Network Working Group
Internet-Draft
Updates: 8995 (if approved)
Intended status: Standards Track
Expires: 7 January 2026

O. Friel
Cisco
R. Shekh-Yusef
Ciena
M. Richardson
Sandelman Software Works

6 July 2025

BRSKI Cloud Registrar
draft-ietf-anima-brski-cloud-16

Abstract

Bootstrapping Remote Secure Key Infrastructures (BRSKI) defines how to onboard a device securely into an operator-maintained infrastructure. It assumes that there is local network infrastructure for the device to discover and help the device. This document extends BRSKI and defines new device behavior for deployments where no local infrastructure is available, such as in a home or remote office. This document defines how the device can use a well-defined "call-home" mechanism to find the operator-maintained infrastructure.

This document defines how to contact a well-known Cloud Registrar, and two ways in which the new device may be redirected towards the operator-maintained infrastructure. The Cloud Registrar enables discovery of the operator-maintained infrastructure, and may enable establishment of trust with operator-maintained infrastructure that does not support BRSKI mechanisms.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-anima-brski-cloud/.

Discussion of this document takes place on the anima Working Group mailing list (mailto:anima@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/anima/. Subscribe at https://www.ietf.org/mailman/listinfo/anima/.

Source for this draft and an issue tracker can be found at https://github.com/anima-wg/brski-cloud.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Bootstrapping Remote Secure Key Infrastructures [BRSKI] BRSKI specifies automated and secure provisioning of nodes (which are called Pledges) with cryptographic keying material (trust anchors and certificates) to enable authenticated and confidential communication with other similarly enrolled nodes. This bootstrapping process is also called enrollment.

In BRSKI, the Pledge performs enrollment by communicating with a BRSKI Registrar belonging to the owner of the Pledge. The Pledge does not know who its owner will be when manufactured. Instead, in BRSKI it is assumed that the network to which the Pledge connects belongs to the owner of the Pledge and therefore network-supported discovery mechanisms can resolve generic, non-owner specific names to the owner's Registrar.

To support enrollment of Pledges without such an owner based access network, the mechanisms of BRSKI Cloud are required which assume that Pledge and Registrar simply connect to the Internet.

This work is in support of [BRSKI], Section 2.7, which describes how a Pledge MAY contact a well-known URI of a Cloud Registrar if a local Registrar cannot be discovered or if the Pledge's target use cases do not include a local Registrar.

This kind of non-network onboarding is sometimes called "Application Onboarding", as the purpose is typically to deploy a credential that will be used by the device in its intended use. For instance, a SIP [RFC3261] phone might have a client certificate to be used with a SIP proxy.

This document updates [BRSKI] by clarifying operations that are left out of scope in [BRSKI]. Two modes of operation are specified in this document. The Cloud Registrar MAY redirect the Pledge to the owner's Registrar, or the Cloud Registrar MAY issue a voucher to the Pledge that includes the domain of the owner's Enrollment over Secure Transport [RFC7030] (EST) server.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the terms Pledge, Registrar, MASA, and Voucher from [BRSKI] and [RFC8366bis].

Cloud Registrar: The default Registrar that is deployed at a URI that is well known to the Pledge.

EST: Enrollment over Secure Transport [RFC7030].

Local Domain: The domain where the Pledge is physically located and bootstrapping from. This may be different from the Pledge owner's domain.

Manufacturer: The term manufacturer is used throughout this document as the entity that created the Pledge. This is typically the original equipment manufacturer (OEM), but in more complex situations, it could be a value added retailer (VAR), or possibly even a systems integrator. Refer to [BRSKI] for more detailed descriptions of manufacturer, VAR and OEM.

Owner Domain: The domain that the Pledge needs to discover and bootstrap with.

Owner Registrar: The Registrar that is operated by the Owner, or the Owner's delegate. There may not be an Owner Registrar in all deployment scenarios.

OEM: Original Equipment Manufacturer. The company that created the device.

Provisional TLS: A mechanism defined in Section 5.1 of [BRSKI] whereby a Pledge establishes a provisional TLS connection with a Registrar before the Pledge is provisioned with a trust anchor that can be used for verifying the Registrar identity.

SIP: Session Initiation Protocol defined in [RFC3261]

VAR: Value Added Reseller. A VAR will often collect products from many OEMs, creating a complete solution, and then sells that composite solution to end customers. A VAR will often need to provision products to be operate in a specific manner. For instance, a VoIP phone might have SIP functionality as well as MGCP functionality, but in a particular deployment, only one will be used.

Cloud VAR Registrar: The non-default Registrar that is operated by a value added reseller (VAR).

1.2. Target Use Cases

This document specifies procedures for two high-level use cases.

* Bootstrap via Cloud Registrar and Owner Registrar: The operator-maintained infrastructure supports BRSKI and has a BRSKI Registrar deployed. More details are provided in Section 1.2.1.

* Bootstrap via Cloud Registrar and Owner EST Service: The operator-maintained infrastructure does not support BRSKI, does not have a BRSKI Registrar deployed, but does have an Enrollment over Secure Transport (EST) [RFC7030] service deployed. More detailed are provided in Section 1.2.2.

There are existing DHCP options that network operators use to configure devices such as a VoIP phone. This includes DHCP options 66 [RFC2132], 150 (TFTP/HTTP server names) [RFC5859], and 120 (SIP Server) [RFC3361], which inform a VoIP phone about how to do application onboarding. A network with an operator that is able to provision these options would also be able to use BRSKI without changes. Such a network has no need for the mechanisms described in this document!

Where the need for the mechanism is needed is to allow the use of BRSKI in many small sites, such as teleworkers working from home, with minimum expectations against their network infrastructure. In particular, the home office user is not qualified or authorized to change DHCP options for the local network.

The procedures defined in this document support situations where a manufacturer sells a number of devices (in bulk) to a Value Added Reseller (VAR). The manufacturer knows which devices have been sold to which VAR, but not who the ultimate owner will be. The VAR then sells devices to other entities, such as enterprises, and records this in the VARs Cloud Registrar. Specifically, the VAR will record that a specific device has been sold to an enterprise, and will know that when this device bootstraps it should be redirected to the enterprise’s Owner Registrar or Owner EST Service.

A typical example is a VoIP phone manufacturer provides telephones to a local system integration company (a VAR). The VAR records this sale in its Cloud VAR Registrar system. The VAR has sold a VoIP system to an enterprise (e.g., a SIP PBX). When a new employee needs a phone at their home office, the VAR ships that unit across town to the employee. When the employee plugs in the device and turns it on, the device will be provisioned with a LDevID and configuration that connections the phone with the Enterprises’ VoIP PBX. The home employee’s network has no special provisions.

The procedures define in this document also support a chain of VARs through chained HTTP redirects. This also supports a situation where in effect, a large enterprise might also stock devices in a central location.

