6tisch Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2020

M. Richardson
Sandelman Software Works

July 08, 2019

6tisch Zero-Touch Secure Join protocol
draft-ietf-6tisch-dtsecurity-zerotouch-join-04

Abstract

This document describes a Zero-touch Secure Join (ZSJ) 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 here is an augmentation to the one-touch mechanism described in [I-D.ietf-6tisch-minimal-security], and is a profile of the constrained voucher mechanism [I-D.ietf-anima-constrained-voucher].

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.

The mechanism described in [I-D.ietf-anima-bootstrapping-keyinfra] has been adapted in [I-D.ietf-anima-constrained-voucher] to produce a protocol that is suited for constrained devices and constrained networks such as 6tisch. The above document/protocol is referred by by it's acronym: BRSKI and constrained-BRSKI. The pronounciation of which is "brew-ski", a common north american slang for beer with a

pseudo-polish ending. This constrained protocol is called Zero-touch Secure Join.

This document is a profile of [I-D.ietf-anima-constrained-voucher]. It uses COSE signatures of CBOR voucher [RFC8366] artifacts, and it uses [I-D.selander-ace-cose-ecdhe] as a Lightweight authenticated key exchange protocol.

[I-D.ietf-anima-constrained-voucher] has options for CMS signatures of CBOR vouchers, and for using DTLS. The protocol described in this document does not make use those options.

Like [I-D.ietf-anima-bootstrapping-keyinfra], 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 where such an operator exists.

This document builds upon the "one-touch" provisioning described in [I-D.ietf-6tisch-minimal-security], reusing the OSCOAP Join Request mechanism when appropriate, but preceeding it with the EDHOC key agreement protocol.

As a second option, a certificate may be deployed using the constrained version of [RFC7030] EST described in [I-D.ietf-ace-coap-est].

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

- HTTP is replaced with CoAP.
- TLS (HTTPS) is replaced with 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 COSE signature
- 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 CoAP based stateless proxy mechanism described in [I-D.ietf-6tisch-minimal-security] section 7.1 is used.

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

- nonce-full vouchers are encouraged, but off-line nonce-less operation is also supported, however, the resulting vouchers would have infinite life.

802.1AR Client certificates are retained, but optionally are specified by reference rather than value (Work in Progress).

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. Prior Bootstrapping Approaches

Constrained devices as used in industrial control systems are usually installed (or replaced) by technicians with expertise in the equipment being serviced, not in secure enrollment of devices.

Devices therefore are typically pre-configured in advance, marked for a particular factory, assembly line, or even down to the specific machine. It is not uncommon for manufacturers to have a unique product (stock keeping unit -SKU) for each customer as the part will be loaded with customer specifc security configuration. The resulting customer-specific parts are hard to inventory and very expensive to provide spares for. Should a part be delivered to the wrong customer, determining the reason for inability to configure is difficult and time consuming.

End-user actions to configure the part at the time of installation, aside from being error prone, also suffer from requiring a part that has a user interface.

1.2. 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], [I-D.ietf-anima-bootstrapping-keyinfra] and [I-D.ietf-anima-constrained-voucher]. The following terms are

imported: drop ship, imprint, enrollment, pledge, join proxy, ownership voucher, and join registrar/coordinator (JRC). 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"

1.3. 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.

1.4. 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 SHOULD be used just to distribute network-wide keys using a protocol such as [I-D.ietf-6tisch-minimal-security].

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

XXX - this document does not yet provide a way to signal which mode the pledge should do.

1.4.1. Key Distribution Process

In addition to being used for the initial enrollment process, the secure channel SHOULD be kept open to use for network rekeying. The CoJP protocol described in [I-D.ietf-6tisch-minimal-security] includes a mechanism for rekeys in section 8.4.3.1.

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 desireable. The CoAP based stateless proxy mechanism described in [I-D.ietf-6tisch-minimal-security] section 7.1 MUST be used.

