used to track where the device attaches.

For most constrained device, network attachment occurs very infrequently, often only once in their lifetime, so tracking opportunities may be rare. Once connected, the long 8-byte EUI64 layer-2 address is usually replaced with a short JRC assigned 2-byte address.

Additionally, during the enrollment process, a DTLS connection or EDHOC connection will be created. TLS1.3 will keep contents of the certificates transmitted private while TLS 1.2 will not. If the client certificate can be observed, then the device identity will be visible to passive observers in the 802.11AR IDevID certificate that is sent.

Even when TLS 1.3 is used, an active attacker could collect the information by creating a rogue proxy.

The use of a manufacturer assigned EUI64 (whether derived from IEEE assignment or created through another process during manufacturing time) is encouraged.

10.2.1. EUI-64 derived address for join time IID

The IID used in the link-local address used during the join process be a vendor assigned EUI-64. After the join process has concluded, the device SHOULD be assigned a unique randomly generated long address, and a unique short address (not based upon the vendor EUI-

64) for use at link-layer address. At that point, all layer-3 content is encrypted by the layer-2 key.

10.3. Privacy Considerations for Join Proxy

TBD.

11. Acknowledgements

Kristofer Pister helped with many non-IETF references.

12. References

12.1. Normative References

[cullenCiscoPhoneDeploy]
Jennings, C., "Transitive Trust Enrollment for Constrained Devices", 2012, <http://www.lix.polytechnique.fr/hipercom/SmartObjectSecurity/papers/CullenJennings.pdf>.

[I-D.ietf-6lo-privacy-considerations]
Thaler, D., "Privacy Considerations for IPv6 Adaptation Layer Mechanisms", draft-ietf-6lo-privacy-considerations-04 (work in progress), October 2016.

[I-D.ietf-6tisch-minimal]
Vilajosana, X., Pister, K., and T. Watteyne, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-21 (work in progress), February 2017.

[I-D.ietf-6tisch-minimal-security]
Vucinic, M., Simon, J., Pister, K., and M. Richardson, "Minimal Security Framework for 6TiSCH", draft-ietf-6tisch-minimal-security-04 (work in progress), October 2017.

[I-D.ietf-6tisch-terminology]
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-09 (work in progress), June 2017.

[I-D.ietf-ace-cbor-web-token]
Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", draft-ietf-ace-cbor-web-token-09 (work in progress), October 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-anima-grasp]
Bormann, C., Carpenter, B., and B. Liu, "A Generic Autonomic Signaling Protocol (GRASP)", draft-ietf-anima-grasp-15 (work in progress), July 2017.

[I-D.ietf-anima-voucher]
Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "Voucher Profile for Bootstrapping Protocols", draft-ietf-anima-voucher-06 (work in progress), October 2017.

[I-D.ietf-core-comi]
Veillette, M., Stok, P., Pelov, A., and A. Bierman, "CoAP Management Interface", draft-ietf-core-comi-01 (work in progress), July 2017.

[I-D.ietf-core-object-security]
Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", draft-ietf-core-object-security-06 (work in progress), October 2017.

[I-D.ietf-core-yang-cbor]
Veillette, M., Pelov, A., Somaraju, A., Turner, R., and A. Minaburo, "CBOR Encoding of Data Modeled with YANG", draft-ietf-core-yang-cbor-05 (work in progress), August 2017.

[I-D.ietf-netconf-keystore]
Watsen, K., "Keystore Model", draft-ietf-netconf-keystore-02 (work in progress), June 2017.

[I-D.richardson-6tisch-join-enhanced-beacon]
Dujovne, D. and M. Richardson, "IEEE802.15.4 Informational Element encapsulation of 6tisch Join Information", draft-richardson-6tisch-join-enhanced-beacon-02 (work in progress), July 2017.

[I-D.richardson-6tisch-minimal-rekey]
Richardson, M., "Minimal Security rekeying mechanism for 6TiSCH", draft-richardson-6tisch-minimal-rekey-02 (work in progress), August 2017.

[I-D.richardson-anima-6join-discovery]
Richardson, M., "GRASP discovery of Registrar by Join Assistant", draft-richardson-anima-6join-discovery-00 (work in progress), October 2016.

[I-D.selander-ace-cose-ecdhe]
Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", draft-selander-ace-cose-ecdhe-07 (work in progress), July 2017.

[I-D.vanderstok-ace-coap-est]
Kumar, S., Stok, P., Kampanakis, P., Furuhed, M., and S. Raza, "EST over secure CoAP (EST-coaps)", draft-vanderstok-ace-coap-est-02 (work in progress), June 2017.

