Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: 17 August 2024

M. Richardson
Sandelman Software Works
W. Pan
Huawei Technologies

14 February 2024

Operational Considerations for BRSKI Registrar draft-richardson-anima-registrar-considerations-08

Abstract

This document describes a number of operational modes that a BRSKI Registration Authority (Registrar) may take on.

Each mode is defined, and then each mode is given a relevance within an over applicability of what kind of organization the Registrar is deployed into. This document does not change any protocol mechanisms.

This document includes operational advice about avoiding unwanted consequences.

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-richardson-anima-registrar-considerations/.

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/.

Source for this draft and an issue tracker can be found at https://github.com/mcr/registrar-operational-considerations.git.

Status of This Memo

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

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

This Internet-Draft will expire on 17 August 2024.

Table of Contents

1. Introduction

[RFC8995] introduces a mechanism for new devices (called pledges) to be onboarded into a network without intervention from an expert operator.

A key aspect of this is that there has to be a thing for the pledge to join! [RFC8995] refers to this thing as the "Domain", identified technically by the "DomainID". The Registrar component embodies the identity, membership and trust anchor of the domain. Membership in the domain is proven by possession of a valid Local DeviceID, a form of [ieee802-1AR] certificate.

The Registrar is the component that implements the domain, authorizing new devices (pledges) to join. Proper and efficient operation of the Registrar is key aspect for the Autonomic mechanisms, and for enabling secure onboarding.

This document provides implementation, deployment and operational guidance for the BRSKI Registrar.

There are however several classes of operator of a local domain: ISP and large managed multi-side Enterprises are the primary target for this document. Medium sized single site Enterprises and Industrial Plant users are a secondary target for this document. Unmanaged small enterprises and home users are addressed in a separate section at the end as special case.

This document first introduces the different scales of deployment as a reference for further discussion and contrasts, and then provides analyses some consequences of architectural choices that may be appropriate for different scales of deployments.

The document includes security best practices for the management of the certificates and the certification authorities.

1.1. Terminology

Although this document is not an IETF Standards Track publication, it adopts the conventions for normative language to provide clarity of instructions to the implementer. 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.

1.2. Reference Network and Diagrams

In order to deal with the full complexity and generality of operations, the reference network described herein is a bit more complicated than many networks actually are.

XXX-some of these diagrams as more complex than the document currently justifies.

1.2.1. Tier-1 Network

In this guide one target is a world-wide Tier-1 ISP. It has three network operations centers (NOC), the two major ones in Frankfurt and Denver, with an secondary center located in Perth, Australia. The exact location of these NOCs is not important: the locations have been chosen to have an hour overlap in their 8-6 daytime shift, typical of world-wide operations. This overlap is also not important, it just adds a degree of realism to this discussion. The use of actual names makes subsequent discussion about failures easier.

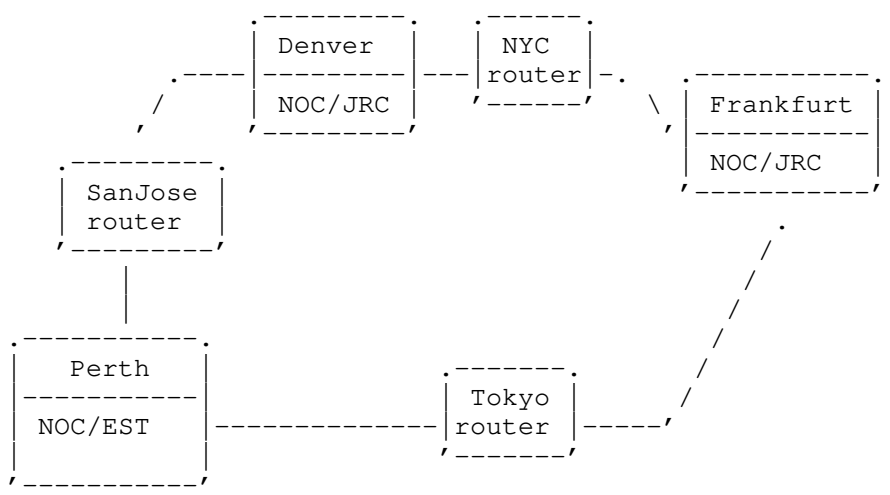

Figure 1: Reference Tier-1 ISP network

XXX-there were some extended consequences that this diagram was anticipating, which have yet to be writen.