2.1. Behavior 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 the EDHOC handshake. The certificate MAY be presented by reference. (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 SHOULD 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, there is a voucher mechanism based upon [RFC8366]. The format and cryptographic mechansim of the constrained vouchers is described in detail in [I-D.ietf-anima-constrained-voucher].

COSE signed vouchers and voucher-requests are MANDATORY.

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 initial algorithms are the ECDSA P-256 (secp256v1). Newer devices SHOULD begin to appear using EdDSA curves using the 25519 curves.

The manufacturer will always know what algorithms are available in the Pledge, and will use an appropriate one. The other components that need to evaluate the IDevID (the Registrar and MASA) are expected to support all common algorithms.

The JRC is expected to be an easily updated appliance that can learn about new algorithms with a regular maintenance cycle.

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 the underlying 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 protocol optimizes for network bandwidth, and does not transfer the public key or certificate chain used to validate the voucher in-band.

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.

This use of a direct key has drawbacks, section Section 9.1 addresses some of them with some operational suggestions.

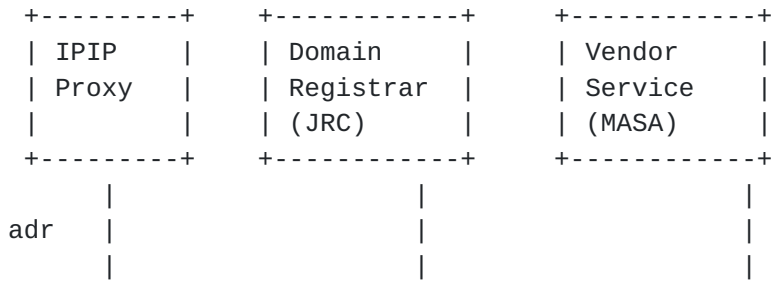
BRSKI places some clear requirements upon the contents of the IDevID, but leaves the exact origin of the voucher serial-number open. This document restricts the process to being the hwSerialNum OCTET STRING. As CWT can handle binary formats, no base64 encoding is necessary.

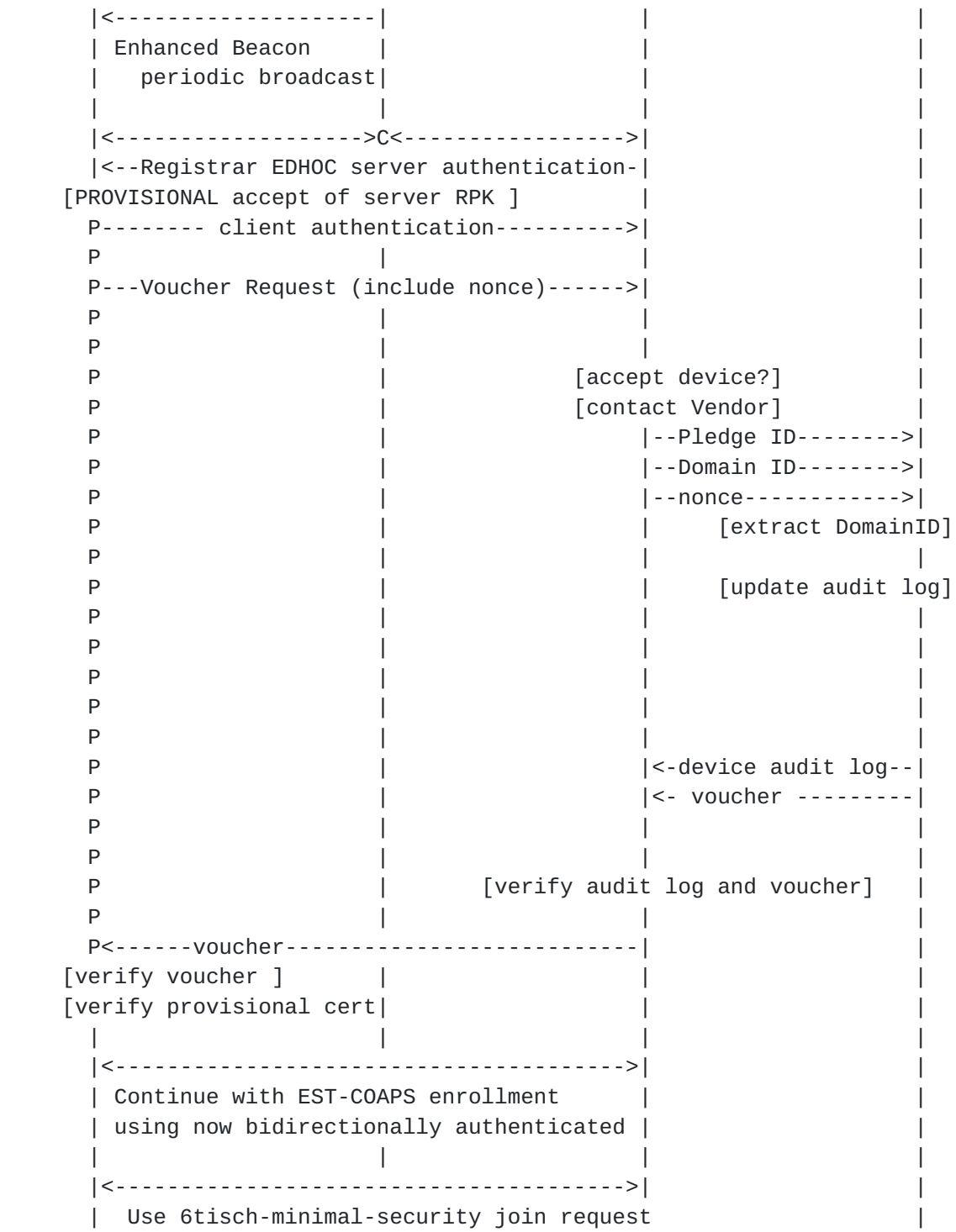
The MASA-URL extension MANDATORY. The inclusion of a MUD URL [RFC8520] is strongly recommended.