[iec62591]
IEC, ., "62591:2016 Industrial networks - Wireless communication network and communication profiles - WirelessHART", 2016, <https://webstore.iec.ch/publication/24433>.

[ieee802-1AR]
IEEE Standard, ., "IEEE 802.1AR Secure Device Identifier", 2009, <http://standards.ieee.org/findstds/standard/802.1AR-2009.html>.

[ieee802154]
IEEE Standard, ., "802.15.4-2015 - IEEE Standard for Low-Rate Wireless Personal Area Networks (WPANs)", 2015, <http://standards.ieee.org/findstds/standard/802.15.4-2015.html>.

[ieee802159]
IEEE Standard, ., "802.15.9-2016 - IEEE Approved Draft Recommended Practice for Transport of Key Management Protocol (KMP) Datagrams", 2016, <http://standards.ieee.org/findstds/standard/802.15.9-2016.html>.

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

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/info/rfc7030>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <https://www.rfc-editor.org/info/rfc7217>.

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

[RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, DOI 10.17487/RFC7250, June 2014, <https://www.rfc-editor.org/info/rfc7250>.

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

[RFC7959] Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in the Constrained Application Protocol (CoAP)", RFC 7959, DOI 10.17487/RFC7959, August 2016, <https://www.rfc-editor.org/info/rfc7959>.

12.2. Informative References

[duckling] 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>.

[I-D.ietf-ace-actors]
Gerdes, S., Seitz, L., Selander, G., and C. Bormann, "An architecture for authorization in constrained environments", draft-ietf-ace-actors-05 (work in progress), March 2017.

[I-D.ietf-core-sid]
Veillette, M., Pelov, A., Turner, R., Minaburo, A., and A. Somaraju, "YANG Schema Item iDentifier (SID)", draft-ietf-core-sid-01 (work in progress), May 2017.

[I-D.ietf-roll-useofrplinfo]
Robles, I., Richardson, M., and P. Thubert, "When to use RFC 6553, 6554 and IPv6-in-IPv6", draft-ietf-roll-useofrplinfo-19 (work in progress), October 2017.

[ISA100] "The Technology Behind the ISA100.11a Standard", June 2010, <http://www.isa100wci.org/Documents/PDF/The-Technology-Behind-ISA100-11a-v-3_pptx>.

[PFS] Wikipedia, ., "Forward Secrecy", August 2016, <https://en.wikipedia.org/w/index.php?title=Forward_secrecy&oldid=731318899>.

[pledge] Dictionary.com, ., "Dictionary.com Unabridged", 2015, <http://dictionary.reference.com/browse/pledge>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <https://www.rfc-editor.org/info/rfc4191>.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

[RFC5056] Williams, N., "On the Use of Channel Bindings to Secure Channels", RFC 5056, DOI 10.17487/RFC5056, November 2007, <https://www.rfc-editor.org/info/rfc5056>.

[RFC7554] Watteyne, T., Ed., Palattella, M., and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement", RFC 7554, DOI 10.17487/RFC7554, May 2015, <https://www.rfc-editor.org/info/rfc7554>.

[RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641, DOI 10.17487/RFC7641, September 2015, <https://www.rfc-editor.org/info/rfc7641>.

[RFC7731] Hui, J. and R. Kelsey, "Multicast Protocol for Low-Power and Lossy Networks (MPL)", RFC 7731, DOI 10.17487/RFC7731, February 2016, <https://www.rfc-editor.org/info/rfc7731>.

Appendix A. Extra text

The following text is from previous versions of this document. The document has been re-organized to match the flow of [I-D.ietf-anima-bootstrapping-keyinfra].

A.1. Assumptions

A.1.1. One-Touch Assumptions

This document interacts with the one-touch solution described in [I-D.ietf-6tisch-minimal-security].

A.1.2. Factory provided credentials (if any)

When a manufacturer installed certificate is provided as the IDevID, it SHOULD contain a number of fields. [I-D.ietf-anima-bootstrapping-keyinfra] provides a detailed set of requirements.

A manufacturer unique serial number MUST be provided in the serialNumber SubjectAltName extension, and MAY be repeated in the Common Name. There are no sequential or numeric requirements on the serialNumber, it may be any unique value that the manufacturer wants to use. The serialNumber SHOULD be printed on the packaging and/or on the device in a discrete way so that failures can be physically traced to the relevant device.

A.1.3. Credentials to be introduced

The goal of the bootstrap process is to introduce one or more new locally relevant credentials:

1. a certificate signed by a local certificate authority/registrar. This is the LDevID of [ieee802-1AR].

2. alternatively, a network-wide key to be used to secure L2 traffic.