1.2.2. Enterprise Network

A second target is a medium Enterprise that has a single (probably on-premise) data center. The Enterprise has Information Technology (IT) operations that include the routers and systems supporting it's office staff in it's buildings. It has Building Operations which integrates the IoT devices found in the buildings that it owns, and it has Operations Technology (OT) that manages the automated systems in it's on-site manufacturing facilities.

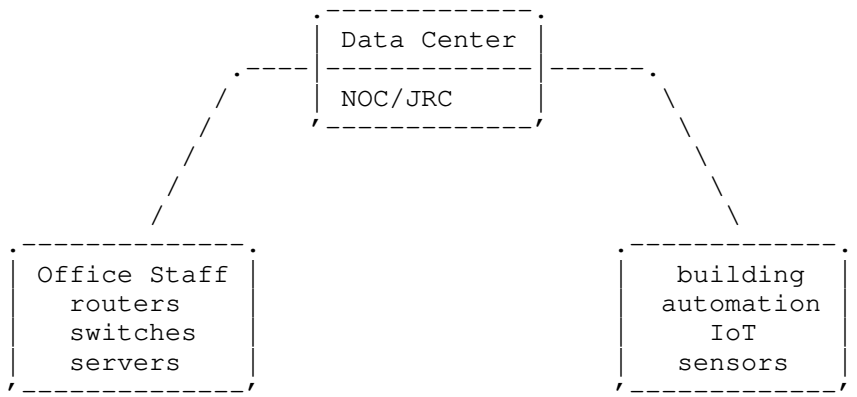

Figure 2: Reference Enterprise network

1.2.3. Home Network

A third target is a resident with a single CPE device. The home owner has a few medium sized devices (a home NAS) as well as a few IoT devices (light bulbs, clothes washing machine).

1.3. Internal architectural view

A Registrar will have four major interfaces, connected together by a common database.

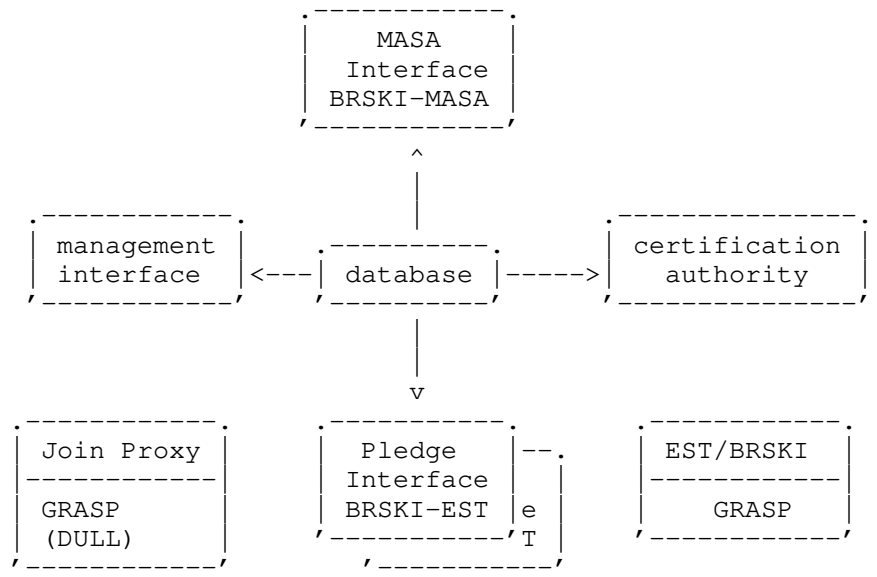

Figure 3: Reference Internal Architecture for Registrar

1.3.1. Pledge Interface (Southbound Interface)

The pledge interface is the southbound interface. This interface runs the BRSKI-EST protocol. It may also offer a constrained-BRSKI protocol using CoAP as described in [I-D.ietf-anima-constrained-voucher]. It may further offer ultra-constrained onboarding protocols such as [I-D.selander-lake-authz].

This interface faces into the operator's network, receiving requests from devices to join the network.

There is no requirement that the different onboarding protocols run on the same system, or from the same IP address. They may also be seperated onto different networks, and perform all of their coordination through the database.

For [RFC8995] use, the pledge interface is an HTTPS interface.

Due to the use of provinsional trust state in the BRSKI-EST interface the pledge never verifies the contents of the TLS server certificate. The registrar may also run on arbitrary port numbers, as the port number is part of the announcements used in the discovery protocol(s). The voucher pins the associated certificate, so the Registrar does not need to have any specific (subjectAltName) dnsName.