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

2.4. Protocol Flow

This 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 mandatory

2.5. Architectural Components

The bootstrap process includes the following architectural components:

2.5.1. 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.

2.5.2. Stateless IPIP Join Proxy

The stateless CoAP provides CoAP connectivity between the pledge and the registrar. The stateless CoAP proxy mechanism is described in [I-D.ietf-6tisch-minimal-security].

2.5.3. Domain Registrar

The Domain Registrar (having the formal name Join Registrar/ Coordinator (JRC)), operates as a CMC Registrar, terminating the CoAP-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.

2.5.4. Manufacturer Service

The Manufacturer 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.

2.6. Certificate Time Validation

2.6.1. 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 COSE 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.

2.7. Cloud Registrar

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

2.8. 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

The voucher-request artifact is defined in [I-D.ietf-anima-constrained-voucher] section 6.1.

For the 6tisch ZSJ protocol defined in this document, only COSE signed vouchers as described in [I-D.ietf-anima-constrained-voucher] section 6.3.2 are supported.

4. Proxying details (Pledge - Proxy - Registrar)

The voucher-request artifact is defined in [I-D.ietf-anima-constrained-voucher].

The 6tisch use of the constrained version differs from the non-constrained version in two ways:

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

2. the pledge uses the proximity-registrar-subject-public-key-info to verify the raw public key for the JRC.

An appendix of [I-D.ietf-anima-constrained-voucher] shows example requests and responses.

5. Proxy details

The role of the Proxy is to facilitate communication. In the constrained situation the proxy needs to be stateless as there is very little ram in constrained nodes, and none can be allocated to keep state for an unlimited number of potential pledges.

5.1. 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 ZSJ, the existence of the proxy is announced by the Enhanced Beacon message described in [I-D.richardson-6tisch-enrollment-enhanced-beacon]. The proxy as described by [I-D.ietf-6tisch-minimal-security] section 10 is to be used in an identical fashion when EDHOC and OSCOAP are used.