The Pledge is not expected to know whether the operator-maintained infrastructure has a BRSKI Registrar deployed or not. The Pledge determines this based upon the response to its Voucher Request message that it receives from the Cloud Registrar. The Cloud Registrar is expected to determine whether the operator-maintained infrastructure has a BRSKI Registrar deployed based upon the identity presented by the Pledge.

A Cloud Registrar will receive BRSKI communications from all devices configured with its URI. This could be, for example, all devices of a particular product line from a particular manufacturer. When this is a significantly large number, a Cloud Registrar may need to be scaled with the usual web-service scaling mechanisms.

1.2.1. Bootstrap via Cloud Registrar and Owner Registrar

A Pledge is bootstrapping from a location with no Local Domain Registrar (for example, the small site or teleworker use case with no local infrastructure to provide for automated discovery), and needs to discover its Owner Registrar. The Cloud Registrar is used by the Pledge to discover the Owner Registrar. The Cloud Registrar redirects the Pledge to the Owner Registrar, and the Pledge completes bootstrap against the Owner Registrar.

This mechanism is useful to help an employee who is deploying a Pledge in a home or small branch office, where the Pledge belongs to the employer. As there is no Local Domain Registrar in the employee's local network, the Pledge needs to discover and bootstrap with the employer's Registrar which is deployed within the employer's network, and the Pledge needs the keying material to trust the Registrar. As a very specific example, an employee is deploying an IP phone in a home office and the phone needs to register to an IP PBX deployed in their employer's office.

Protocol details for this use case are provided in Section 4.1.

1.2.2. Bootstrap via Cloud Registrar and Owner EST Service

A Pledge is bootstrapping where the owner organization does not yet have an Owner Registrar deployed, but does have an EST service deployed. The Cloud Registrar issues a voucher, and the Pledge completes trust bootstrap using the Cloud Registrar. The voucher issued by the cloud includes domain information for the owner's EST service that the Pledge should use for certificate enrollment.

For example, an organization has an EST service deployed, but does not yet have a BRSKI-capable Registrar service deployed. The Pledge is deployed in the organization's domain, but does not discover a Local Domain Registrar or Owner Registrar. The Pledge uses the Cloud Registrar to bootstrap, and the Cloud Registrar provides a voucher that includes instructions on finding the organization's EST service.

This option can be used to introduce the benefits of BRSKI for an initial period when BRSKI is not available in existing EST Servers. Additionally, it can also be used long-term as a security-equivalent solution in which BRSKI and EST Server are set up in a modular fashion.

The use of an EST Server instead of a BRSKI Registrar may mean that not all the EST options required by [BRSKI] may be available and hence this option may not support all BRSKI deployment cases. For example, certificates to enroll into an ACP [RFC8994] needs to include an AcpNodeName (see [RFC8994], Section 6.2.2, which non-BRSKI-capable EST Servers may not support.

Protocol details for this use case are provided in Section 4.2.

2. Architecture

The high-level architectures for the two high-level use cases are illustrated in Figure 1 and Figure 2.

In both use cases, the Pledge connects to the Cloud Registrar during bootstrap.

For use case one, as described in Section 1.2.1, the Cloud Registrar redirects the Pledge to an Owner Registrar in order to complete bootstrap with the Owner Registrar. When bootstrapping against an Owner Registrar, the Owner Registrar will interact with a CA to assist in issuing certificates to the Pledge. This is illustrated in Figure 1.

For use case two, as described Section 1.2.2, the Cloud Registrar issues a voucher itself without redirecting the Pledge to an Owner Registrar. The Cloud Registrar will inform the Pledge what domain to use for accessing EST services in the voucher response. In this model, the Pledge interacts directly with the EST service to enroll. The EST service will interact with a CA to assist in issuing a certificate to the Pledge. This is illustrated in Figure 2.

It is also possible that the Cloud Registrar MAY redirect the Pledge to another Cloud Registrar operated by a VAR, with that VAR’s Cloud Registrar then redirecting the Pledge to the Owner Registrar. This scenario is discussed further in Sections Section 7.2 and 8.3.

The mechanisms and protocols by which the Registrar or EST service interacts with the CA are transparent to the Pledge and are outside the scope of this document.

The architectures show the Cloud Registrar and MASA as being logically separate entities. The two functions could of course be integrated into a single entity.

There are two different mechanisms for a Cloud Registrar to handle voucher requests. It can redirect the request to the Owner Registrar for handling, or it can return a voucher that includes an "est-

domain" attribute that points to the Owner EST Service. When returning a voucher, additional bootstrapping information is embedded in the voucher. Both mechanisms are described in detail later in this document.

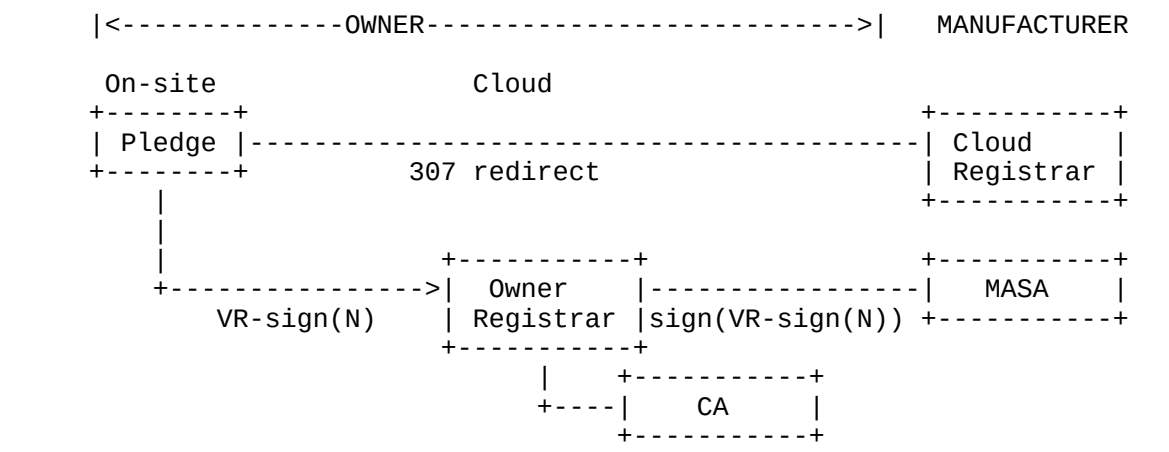


Figure 1: Architecture: Bootstrap via Cloud Registrar and Owner Registrar

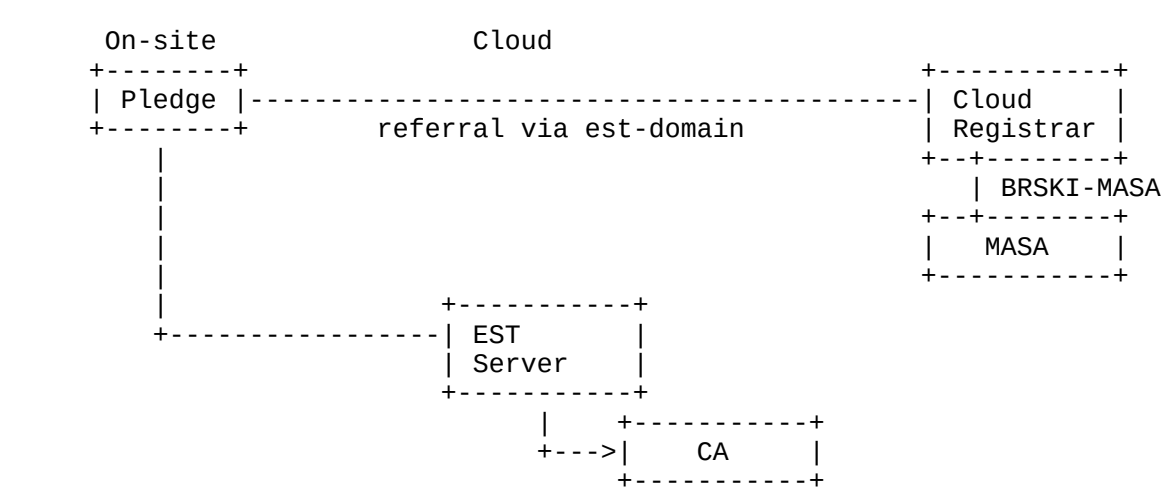


Figure 2: Architecture: Bootstrap via Cloud Registrar and Owner EST Service

As depicted in Figure 1 and Figure 2, there are a number of parties involved in the process. The Manufacturer, or Original Equipment Manufacturer (OEM) builds the device, but also is expected to run the MASA, or arrange for it to exist. The interaction between the Cloud Registrar and the MASA is described by [BRSKI], Section 5.4.

In Figure 1 the two signatures that the Pledge and the Owner Registrar place on the Voucher Request (VR) are shown as VR-sign(N) and sign(VR-sign(N)) This is as described in [BRSKI], Section 3. There are also signatures from Pledge to Cloud Registrar and to MASA in Figure 2, but they are omitted as they would make the diagram too busy.

The network operator or enterprise is the intended owner of the new device: the Pledge. This could be the enterprise itself, or in many cases there is some outsourced IT department that might be involved. They are the operator of the Registrar or EST Server. They may also operate the CA, or they may contract those services from another entity.

There is a potential additional party involved who may operate the Cloud Registrar: the value added reseller (VAR). The VAR works with the OEM to ship products with the right configuration to the owner. For example, SIP telephones or other conferencing systems may be installed by this VAR, often shipped directly from a warehouse to the customer’s remote office location. The VAR and manufacturer are aware of which devices have been shipped to the VAR through sales channel integrations, and so the manufacturer’s Cloud Registrar is able to redirect the Pledge through a chain of Cloud Registrars, as explained in Section 3.3.1.

2.1. Network Connectivity

The assumption is that the Pledge already has network connectivity prior to connecting to the Cloud Registrar. The Pledge must have an IP address so that it is able to make DNS queries, and be able to send requests to the Cloud Registrar. There are many ways to accomplish this, from routable IPv4 or IPv6 addresses, to use of NAT44, to using HTTP or SOCKS proxies.

The Pledge operator has already connected the Pledge to the network, and the mechanism by which this has happened is out of scope of this document.

For many telephony applications, this is typically going to be a wired connection. For wireless use cases, existing Wi-Fi onboarding mechanisms such as [WPS] can be used.

Similarly, what address space the IP address belongs to, whether it is an IPv4 or IPv6 address, or if there are firewalls or proxies deployed between the Pledge and the cloud registrar are all out of scope of this document.

2.2. Pledge Certificate Identity Considerations

Section 5.9.2 of [BRSKI] specifies that the Pledge MUST send an EST [RFC7030] CSR Attributes request to the EST server before it requests a client certificate. For the use case described in Section 1.2.1, the Owner Registrar operates as the EST server as described in [BRSKI], Section 2.5.3, and the Pledge sends the CSR Attributes request to the Owner Registrar. For the use case described in Section 1.2.2, the EST server operates as described in [RFC7030], and the Pledge sends the CSR Attributes request to the EST server. Note that the Pledge only sends the CSR Attributes request to the entity acting as the EST server as per Section 2.6 of [RFC7030], and MUST NOT send the CSR Attributes request to the Cloud Registrar, because the Cloud Registrar does not have authority to issue a certificate for the customer domain. (The Cloud Registrar is not a full EST server) If a Pledge sends a CSR Attributes request to the Cloud Registrar, then the Cloud Registrar MUST reply with 404 response code.

The EST server MAY use this mechanism to instruct the Pledge about the identities it should include in the CSR request it sends as part of enrollment. The EST server MAY use this mechanism to tell the Pledge what Subject or Subject Alternative Name identity information to include in its CSR request. This can be useful if the Subject or Subject Alternative Name identity must have a specific value in order to complete enrollment with the CA.

EST [RFC7030] is not clear on how the CSR Attributes response should be structured, and in particular is not clear on how a server can instruct a client to include specific attribute values in its CSR. [I-D.ietf-lamps-rfc7030-csrattrs] clarifies how a server can use CSR Attributes response to specify specific values for attributes that the client should include in its CSR.

For example, the Pledge may only be aware of its IDevID Subject which includes a manufacturer serial number, but must include a specific fully qualified domain name in the CSR in order to complete domain ownership proofs required by the CA.

As another example, the Registrar may deem the manufacturer serial number in an IDevID as personally identifiable information, and may want to specify a new random opaque identifier that the Pledge should use in its CSR.

2.3. YANG extension for Voucher based redirect

[RFC8366bis] contains the two needed voucher extensions: "est-domain" and "additional-configuration" which are needed when a client is redirected to a local EST server.

3. Protocol Operation

This section outlines the high-level protocol requirements and operations that take place. Section 4 outlines the exact sequence of message interactions between the Pledge, the Cloud Registrar, the Owner Registrar and the Owner EST server.

3.1. Pledge Sends Voucher Request to Cloud Registrar

3.1.1. Cloud Registrar Discovery

BRSKI defines how a Pledge MAY contact a well-known URI of a Cloud Registrar if a Local Domain Registrar cannot be discovered. Additionally, certain Pledge types might never attempt to discover a Local Domain Registrar and might automatically bootstrap against a Cloud Registrar.

The details of the URI are manufacturer specific, with BRSKI giving the example "brski-registrar.manufacturer.example.com".

The Pledge SHOULD be provided with the entire URI of the Cloud Registrar, including the protocol and path components, which are typically "https://" and "/.well-known/brski", respectively.

3.1.2. Pledge - Cloud Registrar TLS Establishment Details

According to [BRSKI], Section 2.7, the Pledge MUST use an Implicit Trust Anchor database (see EST [RFC7030]) to authenticate the Cloud Registrar service. The Pledge MUST establish a mutually authenticated TLS connection with the Cloud Registrar. Unlike the Provisional TLS procedures documented in Section 5.1 of [BRSKI], the Pledge MUST NOT establish a Provisional TLS connection with the Cloud Registrar.

Pledges MUST and Cloud/Owner Registrars SHOULD support the use of the "server_name" TLS extension (SNI, [RFC6066]) when using TLS 1.2. Support for SNI is mandatory with TLS 1.3.

Pledges SHOULD send a valid "server_name" extension (SNI) whenever they know the domain name of the registrar they connect to. A Pledge creating a Provisional TLS connection according to [BRSKI] will often only know the link local IPv6 address of a Join Proxy that connects

it to the Registrar. Registrars are accordingly expected to ignore SNI information, as in most cases, the Pledge will not know how to set the SNI correctly.

The Pledge MUST be manufactured with preloaded trust anchors that are used to verify the identity of the Cloud Registrar when establishing the TLS connection. The TLS connection can be verified using a public Web PKI trust anchor using [RFC9525] DNS-ID mechanisms or a pinned certification authority. This is a local implementation decision. Refer to Section 8.2 for trust anchor security considerations.

The Cloud Registrar MUST verify the identity of the Pledge by sending a TLS CertificateRequest message to the Pledge during TLS session establishment. The Cloud Registrar MAY include a certificate_authorities field in the message to specify the set of allowed IDevID issuing CAs that Pledges MAY use when establishing connections with the Cloud Registrar.

In addition to other protections against DoS attacks, the Cloud Registrar is able to reject TLS connections when it can determine during TLS authentication that it cannot support the Pledge. For example, the Pledge cannot provide an IDevID signed by a CA recognized/supported by the Cloud Registrar.

3.1.3. Pledge Sends Voucher Request Message

After the Pledge has established a mutually authenticated TLS connection with the Cloud Registrar, the Pledge generates a voucher request message as outlined in Section 5.2 of [BRSKI], and sends the voucher request message to the Cloud Registrar.

3.2. Cloud Registrar Processes Voucher Request Message

The Cloud Registrar MUST determine Pledge ownership. Prior to ownership determination, the Registrar checks the request for correctness and if it is unwilling or unable to handle the request, it MUST return a suitable 4xx or 5xx error response to the Pledge as defined by [BRSKI] and HTTP. The Registrar returns the following errors:

* in the case of an unknown Pledge, a 404 is returned.

* for a malformed request, 400 is returned.

* in case of server overload, 503 is returned.

If the request is correct and the Registrar is able to handle it, but unable to determine ownership at that time, then it MUST return a 401 Unauthorized response to the Pledge. This signals to the Pledge that there is currently no known owner domain for it, but that retrying later might resolve this situation. In this scenario, the Registrar SHOULD include a Retry-After header that includes a time to defer. The absence of a Retry-After header indicates to the Pledge not to attempt again. The Pledge MUST restart the bootstrapping process from the beginning.

A Pledge with some kind of indicator (such as a screen or LED) SHOULD consider all 4xx and 5xx errors to be a bootstrapping failure, and indicate this to the operator.

If the Cloud Registrar successfully determines ownership, then it MUST take one of the following actions:

* error: return a suitable 4xx or 5xx error response (as defined by [BRSKI] and HTTP) to the Pledge if the request processing failed for any reason.

* redirect to Owner Registrar: redirect the Pledge to an Owner Registrar via 307 response code.

* redirect to owner EST server: issue a voucher (containing an "est-domain" attribute) and return a 200 response code.

3.2.1. Pledge Ownership Look Up

The Cloud Registrar needs some suitable mechanism for knowing the correct owner of a connecting Pledge based on the presented identity certificate. For example, if the Pledge establishes TLS using an IDevID that is signed by a known manufacturing CA, the Registrar could extract the serial number from the IDevID and use this to look up a database of Pledge IDevID serial numbers to owners.

The mechanism by which the Cloud Registrar determines Pledge ownership is, however, outside the scope of this document. The Cloud Registrar is strongly tied to the manufacturers' processes for device identity.

3.2.2. Bootstrap via Cloud Registrar and Owner Registrar

Once the Cloud Registrar has determined Pledge ownership, the Cloud Registrar MAY redirect the Pledge to the Owner Registrar in order to complete bootstrap. If the owner wants the Cloud Registrar to redirect Pledges to their Owner Registrar, the owner must register their Owner Registrar URI with cloud Registrar. The mechanism by which Pledge owners register their Owner Registrar URI with the Cloud Registrar is outside the scope of this document.

In case of redirection, the Cloud Registrar replies to the voucher request with an HTTP 307 Temporary Redirect response code, including the owner’s Local Domain in the HTTP Location header.

3.2.3. Bootstrap via Cloud Registrar and Owner EST Service

If the Cloud Registrar issues a voucher, it returns the voucher in an HTTP response with a 200 response code.

The Cloud Registrar MAY issue a 202 response code if it is willing to issue a voucher, but will take some time to prepare the voucher.

The voucher MUST include the new "est-domain" field as defined in [RFC8366bis]. This tells the Pledge where the domain of the EST service to use for completing certificate enrollment.

The voucher MAY include the new "additional-configuration" field. This field points the Pledge to a URI where Pledge specific additional configuration information SHOULD be retrieved. For example, a SIP phone might retrieve a manufacturer specific configuration file that contains information about how to do SIP Registration. One advantage of this mechanism over current mechanisms like DHCP options 120 defined in [RFC3361] or option 125 defined in [RFC3925] is that the voucher is returned in a confidential (TLS-protected) transport, and so can include device-specific credentials for retrieval of the configuration.

The exact Pledge and Registrar behavior for handling and specifying the "additional-configuration" field is outside the scope of this document.

3.3. Pledge Handles Cloud Registrar Response

3.3.1. Bootstrap via Cloud Registrar and Owner Registrar

The Cloud Registrar has returned a 307 response to a voucher request. The Cloud Registrar MAY be redirecting the Pledge to the Owner Registrar, or to a different Cloud Registrar operated by a VAR.

The Pledge MUST restart its bootstrapping process by sending a new voucher request message (with a fresh nonce) using the location provided in the HTTP redirect.

The Pledge MUST attempt to validate the identity of the Cloud VAR Registrar specified in the 307 response using its Implicit Trust Anchor Database. If validation of this identity succeeds using the Implicit Trust Anchor Database, then the Pledge MAY accept a subsequent 307 response from this Cloud VAR Registrar.

The Pledge MAY continue to follow a number of 307 redirects provided that each 307 redirect target Registrar identity is validated using the Implicit Trust Anchor Database.

However, if validation of a 307 redirect target Registrar identity using the Implicit Trust Anchor Database fails, then the Pledge MUST NOT accept the 307 responses from the Registrar. At this point, the TLS connection that has been established is considered a Provisional TLS, as per Section 5.1 of [BRSKI].

The Pledge then (re)sends a voucher-request on this connection. As explained by [BRSKI], the connection is validated using the pinned credential from the voucher.

The Pledge MUST process any error messages as defined in [BRSKI], and in case of error MUST restart the process from its provisioned Cloud Registrar. The exception is that a 401 Unauthorized code SHOULD cause the Pledge to retry a number of times over a period of a few hours.

In order to avoid permanent bootstrap cycles, the Pledge MUST NOT revisit a prior location. Section 7.2 further outlines risks associated with redirects. However, in some scenarios, Pledges MAY visit the current location multiple times, for example when handling a 401 Unauthorized response, or when handling a 503 Service Unavailable that includes a Retry-After HTTP header. If it happens that a location is repeated, then the Pledge MUST fail the bootstrapping attempt and go back to the beginning, which includes listening to other sources of bootstrapping information as specified in [BRSKI] section 4.1 and 5.0. The Pledge MUST also have a limit on the total number of redirects it will a follow, as the cycle detection requires that it keep track of the places it has been. That limit MUST be in the dozens or more redirects such that no reasonable delegation path would be affected.

When the Pledge cannot validate the connection, then it MUST establish a Provisional TLS connection with the specified Local Domain Registrar at the location specified.

The Pledge then sends a voucher request message via the Local Domain Registrar.

After the Pledge receives the voucher, it verifies the TLS connection to the Local Domain Registrar and continues with enrollment and bootstrap as per standard BRSKI operation.

The Pledge MUST process any error messages as defined in [BRSKI], and in case of error MUST restart the process from its provisioned Cloud Registrar.

The exception is that a 401 Unauthorized code SHOULD cause the Pledge to retry a number of times over a period of a few hours.

3.3.2. Bootstrap via Cloud Registrar and Owner EST Service

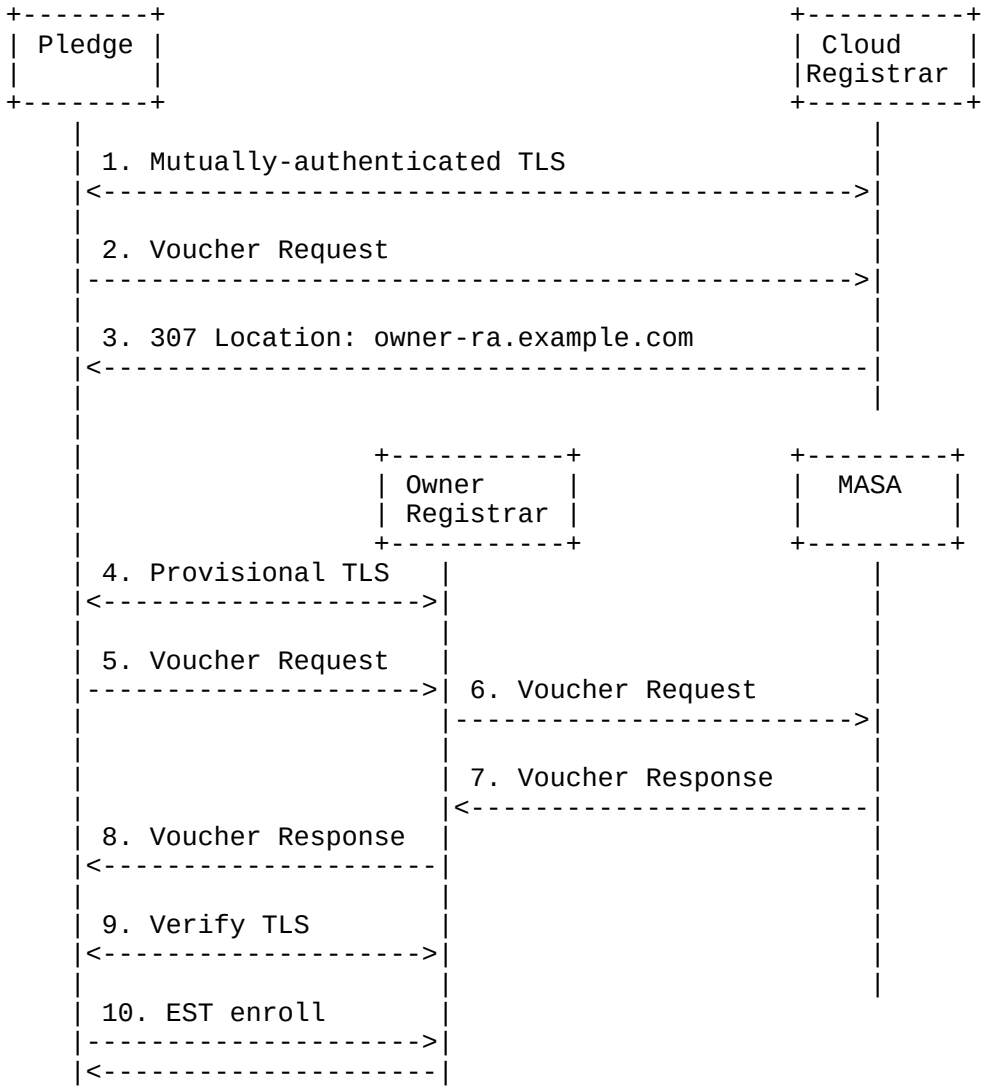
The Cloud Registrar returned a voucher to the Pledge. The Pledge MUST perform voucher verification as per Section 5.6.1 of [BRSKI].

The Pledge SHOULD extract the "est-domain" field from the voucher, and SHOULD continue with EST enrollment as per standard EST operation. Note that the Pledge has been instructed to connect to the EST server specified in the "est-domain" field, and therefore SHOULD use EST mechanisms, and not BRSKI mechanisms, when connecting to the EST server.

4. Protocol Details

4.1. Bootstrap via Cloud Registrar and Owner Registrar

This flow illustrates the "Bootstrap via Cloud Registrar and Owner Registrar" use case. A Pledge is bootstrapping in a remote location with no Local Domain Registrar. The assumption is that the Owner Registrar domain is accessible, and the Pledge can establish a network connection with the Owner Registrar. This may require that the owner network firewall exposes the Owner Registrar on the public internet.



The process starts, in step 1, when the Pledge establishes a Mutual TLS channel with the Cloud Registrar using the IDevID certificate and the trust anchors created during the manufacturing process of the Pledge.

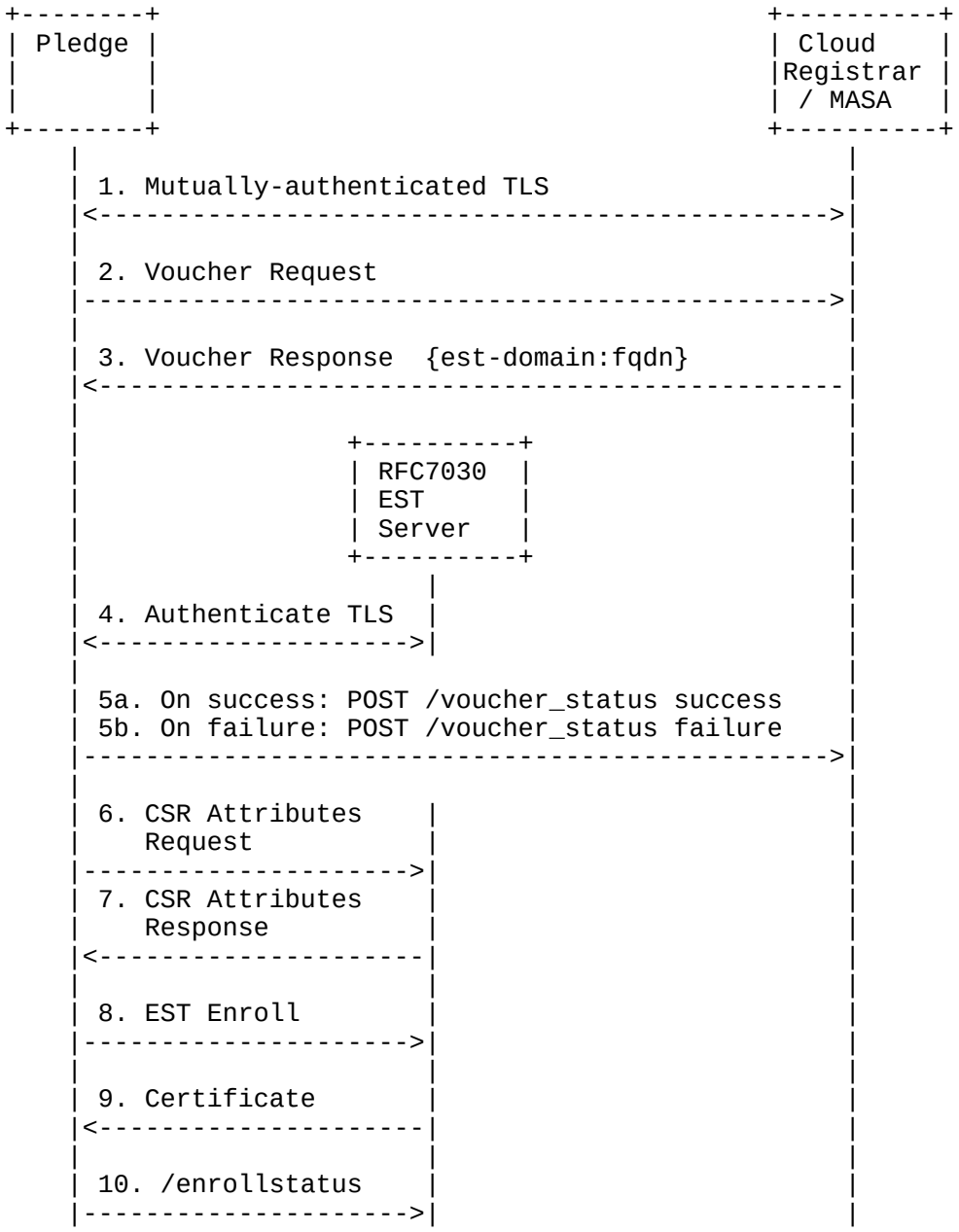
In step 2, the Pledge sends a voucher request to the Cloud Registrar.

The Cloud Registrar determines Pledge ownership look up as outlined in Section 3.2.1, and determines the Owner Registrar domain. In step 3, the Cloud Registrar redirects the Pledge to the Owner Registrar domain.

Steps 4 and onwards follow the standard BRSKI flow, which includes doing EST enroll operations. The Pledge establishes a Provisional TLS connection with the Owner Registrar, and sends a voucher request to the Owner Registrar. The Registrar forwards the voucher request to the MASA. Assuming the MASA issues a voucher, then the Pledge verifies the TLS connection with the Registrar using the pinned-domain-cert from the voucher and completes the BRSKI flow.

4.2. Bootstrap via Cloud Registrar and Owner EST Service

This flow illustrates the "Bootstrap via Cloud Registrar and Owner EST Service" use case. A Pledge is bootstrapping in a location with no Local Domain Registrar. The Cloud Registrar is instructing the Pledge to connect directly to an EST server for enrollment using EST mechanisms. The assumption is that the EST domain is accessible, and the Pledge can establish a network connection with the EST server.



The process starts, in step 1, when the Pledge establishes a Mutual TLS channel with the Cloud Registrar/MASA using artifacts created during the manufacturing process of the Pledge.

In step 2, the Pledge sends a voucher request to the Cloud Registrar/ MASA.

In step 3, the the Cloud Registrar/MASA replies to the Pledge with an [RFC8366bis] format voucher that includes its assigned EST domain in the "est-domain" attribute.

In step 4, the Pledge establishes a TLS connection with the EST RA that was specified in the voucher "est-domain" attribute. The connection may involve crossing the Internet requiring a DNS look up on the provided name. It MAY also be a local address that includes an IP address literal including both IPv4 [RFC1918] and IPv6 Unique Local Addresses [RFC4193]. The Pledge attempts to authentiate the TLS connection and verify the EST server identity. The artifact provided in the pinned-domain-cert is trusted as a trust anchor, and is used to verify the EST server identity. The EST server identity MUST be verified using the pinned-domain-cert value provided in the voucher as described in [RFC7030] section 3.3.1.

There is a case where the pinned-domain-cert is the identical End-Entity (EE) Certificate as the EST server. It also explicitly includes the case where the EST server has a self-signed EE Certificate, but it MAY also be an EE certificate that is part of a larger PKI. If the certificate is not a self-signed or EE certificate, then the Pledge SHOULD apply [RFC9525] DNS-ID verification on the certificate against the domain provided in the "est-domain" attribute. If the "est-domain" was provided with an IP address literal, then it is unlikely that it can be verified, and in that case, it is expected that either a self-signed certificate or an EE certificate will be pinned by the voucher.

In steps 5.a and 5.b, the Pledge MAY optionally notify the Cloud Registrar/MASA of the success or failure of its attempt to establish a secure TLS channel with the EST server. This is described in Section 5.7 of [BRSKI] This telemetry returns allow for the Registrar to better provide diagnostics in the event of failure to onboard. if the Pledge fails to verify the identity of the EST server, it MUST drop the connection and MUST NOT continue with a CSR Attributes request or an EST Enroll request.

In step 6, the Pledge follows the procedures outlined in {pledge-certificate-identity-considerations} and sends a CSR Attributes request to the EST server before sending the EST Enroll request.

In step 7, the EST server returns the CSR Attributes response.

In step 8, the Pledge sends an EST Enroll request with the CSR.

In step 9, the EST server returns the requested certificate.

Step 10 is described in Section 5.9.4 of [BRSKI] as the Enrollment Status Telemetry. This telemetry return also allows for better diagnostics in the event of a failure.

5. Lifecycle Considerations

BRSKI and the Cloud Registrar support provided in this document are dependent upon the manufacturer maintaining the required infrastructure. Section 10.7 of [BRSKI] outlines additional considerations about manufacturer life span.

Sections 11.5 and 11.6 of [BRSKI] outline additional considerations about device trust anchors and how devices establish trust.

The well-known URL that is used is specified by the manufacturer when designing its firmware, and is therefore completely under the manufacturer’s control. If the manufacturer wishes to change the URL, or discontinue the service, then the manufacturer will need to arrange for a firmware update where appropriate changes are made.

Often the firmware can not be updated because there is significant inventory in a warehouse. If the Pledge were powered on and connected, then it would get firmware updates. Since it is not, any URLs built-in to the old firmware need to be maintained until all copies of that firmware have been replaced. This could be a challenge if a company is going out of business, and in which case the considerations from [BRSKI], Section 10.7 apply.

If a merger between two companies happens, then it is possible to consolidate the MASA of each company into a single system. The consolidated MASA will need access to a MASA signing key for both companies to operate correctly. One way is for both MASA names (such as masa.company1.example, and masa.company2.example) to be added as SubjectAltNames for the HTTPS certificates used by the MASA. The Cloud Registrar will need a similar treatment. As an alternative to operating a Registrar under two names, all access to one Cloud Registrar could be replaced with a 307 redirect as described in Section 7.2.

Additionally, in the hosted Registrar use case, with an Owner EST Server Section 4.2 use case, the Cloud Registrar MUST know the certificate for the EST Server in order to pin it properly. In that case, when the owner of the EST Server wishes to change their certificate, then they MUST coordinate this with the upstream Cloud Registrar operator.

6. IANA Considerations

This document makes no IANA requests.

7. Implementation Considerations

7.1. Captive Portals

A Pledge might find itself deployed in a network where a captive portal or an intelligent home gateway that provides access control on all connections is also deployed. Captive portals that do not follow the requirements of Section 1 of [RFC8952] might forcibly redirect HTTPS connections. While this is a deprecated practice as it breaks TLS in a way that most users can not deal with, it is still common in many networks.

When the Pledge attempts to connect to any Cloud Registrar, an incorrect connection will be detected because the Pledge will be unable to verify the TLS connection to its Cloud Registrar via DNS-ID check Section 6.3 of [RFC9525]. That is, the certificate returned from the captive portal will not match.

At this point a network operator who controls the captive portal, noticing the connection to what seems a legitimate destination (the Cloud Registrar), MAY then permit that connection. This enables the first connection to go through.

The connection is then redirected to the Registrar via 307, or to an EST server via "est-domain" in a voucher. If it is a 307 redirect, then a Provisional TLS connection will be initiated, and it will succeed. The Provisional TLS connection does not do DNS-ID verification ([RFC9525], Section 6.3), so the forced redirection to a captive portal system will not be detected. However, the subsequent BRSKI POST of a voucher request will most likely be met by a 404 or 500 HTTP code. Even if somehow it did work (because the captive portal was in fact an attacker), any returned voucher would not be signed by a trusted MASA.

It is RECOMMENDED therefore that the Pledge look for Captive-Portal Identification attributes [RFC8910] in DHCP, and if present, use the Captive-Portal API [RFC8908] to learn if it is captive.

The scenarios outlined here when a Pledge is deployed behind a captive portal may result in failure scenarios, but do not constitute a security risk, so long as the Pledge is correctly verifying all TLS connections as per [BRSKI].

7.2. Multiple HTTP Redirects

If the Redirect to Registrar method is used, as described in Section 4.1, there MAY be a series of 307 redirects. An example of why this might occur is that the manufacturer only knows that it resold the device to a particular value added reseller (VAR), and there MAY be a chain of such VARs. It is important the Pledge avoid being drawn into a loop of redirects. This could happen if a VAR does not think they are authoritative for a particular device. A "helpful" programmer might instead decide to redirect back to the manufacturer in an attempt to restart at the top: perhaps there is another process that updates the manufacturer’s database and this process is underway. Instead, the VAR MUST return a 404 error if it cannot process the device. This will force the device to stop, timeout, and then try all mechanisms again.

There are additional considerations regarding TLS certificate validation as outlined in Section 3.3.1. If the Registrar returns a 307 response, the Pledge MUST NOT follow this redirect if the Registrar identity was not validated using its Implicit Trust Anchor Database. If the Registrar identity was validated using the Implicit Trust Anchor Database, then the Pledge MAY follow the redirect.

8. Security Considerations

The Cloud Registrar described in this document inherits all the strong security properties that are described in [BRSKI], and none of the security mechanisms that are defined in [BRSKI] are bypassed or weakened by this document. The Cloud Registrar also inherits all the potential issues that are described in [BRSKI]. This includes dependency upon continued operation of the manufacturer provided MASA, as well as potential complications where a manufacturer might interfere with resale of a device.

In addition to the dependency upon the MASA, the successful enrollment of a device using a Cloud Registrar depends upon the correct and continued operation of this new service. This internet accessible service might be operated by the manufacturer and/or by one or more value-added-resellers. All the considerations for operation of the MASA also apply to operation of the Cloud Registrar.

8.1. Security Updates for the Pledge

Unlike many other uses of BRSKI, in the Cloud Registrar case it is assumed that the Pledge has connected to a network, such as the public Internet, on which some amount of connectivity is possible, but there is no other local configuration available. (Note: there are many possible configurations in which the device might not have unlimited connectivity to the public Internet, but for which there might be connectivity possible. For instance, the device could be without a default route or NAT44, but able to make HTTP requests via an HTTP proxy configured via DHCP.)

There is another advantage to being online: the Pledge SHOULD contact the manufacturer before bootstrapping in order to apply any available firmware patches. Manufacturers are encouraged to make MUD [RFC8520] files available, and in those definitions to allow for retrieval of firmware updates. This may also include updates to the Implicit list of Trust Anchors. In this way, a Pledge that may have been in a dusty box in a warehouse for a long time can be updated to the latest (exploit-free) firmware before attempting bootstrapping.

8.2. Trust Anchors for Cloud Registrar

In order to validate the HTTPS connections to the (series of) Cloud Registrars, the Pledge will need to have an Implicit Trust Anchor database, as described in [RFC7030], Section 3.6.1, to verify the Cloud Registrar’s certificate.

There is no requirement that Cloud Registrar’s certificates are part of the public (WebPKI) database, but it is likely simpler and cheaper for most such systems to use easily obtained certificates.

Device manufacturers therefore need to include enough trust anchor in their devices (the Pledges) so that all expected Cloud Registrar’s can be validated. This argues for including more trust anchors.