3. alternatively, a network-wide key to be used to authenticate per-peer keying of L2 traffic using a mechanism such as provided by [ieee802159].

A.2. Network Assumptions

This document is about enrollment of constrained devices [RFC7228] to a constrained network. Constrained networks is such as [ieee802154], and in particular the time-slotted, channel hopping (tsch) mode, feature low bandwidths, and limited opportunities to transmit. A key feature of these networks is that receivers are only listening at certain times.

A.2.1. Security above and below IP

802.15.4 networks have three kinds of layer-2 security:

- a network key that is shared with all nodes and is used for unicast and multicast. The key may be used for privacy, and it may be used in some cases for authentication only (in the case of enhanced beacons).

- a series of network keys that are shared (agreed to) between pairs of nodes (the per-peer key)

- a network key that is shared with all nodes (through a group key management system), and is used for multicast traffic only, while a per-pair key is used for unicast traffic

Setting up the credentials to bootstrap one of these kinds of security, (or directly configuring the key itself for the first case) is required. This is the security below the IP layer.

Security is required above the IP layer: there are three aspects which the credentials in the previous section are to be used.

- to provide for secure connection with a Path Computation Element [RFC4655], or other LLC (see ({RFC7554}} section 3).

- to initiate a connection between a Resource Server (RS) and an application layer Authorization Server (AS and CAS from [I-D.ietf-ace-actors]).

A.2.1.1. Perfect Forward Secrecy

Perfert Forward Secrecy (PFS) is the property of a protocol such that complete knowledge of the crypto state (for instance, via a memory

dump) at time X does not imply that data from a disjoint time Y can also be recovered. ([PFS]).

PFS is important for two reasons: one is that it offers protection against the compromise of a node. It does this by changing the keys in a non-deterministic way. This second property also makes it much easier to remove a node from the network, as any node which has not participated in the key changing process will find itself no longer connected.

A.2.2. Join network assumptions

The network which the new pledge will connect to will have to have the following properties:

- a known PANID. The PANID 0xXXXX where XXXX is the assigned RFC# for this document is suggested.

- a minimal schedule with some Aloha time. This is usually in the same slotframe as the Enhanced Beacon, but a pledge MUST listen for an unencrypted Enhanced Beacon to so that it can synchronize.

A.2.3. Number and cost of round trips

TBD.

A.2.4. Size of packets, number of fragments

TBD

A.3. Target end-state for join process

At the end of the zero-touch join process there will be a symmetric key protected channel between the Join Registrar/Coordinator and the pledge, now known as a Joined Node. This channel may be rekeyed via new exchange of asymmetric exponents (ECDH for instance), authenticated using the domain specific credentials created during the join process.

This channel is in the form of an OSCOAP protected connection with [I-D.ietf-core-comi] encoded objects. This document includes definition of a [I-D.ietf-netconf-keystore] compatible objects for encoding of the relevant [I-D.ietf-anima-bootstrapping-keyinfra] objects.

Appendix B. Join Protocol

The pledge join protocol state machine is described in [I-D.ietf-6tisch-minimal-security], in section XYZ. The pledge recognizes that it is in zero-touch configuration by the following situation:

- no PSK has been configured for the network in which it has joined.
- the pledge has no locally defined certificate (no LDevID), only an IDevID.
- the network asserts an identity that the pledge does not recognize.

All of these conditions MUST be true. If any of these are not true, then the pledge has either been connected to the wrong network, or it has already been bootstrapped into a different network, and it should wait until it finds that network.

The zero-touch process consists of three stages:

1. the key agreement process
2. the provisional enrollment process
3. the key distribution process

B.1. Key Agreement process

The key agreement process is identical to [I-D.ietf-6tisch-minimal-security]. The process uses EDHOC with certificates.

The pledge will have to trust the JRC provisionally, as described in [I-D.ietf-anima-bootstrapping-keyinfra], section 3.1.2, and in section 4.1.1 of [RFC7030].

The JRC will be able to validate the IDevID of the pledge using the manufacturer's CA.

The pledge may not know if it is in a zero-touch or one-touch situation: the pledge may be able to verify the JRC based upon trust anchors that were installed at manufacturing time. In that case, the pledge runs the simplified one-touch process.

The pledge signals in the EDHOC message_2 if it has accepted the JRC certificate. The JRC will in general, not trust the pledge with the

network keys until it has provided the pledge with a voucher. The pledge will notice the absence of the provisioning keys.

XXX - there could be some disconnect here. May need additional signals here.