5.2. HTTPS proxy connection to Registrar

HTTPS connections are not used between the Pledge, Proxy and Registrar. The Proxy relays CoAP packets and does not interpret or terminate CoAP connections.

HTTPS is still used between the Registrar and MASA!

5.3. 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 JRC by the Join Response in in [I-D.ietf-6tisch-minimal-security] section 9.3.2. (As described in that section, the address of the registrar otherwise defaults to be that of the DODAG root)

Whether or not a 6LR will announce itself as a possible Join Proxy is outside the scope of this document.

6. Protocol Details (Pledge - Registrar - MASA)

BRSKI is specified to run over HTTPS. This document respecifies it to run over CoAP with either DTLS or EDHOC-provided OSCOAP security.

BRSKI introduces the concept of a provisional state for EST.

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

As in [I-D.ietf-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.

6.1. BRSKI-EST (D)TLS establishment details

6tisch ZSJ does not use TLS. The connection is CoAP with EDHOC security.

6.1.1. BRSKI-EST CoAP estasblishment details

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

The registrar SHOULD authenticate itself with a raw public key. A 256 bit ECDSA raw public key is RECOMMENDED. Pledges SHOULD support EDDSA keys if they contain hardware that supports doing so efficiently.

TBD: the Pledge needs to signal what kind of Raw Public Key it supports before the Registrar sends its ServerCertificate. Can SNI be used to do this?

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

6.1.2. BRSKI-EST CoAP/EDHOC estasblishment details

[I-D.selander-ace-cose-ecdhe] details how to use EDHOC. The EDHOC description identifiers a party U (the initiator), and a party V. The Pledge is the party U, and the JRC is the party V.

The communication from the Pledge is via CoAP via the Join Proxy. The Join proxy relays traffic to the JRC, and using the mechanism described in [I-D.ietf-6tisch-minimal-security] section 5.1. This is designed so that the Join Proxy does not need to know if it is performing the one-touch enrollment described in [I-D.ietf-6tisch-minimal-security] or the zero-touch enrollment protocol described in this document. A network could consist of a mix of nodes of each type.

As generating ephemeral keys is expensive for a low-resource Pledge, the use of a common E_U by the Pledge for multiple enrollment attempts (should the first turn out to be the wrong network) is encouraged.

The first communication detailed in [I-D.ietf-ace-coap-est] is to query the "/.well-known/core" resource to request the Link for EST. This is where the initial CoAP request is to sent.

The JRC MAY replace it's E_V ephermal key on a periodic basis, or even for every communication session.

The Pledge's ID_U is the Pledge's IDevID. It is transmitted in an x5bag [I-D.schaad-cose-x509]. An x5u (URL) MAY be used. An x5t (hash) MAY also be used and would be the smallest, but the Registrar may not know where to find the Pledge's IDevID unless the JRC has been preloaded will all the IDevIDs via out-of-band mechanism. It is impossible for the Pledge to know if the JRC has been loaded in such a way so x5t is discouraged for general use.

The JRC's ID_V is the JRC's Raw Public Key. It is transmitted as a key in COSE's YYY parameter.

The initial Mandatory to Implement (MTI) of an HKDF of SHA2-256, an AEAD based upon AES-CCM-16-64-128, a signature verification of TBD:BBBB, and signature generation of TBD:BBBB. The Pledge proposes a set of algorithms that it supports, and Pledge need not support more than one combination.

JRCs are expected to run on non-constrained servers, and are expected to support the above initial as MTI, and any subsequent ones that become common.

A JRC SHOULD support all available algorithms for a significant amount of time.

Even when algorithms become weak or suspect, it is likely that it will still have to perform secure join for older devices. A JRC that responds to such an older device might not in the end accept the

device into the network, but it is important that it be able to audit the event and communicate the event to an operator.

While EDHOC supports sending additional data in the message_3, in the constrained network situation, it is anticipated that the size of the this message will already be large, and no additional data is to be sent.

A COAP confirmable message SHOULD be used.