[I-D.ietf-anima-constrained-join-proxy] describes a mechanism to provide a stateless proxy of CoAPS connections, in which case DTLS traffic will be proxied by the Join Proxy to the port that the Registrar announces via GRASP within the ACP. In this case, then there is DTLS layer below the CoAP layer.

[RFC9031] describes a proxy mechanism that can be used with [I-D.selander-lake-authz] to pass CoAP traffic. In this case, depending upon the chosen AKE, the key agreement protocol would be above CoAP.

[I-D.richardson-anima-state-for-joinrouter] offers some additional mechanisms, one of which involves dynamically created IPIP tunnels. If these mechanisms are in use, then the southbound interface would need to support these options as well.

The Pledge Interface requires a TLS ServerCertificate, and Section 5.1 discusses option for creating this certificate.

The certificates (or DH keys) used for the different protocols could entirely different. If horizontal scaling is used, where there are multiple systems offering a BRSKI-EST interface (probably using a load balancing mechanism) then it is not necessary to have the same private keys for each system. This assumption requires that the entire BRSKI-EST protocol exchange occur in a single TLS session (i.e. using HTTP/1.1 sessions), or that the load balancing system is able to consistently map each pledge to the same BRSKI-EST interface.

As explained above, the Pledge Inteface does not require a public IP address, nor does it have have to run on port 443. The address and port of the Pledge interface to the Registrar is advertised by the Registrar using GRASP, according to [RFC8995] section 4.1.1. The service may run on any available port. The HTTPS, CoAP and CoAPS port numbers do not need to be coordinated.

In an ACP application ([RFC8994]), the Pledge Interface SHOULD have an IPv6 Unique Local Address (ULA) address from the prefix allocated to the ACP. Section 2 provides some options for how the Pledge Interface can be best connected to the ACP.

Outside of the ACP context, running the Pledge interface on an IP address that has a FQDN that resolves to that IP address (if only internally), and operating it on port 443 may have operational advantages. The Registrar may have additional management functions, it may also serve as an EST end point for certificate renewal, and [I-D.ietf-anima-brski-cloud] proposes a mechanism to bootstrap devices which are not connected by a convex ACP, or no ACP. The Registrar may be accessible via multiple interfaces.

1.3.2. MASA client (Northbound Interface)

The MASA client interface connects outward to the Internet to speak to the Manufacturer Authorized Signing Authority (MASA). This is a TLS Client interface.

Use of a TLSClientCertificate is RECOMMENDED as this may be the best way for a manufacturer to identify clients. Section 5.2 discusses options for signing this certificate.

The Northbound interface (V->W) described in [I-D.selander-lake-authz] may require a proof of possesion of the (private) key which the pledge (U) has witnessed. In that case, this proof of posession may need to be done in the Southbound BRSKI-EST interface, and stored in the database for use by the Northbound BRSKI-MASA system. The private keys from the Southbound interfaces SHOULD NOT be made available on the Northbound interfaces.

The MASA client interface is outgoing only and does not require any special connectivity. It may be placed behind a typical enterprise or residential NAT44 gateway. IPv6 connectivity is RECOMMENDED however, as an increasing number of MASA may prefer IPv6 only connectivity. It does need access to DNS, and the DNS lookups SHOULD be validated with DNSSEC.

The MASA client interface will need to validate the server certificates of the MASA, and to do this it will need access to the common public WebPKI ([WebPKI]) trust anchors to validate the MASA. The MASA client MAY also require access to a database of pinned certificates to validate specific manufacturers as called out for in [RFC8995] section 2.8 and section 5.4.

1.3.3. Join Proxy (Southbound Interface)

In the ACP context, the Registrar is expected to have a Join Proxy operating on the Southbound Interface in order to announce the existence of the Registrar to the local network, for the benefit of directly connected devices. This permits the systems on the LAN in the NOC itself to autonomically join the domain.

The Join Proxy MAY announce the IP address (ULA) and port of the actual Pledge Interface, rather than announcing a link-local address and then performing a proxy operation.

1.3.4. EST and BRSKI GRASP announcements

As specified in [RFC8995] section 4.3, in an ACP context, the Registrar MUST announce itself inside the ACP using GRASP. The Registrar MUST incorporate enough of a GRASP daemon in order to perform the M_FLOOD announcements.

As specified in [RFC8995] section 6.1.2, in an ACP context, if the Registrar will also be providing for renewal of certificates using EST, then it SHOULD announce itself inside the ACP using GRASP. See [RFC8994] section 6.1.5.1 for details. Unless made impossible due to loading concerns, it is RECOMMENDED that all Registrar instances offer certificate renewal services in this fashion.