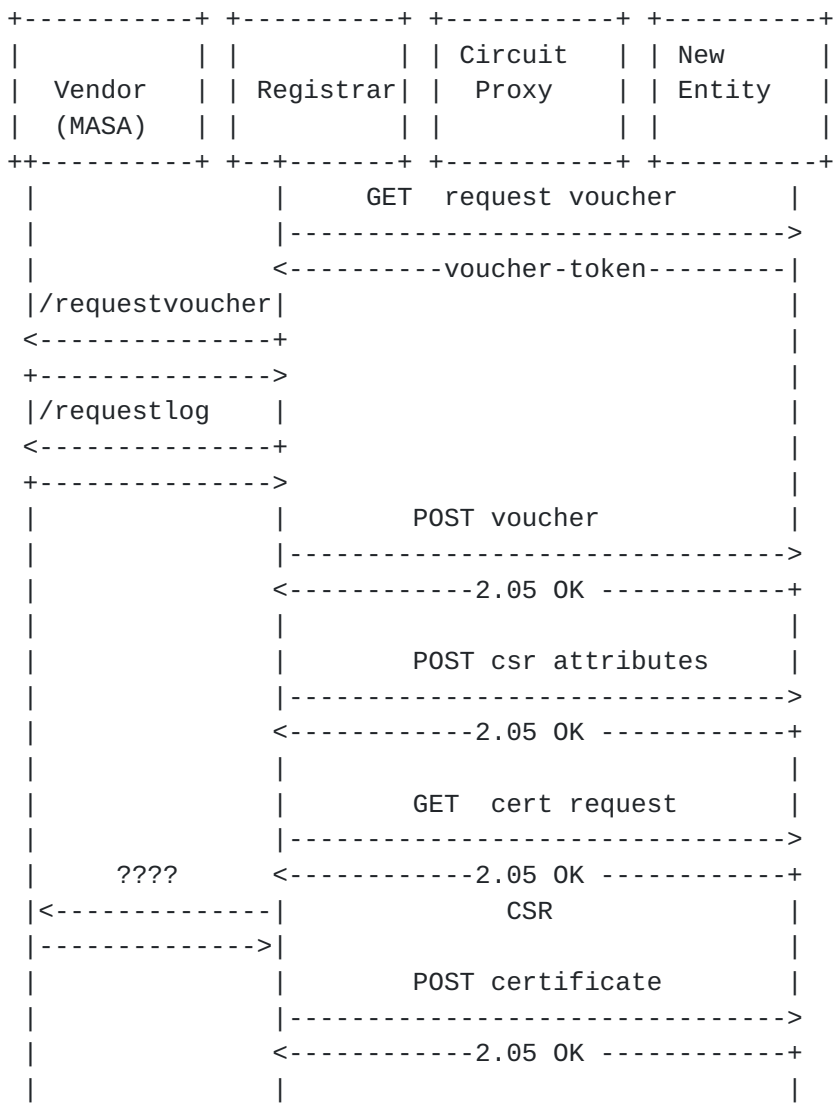
B.2. Provisional Enrollment process

When the pledge determines that it can not verify the certificate of the JRC using built-in trust anchors, then it enters a provisional state. In this state, it keeps the channel created by EDHOC open.

A new EDHOC key derivation is done by the JRC and pledge using a new label, "6tisch-provisional".

The pledge runs as a passive CoMI server, leaving the JRC to drive the enrollment process. The JRC can interrogate the pledge in a variety of fashions as shown below: the process terminates when the JRC provides the pledge with an ownership voucher and the pledge leaves the provisional state.

A typical interaction involves the following requests:



Appendix C. IANA Considerations

This document allocates one value from the subregistry "Address Registration Option Status Values": ND_NS_JOIN_DECLINED Join Assistant, JOIN DECLINED (TBD-AA)

Appendix D. Protocol Definition

D.1. Discovery

Only IPv6 operations using Link-Local addresses are supported. Use of a temporary address is NOT encouraged as the critial resource on the Proxy device is the number of Neighbour Cache Entries that can be used for untrusted pledge entries.

D.1.1. Proxy Discovery Protocol Details

The Proxy is discovered using the enhanced beacon defined in [I-D.richardson-6tisch-join-enhanced-beacon].

D.1.2. Registrar Discovery Protocol Details

The Registrar is not discovered by the Proxy. Any device that is expected to be able to operate as a Registrar MAY be told the address of the Registrar when that device joins the network. The address MAY be included in the [I-D.ietf-6tisch-minimal-security] Join Response. If the address is NOT included, then Proxy may assume that the Registrar can be found at the DODAG root, which is well known in the 6tisch's use of the RPL protocol.

Authors' Addresses

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

Benjamin Damm
Silver Spring Networks

Email: bdamm@ssni.com