[I-D.ietf-6tisch-minimal-security] section 6 details how to setup OSCORE context given a shared key derived by EDHOC.

The registrar SHOULD authenticate itself with a raw public key.

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

6.2. 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 proximity-registrar-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 mandates that all voucher requests be signed.

6.3. 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-request and voucher response is COSE, 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. It is however, too late for the Registar to use a different key, but at least it can log a reason for a failure.

It is likely that the ZSJ-BRSKI-EST connection has already failed, and this step is never reached.

6.3.1. MASA renewal of expired vouchers

There are assumed to be no useful real-time clocks on constrained devices, so all vouchers are in effect infinite duration. Pledges will use nonces for freshness, and a request for a new voucher with a new voucher for the same Registrar is not unusual.

A token-bucket system SHOULD be used such that no more than 24 vouchers are issued per-day, but more than one voucher can be issued in a one hour period. Tokens should not accumulate for more than one day.

6.3.2. MASA verification of voucher-request signature consistency

The voucher-request is signed by the Registrar using it's Raw Public Key. There is no additional certificate authority to sign this key. The MASA MAY have this key via sales-channel integration, but in most cases it will be seeing the key for the first time.

XXX-should the TLS connection from Registrar to MASA have a ClientCertificate? If so, then should it use the same Public Key? Or a different one?

6.3.3. MASA authentication of registrar (certificate)

IDEA: The MASA SHOULD pin the Raw Public Key (RPK) to the IP address that was first used to make a request with it. Should the RPK <-> IP address relationship be 1:1, 1:N, N:1? Should we take IP address to mean, "IP subnet", essentially the IPv4/24, and IPv6/64? The value of doing is about DDoS mitigation?

Should above mapping be on a per-Pledge basis?

6.3.4. MASA revocation checking of registrar (certificate)

As the Registrar has a Raw Public Key as an identity, there is no meaningful standard revocation checking that can be done. The MASA SHOULD have a blacklist table, and a way to add entries, but this process is out of scope.

6.3.5. MASA verification of pledge prior-signed-voucher-request

The Registrar will put the signed pledge voucher-request into it's voucher-request as 'prior-signed-voucher-request'. The MASA can

verify the signature from the Pledge using the MASA's copy of the Pledge's IDevID public key.

6.3.6. MASA pinning of registrar

When the MASA creates a voucher, it puts the Registrar's Raw Public Key into the 'pinned-domain-subject-public-key-info' leaf of the voucher.

The MASA does not include the 'pinned-domain-cert' field in such vouchers.

6.3.7. MASA nonce handling

Use of nonces is highly RECOMMENDED, but there are situations where not all components are connected at the same time in which the nonce will not be present.

There are no significant changes from BRSKI.

6.4. MASA Voucher Response

As exaplained in [I-D.ietf-anima-constrained-voucher] section 6.3.2, when a voucher is returned by the MASA to the JRC, a public key or certificate container that will verify the voucher SHOULD also be returned.

In order to do this, the MASA MAY return a multipart/related return, within that body, two items SHOULD be returned:

1. An application/voucher-cose+cbor body.

2. An application/TBD:SOMETHING containing a Raw Public Key.

A MASA is not obligated to return the public key, and MAY return only the application/voucher-cose+cbor object. In that case, the JRC will be unable to validate it, and will have to just audit the contents.

6.4.1. Pledge voucher verification

The Pledge receives the voucher from the Registrar over it's CoAP connection. It verifies the signature using the MASA anchor built in, as in the BRSKI case.

6.4.2. Pledge 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.

6.5. Pledge 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.

6.6. Registrar audit log request

There are no changes to the Registrar audit log request.

6.6.1. MASA audit log response

There are no changes to the MASA audit log response.

6.6.2. Registrar audit log verification

There are no changes to how the Registrar verifies the audit log.

6.6.3. EST CSR Attributes

In 6tisch, no Autonomic Control Plane will be created, so none of the criteria for SubjectAltname found in [I-D.ietf-anima-autonomic-control-plane] apply.