On the other hand, minimizing the number of trust anchors reduces the security exposure should fraudulent certificates ever be issued. More trust anchors also implies more maintenance to maintain and update this Implicit Trust Anchor database as different certification authorities renew their trust anchors.

A device manufacturer could instead ship only their own internal, private trust anchors for a PKI that the manufacturer operates. This is described in in [I-D.irtf-t2trg-taxonomy-manufacturer-anchors] section 3. This would imply that all Cloud Registrars (likely operated by VARs) would have to obtain a certificate from the manufacturer. This has advantages in reliability and predictability,

but likely makes the Cloud Registrars much more costly to operate. In particular, tying the VARs' Cloud Registrar to a single manufacturer means that the VARs might have to operate a Cloud Registrar for each brand of equipment that they represent.

The recommendation is therefore for manufacturers to work with their VARs to determine if there is a subset of public PKIs that would satisfy all their VARs, and to ship only that subset.

The final onboarding step, wherein an [RFC8366bis] voucher artifact is returned to authenticate the provisional TLS connection, can use any kind of trust anchor: private or public. In most cases, the end customer's Registrar will have a private PKI that will be pinned by the voucher.

8.3. Considerations for HTTP Redirect

When the default Cloud Registrar redirects a Pledge using HTTP 307 to an Owner Registrar, or another Cloud Registrar operated by a VAR, the Pledge MUST have validated the TLS connection using an Implicit Trust Anchor.

However, when connecting to the target Owner Registrar, a provisional TLS connection is required as explained in [BRSKI], Section 5.1.

There is a conflict between these requirements: one says to validate, and the other one says not to. This is resolved by having the Pledge attempt validation, and if it succeeds, then an HTTP 307 redirect will be accepted. If validation fails, then an HTTP 307 redirect MUST be rejected as an error. If that occurs, then the onboarding process SHOULD restart after a delay. This failure should be reported to the initial Cloud Registrar via the mechanism described in [BRSKI], Section 5.7.

Note that for use case two, in which redirection to an EST Server occurs, then there is no provisional TLS connection at all. The connection to the last Cloud Registrar is validated using the Implicit Trust Database, while the EST Server connection is validated by the certificate pinned by the Voucher artifact.

8.4. Considerations for Voucher est-domain

A Cloud Registrar supporting the same set of Pledges as a MASA MAY be integrated with the MASA to avoid the need for a network based API between them, and without changing their external behavior as specified here.

When a Cloud Registrar handles the scenario described in {bootstrap-via-cloud-registrar-and-owner-est-service} by the returning "est-domain" attribute in the voucher, the Cloud Registrar MUST do all the voucher processing as specified in [BRSKI]. This is an example deployment scenario where the Cloud Registrar MAY be operated by the same entity as the MASA, and it MAY even be integrated with the MASA.

When a voucher is issued by the Cloud Registrar and that voucher contains an "est-domain" attribute, the Pledge MUST verify the TLS connection with this EST server using the "pinned-domain-cert" attribute in the voucher.

The reduced operational security mechanisms outlined in Sections 7.3 and 11 of [BRSKI] MAY be supported when the Pledge connects with the EST server. These mechanisms reduce the security checks that take place when the Pledge enrolls with the EST server. Refer to [BRSKI] sections 7.3 and 11 for further details.

Acknowledgements

The authors would like to thank for following for their detailed reviews: (ordered by last name): Esko Dijk, Toerless Eckert, Sheng Jiang.

References

Normative References

[BRSKI] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

[I-D.ietf-lamps-rfc7030-csrattrs] Richardson, M., Friel, O., von Oheimb, D., and D. Harkins, "Clarification and enhancement of RFC7030 CSR Attributes definition", Work in Progress, Internet-Draft, draft-ietf-lamps-rfc7030-csrattrs-23, 28 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-rfc7030-csrattrs-23>.

[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/rfc/rfc2119>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/rfc/rfc6066>.

[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/rfc/rfc7030>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8366bis] Watsen, K., Richardson, M., Pritikin, M., Eckert, T. T., and Q. Ma, "A Voucher Artifact for Bootstrapping Protocols", Work in Progress, Internet-Draft, draft-ietf-anima-rfc8366bis-14, 1 April 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-anima-rfc8366bis-14>.

[RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <https://www.rfc-editor.org/rfc/rfc8994>.

Informative References

[I-D.irtf-t2trg-taxonomy-manufacturer-anchors] Richardson, M., "A Taxonomy of operational security considerations for manufacturer installed keys and Trust Anchors", Work in Progress, Internet-Draft, draft-irtf-t2trg-taxonomy-manufacturer-anchors-09, 28 May 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-t2trg-taxonomy-manufacturer-anchors-09>.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/rfc/rfc1918>.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <https://www.rfc-editor.org/rfc/rfc2132>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/rfc/rfc3261>.

[RFC3361] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers", RFC 3361, DOI 10.17487/RFC3361, August 2002, <https://www.rfc-editor.org/rfc/rfc3361>.

[RFC3925] Littlefield, J., "Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4)", RFC 3925, DOI 10.17487/RFC3925, October 2004, <https://www.rfc-editor.org/rfc/rfc3925>.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <https://www.rfc-editor.org/rfc/rfc4193>.

[RFC5859] Johnson, R., "TFTP Server Address Option for DHCPv4", RFC 5859, DOI 10.17487/RFC5859, June 2010, <https://www.rfc-editor.org/rfc/rfc5859>.

[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/rfc/rfc8520>.

[RFC8908] Pauly, T., Ed. and D. Thakore, Ed., "Captive Portal API", RFC 8908, DOI 10.17487/RFC8908, September 2020, <https://www.rfc-editor.org/rfc/rfc8908>.

[RFC8910] Kumari, W. and E. Kline, "Captive-Portal Identification in DHCP and Router Advertisements (RAs)", RFC 8910, DOI 10.17487/RFC8910, September 2020, <https://www.rfc-editor.org/rfc/rfc8910>.

[RFC8952] Larose, K., Dolson, D., and H. Liu, "Captive Portal Architecture", RFC 8952, DOI 10.17487/RFC8952, November 2020, <https://www.rfc-editor.org/rfc/rfc8952>.

[RFC9525] Saint-Andre, P. and R. Salz, "Service Identity in TLS", RFC 9525, DOI 10.17487/RFC9525, November 2023, <https://www.rfc-editor.org/rfc/rfc9525>.

[WPS] Wi-Fi Alliance, "Wi-Fi Protected Setup (WPS)", January 2025, <https://www.wi-fi.org/discover-wi-fi/wi-fi-protected-setup>.

Authors' Addresses

Owen Friel
Cisco
Email: ofriel@cisco.com

Rifaat Shekh-Yusef
Ciena
Email: rifaat.s.ietf@gmail.com

Michael Richardson
Sandelman Software Works
Email: mcr+ietf@sandelman.ca