The use of [RFC8739] Short-Term Automatically-Renewed Certificates is RECOMMENDED. This mandates that the EST server be highly available. If STAR-style renewals are not used, then the Certification Authority will need to make OCSP or CRL Distribution points available.

1.3.5. Certification Authority

If the Enterprise/ISP has an existing certification authority system that it wishes to use, then an interface to it has to be enabled. This may run protocols like EST, CMP or ACME.

Smaller Enterprises and Residential uses of BRSKI are encouraged to use an internal (private) certification authority. See Section 3 for a discussion of securing this CA.

1.3.6. Management Interface

The Registrar will require a management interface. As is the trend, this will often be a web-based single page application using AJAX API calls to perform communications. This interface SHOULD be made available on the Southbound NOC interface only, and it MUST be on a different IP address and port number then the BRSKI-EST interface. It should be secured with HTTPS, and use of a public ([WebPKI]) anchor is reasonable as it may be that the internal certification authority may be unavailable or require maintenance.

An entirely separate process is justified with the only connection to the other procesess being the database. (This does not mean it can not share code modules)

2. Connecting the Autonomic Control Plane to the Network Operations Center (NOC)

[RFC8994] section 8.1 describes a mechanism to connect non-ACP capable systems to the ACP. The use of this mechanism is critical to incremental deployment of ANIMA and BRSKI in operators.

The deployment of BRSKI capable equipment would ideally occur in an outward wave, a concentric ring, from the NOC.

(EDNOTE: INSERT DIAGRAMS)

This would start by an upgrade of the router that connects the NOC to the production network. This device needs to support the ACP connect functionality.

It is possible, but beyond the scope of this document, to do initial connectivity of the ACP and of multiple NOCs by manually configured IPsec tunnels. This is likely an important step for incremental initial deployment.

The Registrar described in the next section either needs to be connected via one of the above mentioned tunnels, or it must be located on a network with ACP Connect, or it must itself be part of an automatically configured ACP. It is quite reasonable for the Registrar to be part of a larger appliance that also includes an ACP Connect functionality.

3. Public Key Infrastructure Recommendations for the Registrar

The Registrar requires access to, or must contain a Certification Authority (CA).

This section deals with the situation where the CA is provided internally. [I-D.ietf-acme-integrations] deals with the case where the CA is provided by an external service, and the CA trust anchors are public. These use ACME ([RFC8555]) is used as the interface. That is out of scope for this document.

There are also a number of commercial offerings where a private CA is operated by an external entity using a wide variety of protocols, including proprietary ones. Those are also out of scope for this document.

The requirements for the PKI depends upon what kind of network is being managed.

3.1. PKI recommendations for Tier-1/ISP Networks

A three-tier PKI infrastructure is appropriate for an ISP. This entails having a root CA created with the key kept offline, and a number of intermediate CAs that have online keys that issue "day-to-day" certificates.

Whether the root private key is secured by applying secret-splitting, and then storing the results on multiple USBs key kept in multiple safes, or via Hardware Security Module is a local decision informed by best current practices.

The root CA is then used to sign a number of intermediate entities: this will include an intermediate CA for the Registrar that is deployed into each redundant NOC location. Multiple intermediate CAs with a common root provides significantly more security and operational flexibility than attempts to share a private key among locations.

While the root CA should have a longevity of at least 5 years, after which it can be re-signed rather than re-generated. (Resigning an existing key might not require replacement of trust anchors on all devices)

The intermediate CA keys need only have a 1-2 year duration, and before the end of their lifetime, a new private key should be generated and replace the old one.

Shorter periods are possible, but until there is more experience with them, not recommended. The intermediate CA key should be regenerated because the intermediate CA private key will need to be online, available to the Registrar CA system. There are many more opportunities for the key to leak, such as into backups.

The intermediate CA is then used to sign End-Entity certificates which are returned as part of the BRSKI-EST mechanism.

The Registrar needs both of client and server certificates for it's BRSKI-EST and BRSKI-MASA connections. It is recommended that an additional intermediate CA can be created for manually issued certificates such as these. This intermediate CA could be called the NOC Infrastructure CA, and could be used to issue certificates for all manner of infrastructure such as web-based monitoring tools. The private root CA certificate should be installed into the browsers of NOC personnel.

The document [I-D.moskowitz-ecdsa-pki] provides some practical instructions on setting up this kind of system.

This document recommends the use of ECDSA keys for the root and subordinate CAs, but there may be operational reasons why an RSA subordinate CA will be required for some legacy equipment.

3.2. Enterprise Network

Enterprises that have multiple Network Operations Center should consider the recommendations above for an ISP.

This section applies to Enterprises that have all NOC operations/ personel into a single location, which is probably on-premise data center. This is not a hard rule for scaling, but the intent is that physical security for the ACP Connect network is rather easy, that only a single legal jurisdiction will apply, and that it is possible to get people together easily to do things like resign keys.

A three-tier PKI infrastructure is still recommended for the reason that it provides operational continuity options not available with a two-level system. The recommendation is to have a root CA mechanism installed on a Virtual Machine which is not connected to a network. The root CA private key is kept offline, secret split among a number of USB keys, kept in the possession of key personnel.

The secret split should have at least five components, of which at least three are required to reconstruct the key. See [I-D.hallambaker-mesh-udf] section 4.5 for one such mechanism, there are many others, and there are no interoperability requirements for the secret split.

The essential point is that the Enterprise is able to recover the root CA key even without some number of personnel and is able to continue operating it's network.

As in the ISP case, the intermediate CA is then used to sign End-Entity certificates which are returned as part of the BRSKI-EST mechanism. One intermediate CA key suffices as there is only one NOC location with a Registrar. Incidental certificates for internal operations (such as internal web servers, email servers, etc.), and for the BRSKI-EST server certificate can be done with this single intermediate CA.

The BRSKI-MASA TLS Client Certificate key for an enterprise may not be needed; it depends upon the policies of the manufacturers which are involved. It may be simpler to use a certificate produced by a public CA (such as LetsEncrypt), as this makes it easier for manufacturers to validate the provided certificate.

The document [I-D.moskowitz-ecdsa-pki] provides some practical instructions on setting up this kind of system. This document recommends the use of ECDSA keys for the root and intermediate CAs. In an Enterprise, there are likely many more legacy devices that might need to become involved in the secure domain. It is recommended that an RSA root and intermediate CA be more strongly considered.

3.3. Home Network

Home networks and small offices that use residential class equipment are the most challenging situation. The three-tier PKI architecture is not justified because the ability to keep the root CA offline has no operational value.

The home network registrar should be initialized with a single key pair used as the certification authority.

Secret splitting is useful in order to save the generated key with a few neighbours. It is recommended that the entire PKI system database (including CA private key) be encrypted with a symmetric key and the results made available regularly for download to a variety of devices. The symmetric key is split among the neighbours.

The most difficult part of the Home Network PKI and Registrar is where to locate it. Generally it should be located on a device that is fully owned by the home user. This is sometimes the Home Router, but in a lot of situations the Home Router is the ISP's CPE router. If the home has a Network Attached Storage (NAS) system, then running it there is probably better.

A compromise for CPE devices owned by the ISP that can run containers is for the Registrar to be located on detachable storage that is inserted into the CPE. The detachable storage is owned by the home

owner, and can be removed from the CPE device if it is replaced. More experience will be necessary in order to determine if this is a workable solution.

4. Architecture Considerations for the Registrar

There are a number of ways to scale the Registrar. Web framework three-tier mechanisms are the most obvious. See [threetier] for an overview. This architecture is very familiar and can work well for a Registrar. There are a few small issues that need to be addressed relating to the TLS connections.

The BRSKI-EST connection uses TLS Client Certificate information, so it is necessary for the presentation tier to pass the entire certificate through to the application layer. The presentation tier MUST accept all Client Certificates, many of which might it might not have anchors for. Many n-tier systems provide for non-standard ways to transmit the client certificate from presentation layer to application layer, but [I-D.bdc-something-something-certificate] also intends to provide a standards track mechanism.

In addition, the Registrar Voucher-Request MUST be signed using the same key pair that is used to terminate the TLS connection, so the application layer will need access to the same keypair that the presentation tier uses. This can be operationally challenging if the presentation tier is provided by a hardware-based TLS load balancer.

For this reason, an alternate architecture where the front-end load balancer provides TCP level load balancing, leaving the TLS operations to software TLS implementations in the application layer may be simpler to build. Given that the Registrar is an inward facing system, and is not subject to the Internet-scale loads typical of "Black Friday" web system, the same kind of extreme scaling is not necessary.

The BRSKI-EST flow includes a back-end call to the BRSKI-MASA flow. That is, during the BRSKI-EST /voucherrequest call, a voucher will need to be fetched from the MASA using a BRSKI-MASA connection. There are three ways to do this.

4.1. Completely Synchronous Registrar

In this simplest version the Registrar operates as a single thread, processing the voucher-request from the Pledge, and then starting a BRSKI-MASA client session, while the connection from the Pledge waits.

This flow is very simple to implement, but requires an entire processing thread to block while the BRSKI-MASA protocol executes. The Pledge may timeout on this request, disconnect and retry. Experience so far is that typical default timeouts work fine.

It is recommended that the voucher-request be recorded in a database, and if a corresponding fresh voucher is also found in the database, that it be returned rather than fetching a new voucher from the MASA. This accomodates the situation where the Pledge did timeout, but the BRSKI-MASA protocol did complete. This results in the Pledge receiving the voucher upon retrying without having to go through the process of getting a new voucher. This only works if the Pledge retries with the same Nonce each time.

4.2. Partially Synchronous Registrar

A slightly more complicated version is for the Registrar to look in a database for a matching voucher-request, and if none is found, to return a 202 code upon the voucher-request, asking the Pledge to retry.

In the meantime the BRSKI-MASA connection can be performed, and the resulting voucher stored in a database. The connection can be done in the same thread that just deferred the connection, or in another thread kicked off for this purpose.

4.3. Asynchronous Registrar

In the completely asynchronous architecture, things work as with the Partially Synchronous version. The voucher request is placed into a database as before.

A completely separate system, probably with different network connectivity, but connected to the same database, performs the BRSKI-MASA processing, then fills the database with the answer.

This version may have a noticeably higher latency as it requires a database operation and a database trigger to invoke the process. This architecture has the advantage, however, that the internal facing Registrar never connects to the Internet. Furthermore, the number of internal facing Registrar instances can be tuned independently from the number of outward facing clients. This may be an advantage for networks that need to deal with a high number of malicious or lost internal clients.

5. Certificates needed for the Registrar

In addition to hosting a PKI root, the Registrar needs several other key pairs. They are:

5.1. TLS Server Certificate for BRSKI-EST

A certificate to be used to answer TLS connections from new devices (pledges). This must be of a type that expected pledges can understand. Returning an RSA key to a client that can validate only ECDSA chains is a problem. The constrained IoT ecosystem prefers ECDSA, and often does not have code that can verify RSA. Meanwhile, older FIPS-140 validated libraries present in many router operating systems support only RSA!

The recommendation is to use ECDSA keys, with a sensitiviity to when a majority of systems might support EdDSA. There are well established mechanisms in most TLS server libraries to permit multiple certificates to be loaded and to return an appropriate key based upon the client capabilities. This should be used.

The certificate used for the BRSKI-EST end point is not validated by the BRSKI pledge using public trust anchors, but rather it is pinned by the [RFC8366] voucher. As such it can come from the private CA, as recommended above: either signed by a specific intermediate CA or via a root CA as appropriate for the environment.

5.2. TLS Client Certificate for BRSKI-MASA

A certificate may optionally be used for authentication of the Registrar to the MASA. It is recommended to always include one.

It can be the same certificate used by TLS Server Certificate above, and this is most appropriate in small Registrars which are not distributed, such as ones aimed as Residential/Home networks.

In larger, distributed Registrars, cryptographic hygiene dictates that the private key not be distributed, so a unique certificate per Registrar client is appropriate. They should all be signed by the same intermediate CA, with the intermediate and root CA certificates being supplied in the TLS connection.

5.2.1. Use of Publically Anchored TLS Client Certificate with BRSKI-MASA connection

The use TLS Client Certificate which has a public anchor (such as from LetsEncrypt) has an advantage that is makes it easier for the MASA to reject malicious clients.

If the Registrar is not using a supply chain integration that includes the MASA being aware of the cryptographic identity of the Registrar, then the use of a publically anchored certificate is RECOMMENDED.

5.3. Certificate for signing of Voucher-Requests

As part of the BRSKI voucher-request process the Pledge's Voucher-Request is wrapped by the Registrar in another voucher-request and signed. It is this certificate which pinned by MASA to validate the connection.

The certificate used to sign the (parboiled) voucher-request MUST be the same as the one that is used for the TLS Server Connection. This implies that the signed voucher-request MUST be constructed on the same machine that terminates the BRSKI-EST connection.

6. Autonomic Control Plane Addressing

In the Enterprise and ISP use cases, the creation of an [RFC8994] Autonomic Control Plane is assumed. (The use of an ACP for the Home Network of IoT devices is considered unnecessary due to HNCP)

In these context the certificates which are returned by the Registrar MUST contain a unique IPv6 ULA address. [RFC8994] section 6.10 outlines several addressing schemes for the ULA addresses. The use of the ACP Vlong Addressing Sub-Scheme (6.10.5) is recommended as it provides the most flexibility for devices.

The use of this mode limits the number of nodes in the network to between 32768 and 8 Million. 32K routers in an ISP network seems like quite a lot already, but use of F=0 addresses provides for up to 8 Million devices, each with 256 management end points.

It should be noted that a mix of F=0 and F=1 addresses may be used, but the BRSKI protocol does not directly provide a way to negotiated this. This could be done as part of the Certificate Signing Request: the device could decide which kind of address to ask for by changing the address that it asks for, but this is non-standardized and may not work.

A network manager that saw that a device was running out of F=0 space, that is if 256 addresses was not enough for a device, could allocate an F=1 address in a management interface. At the next certificate renewal (which could be forced by a management action), then a new certificate would be issues with the larger address space.

256 addresses for a single device may seem like a lot, but it is increasing the case that routers may have a large number of virtualized functions within and each may reasonably need to be separately connected to it's SDN controller.

7. Privacy Considerations

Section 10.2 of [RFC8995] details a number of things that are revealed by the BRSKI-EST protocol. A multi-location Registrar with different TLS Server Certificates will have a different privacy profile than a Registrar that uses only a single certificate.

Section 10.3 of [RFC8995] details what is revealed by the BRSKI-MASA protocol. The operational recommendations of this document do not affect or mitigate things at all.

8. Security Considerations

Section 11 of [RFC8995] does not deal with any attacks against the Registrar, as the Registrar is considered to be an internally facing system.

In the context of the Autonomic Control Plane ([RFC8995] section 9, and [RFC8994]) it is expected that the majority of equipment attached to a network are connected by wired ethernet. The opportunity for a massive attack against the Registrar is considered low in an ISP, or multi-side Enterprise backbone network.

8.1. Denial of Service Attacks against the Registrar

However, there are some exposures which need to be taken into account, particular in the Enterprise or Institutional Campus network: typically these networks have large number of access ports, one for each desktop system. Those systems can be infected with Malware, or may be located in student computer labs physically accessible with minimal authorization. While an attack on the Registrar might be part of some kind of student protest, an attack by malware seems more likely.

The different architectures proposed in Section 4 of this document provides some recommendations on differing scales. The use of a fully asynchronous design is recommended for Enterprise uses of BRSKI where there may be a large number of IoT devices that are expected to onboard. The ability to scale the BRSKI-EST Pledge Interface without having the scale the rest of the system provides for resiliency of the Registrary.

It bears repeating that the use of of a stateless technology in the Join Proxy moves the load due to attacking systems from the Join Proxy into the Registrar. This increases the network bandwidth required from the Join Proxy to the Registrar with the benefit of simplifying the Join Proxy.

This is an intentional design decision to centralize the impact into the purpose built Registrar system(s).

8.2. Loss of Keys/Corruption of Infrastructure

In Home/Residential Network ("homenet") uses of [RFC8995] the biggest risk is likely that of loss of the Registrar's key pairs. That is, accidental loss of the private key is more likely than loss to a malicious entity that steals them with intent to cause damage.

This can be due to failure to backup the database followed by a CPE device failure, but the case where a CPE device is simply thrown away to be replaced by an uninformed technician or household member is probably the most likely situation.

This situation results in loss of control for all devices in the home, and much frustration from the home owner who has to go through an onboarding process for all the devices. The use of a standardized onboarding protocol significantly mitigates the hassle; the current "state of the art" proccess involves a series of appliance-specific smartphone applications, which may or not not actually work on more recent devices.

This is why the focus on saving of the database along with a secret splitting process to secure it. At present there is no cross-vendor format for this database, so the saved data is only useable with a Registrar from the same vendor. So this protects against device failure, where it is replaced by an identical device or an upward compatible device from the same manufacturer, but not against changes of vendor.

9. IANA Considerations

This document makes no IANA allocations.

10. Acknowledgements

Your name here.

11. Changelog

12. References

12.1. Normative References

[I-D.ietf-anima-constrained-voucher]
Richardson, M., Van der Stok, P., Kampanakis, P., and E. Dijk, "Constrained Bootstrapping Remote Secure Key Infrastructure (cBRSKI)", Work in Progress, Internet-Draft, draft-ietf-anima-constrained-voucher-23, 10 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-anima-constrained-voucher-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/info/rfc2119>.

[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/info/rfc8174>.

[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>.

[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/info/rfc8994>.

[RFC8995] 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/info/rfc8995>.

12.2. Informative References

[I-D.bdc-something-something-certificate]
Campbell, B., "Client-Cert HTTP Header: Conveying Client Certificate Information from TLS Terminating Reverse Proxies to Origin Server Applications", Work in Progress, Internet-Draft, draft-bdc-something-something-certificate-05, 23 March 2021, <https://datatracker.ietf.org/doc/html/draft-bdc-something-something-certificate-05>.

[I-D.friel-acme-integrations]
Friel, O., Barnes, R., and R. Shekh-Yusef, "ACME Integrations", Work in Progress, Internet-Draft, draft-friel-acme-integrations-02, 24 October 2019, <https://datatracker.ietf.org/doc/html/draft-friel-acme-integrations-02>.

[I-D.hallambaker-mesh-udf]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part II: Uniform Data Fingerprint.", Work in Progress, Internet-Draft, draft-hallambaker-mesh-udf-18, 28 June 2023, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-udf-18>.

[I-D.ietf-acme-integrations]
Friel, O., Barnes, R., Shekh-Yusef, R., and M. Richardson, "ACME Integrations for Device Certificate Enrollment", Work in Progress, Internet-Draft, draft-ietf-acme-integrations-17, 13 July 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-acme-integrations-17>.

[I-D.ietf-anima-brski-cloud]
Friel, O., Shekh-Yusef, R., and M. Richardson, "BRSKI Cloud Registrar", Work in Progress, Internet-Draft, draft-ietf-anima-brski-cloud-08, 24 August 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-anima-brski-cloud-08>.

[I-D.ietf-anima-constrained-join-proxy]
Richardson, M., Van der Stok, P., and P. Kampanakis, "Join Proxy for Bootstrapping of Constrained Network Elements", Work in Progress, Internet-Draft, draft-ietf-anima-constrained-join-proxy-15, 6 November 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-anima-constrained-join-proxy-15>.

[I-D.moskowitz-ecdsa-pki]
Moskowitz, R., Birkholz, H., Xia, L., and M. Richardson, "Guide for building an ECC pki", Work in Progress, Internet-Draft, draft-moskowitz-ecdsa-pki-10, 31 January 2021, <https://datatracker.ietf.org/doc/html/draft-moskowitz-ecdsa-pki-10>.

[I-D.richardson-anima-state-for-joinrouter]
Richardson, M., "Considerations for stateful vs stateless join router in ANIMA bootstrap", Work in Progress, Internet-Draft, draft-richardson-anima-state-for-

joinrouter-03, 22 September 2020,
<https://datatracker.ietf.org/doc/html/draft-richardson-anima-state-for-joinrouter-03>.

[I-D.selander-lake-authz]
Selander, G., Mattsson, J. P., Vuini, M., Richardson, M., and A. Schellenbaum, "Lightweight Authorization using Ephemeral Diffie-Hellman Over COSE", Work in Progress, Internet-Draft, draft-selander-lake-authz-03, 7 July 2023, <https://datatracker.ietf.org/doc/html/draft-selander-lake-authz-03>.

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

[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>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/info/rfc8555>.

[RFC8739] Sheffer, Y., Lopez, D., Gonzalez de Dios, O., Pastor Perales, A., and T. Fossati, "Support for Short-Term, Automatically Renewed (STAR) Certificates in the Automated Certificate Management Environment (ACME)", RFC 8739, DOI 10.17487/RFC8739, March 2020, <https://www.rfc-editor.org/info/rfc8739>.

[RFC9031] Vuini, M., Ed., Simon, J., Pister, K., and M. Richardson, "Constrained Join Protocol (CoJP) for 6TiSCH", RFC 9031, DOI 10.17487/RFC9031, May 2021, <https://www.rfc-editor.org/info/rfc9031>.

[threetier]
Wikipedia, "Multitier architecture", December 2019, <https://en.wikipedia.org/wiki/Multitier_architecture>.

[WebPKI] CA/Browser Forum, "CA/Browser Forum Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, v.1.2.2", October 2014, <https://cabforum.org/wp-content/uploads/BRv1.2.2.pdf>.

Authors' Addresses

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

Wei Pan
Huawei Technologies
Email: william.panwei@huawei.com