The CSR Attributes request SHOULD NOT be performed.

6.6.4. EST Client Certificate Request

6tisch will use a certificate to:

1. to authenticate an 802.15.9 key agreement protocol.

2. to terminate an incoming DTLS or EDHOC key agreement as part of application data protection.

It is recommended that the requested subjectAltName contain only the [RFC4514] hwSerialNum.

6.6.5. Enrollment Status Telemetry

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

6.6.6. Multiple certificates

Multiple certificates are not supported.

6.6.7. EST over CoAP

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

6.7. Use of Secure Transport for Minimal Join

Rather than bootstrap to a public key infrastructure, the secure channel MAY instead be for the minimal security join process described in [I-D.ietf-6tisch-minimal-security].

The desire to do a minimal-security join process is signaled by the Registrar in it's voucher-request by including a 'join-process' value of 'minimal'. The MASA copies this value into the voucher that is creates, and also logs this to the audit log.

When the secure channel was created with EDHOC, then the keys setup by EDHOC are simply used by OSCORE exactly as if they had been Pre-Shared. The keys derived by EDHOC SHOULD be stored by both Registrar and Pledge as their long term key should the join process need to be repeated.

7. IANA Considerations

No specific requests are made

8. 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.

8.1. Privacy Considerations for Production network

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

8.2. 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.

8.2.1. 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.

9. Security Considerations

TBD

9.1. Security of MASA voucher signing key(s)

TBD

10. Acknowledgements

Kristofer Pister helped with many non-IETF references.

11. References

11.1. 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-security]
Vucinic, M., Simon, J., Pister, K., and M. Richardson, "Minimal Security Framework for 6TiSCH", draft-ietf-6tisch-minimal-security-11 (work in progress), June 2019.

[I-D.ietf-6tisch-terminology]
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terms Used in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-10 (work in progress), March 2018.

[I-D.ietf-ace-coap-est]
Stok, P., Kampanakis, P., Richardson, M., and S. Raza, "EST over secure CoAP (EST-coaps)", draft-ietf-ace-coap-est-12 (work in progress), June 2019.

[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-22 (work in progress), June 2019.

[I-D.ietf-anima-constrained-voucher]
Richardson, M., Stok, P., and P. Kampanakis, "Constrained Voucher Artifacts for Bootstrapping Protocols", draft-ietf-anima-constrained-voucher-04 (work in progress), July 2019.

[I-D.ietf-anima-voucher]
Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "Voucher Profile for Bootstrapping Protocols", draft-ietf-anima-voucher-07 (work in progress), January 2018.

[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-16 (work in progress), March 2019.

[I-D.richardson-6tisch-enrollment-enhanced-beacon]
Dujovne, D. and M. Richardson, "IEEE802.15.4 Informational Element encapsulation of 6tisch Join and Enrollment Information", draft-richardson-6tisch-enrollment-enhanced-beacon-01 (work in progress), April 2018.

[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-03 (work in progress), January 2018.

[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.schaad-cose-x509]
Schaad, J., "CBOR Object Signing and Encryption (COSE): Headers for carrying and referencing X.509 certificates", draft-schaad-cose-x509-03 (work in progress), December 2018.

[I-D.selander-ace-cose-ecdhe]
Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", draft-selander-ace-cose-ecdhe-13 (work in progress), March 2019.

[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>.

[RFC4514] Zeilenga, K., Ed., "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names", RFC 4514, DOI 10.17487/RFC4514, June 2006, <https://www.rfc-editor.org/info/rfc4514>.

[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>.

[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>.

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

11.2. Informative References

[I-D.ietf-anima-autonomic-control-plane] Eckert, T., Behringer, M., and S. Bjarnason, "An Autonomic Control Plane (ACP)", draft-ietf-anima-autonomic-control-plane-19 (work in progress), March 2019.

[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>.

[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>.

[RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <https://www.rfc-editor.org/info/rfc8520>.

Author's Address

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca