ANIMA WG
Internet-Draft
Intended status: Standards Track
Expires: January 11, 2021

S. Fries
H. Brockhaus
Siemens
E. Lear
Cisco Systems

July 10, 2020

Support of asynchronous Enrollment in BRSKI (BRSKI-AE) draft-ietf-anima-brski-async-enroll-00

Abstract

This document describes enhancements of bootstrapping a remote secure key infrastructure (BRSKI) to also operate in domains featuring no or only timely limited connectivity between involved components. It addresses connectivity to backend services supporting enrollment like a Public Key Infrastructure (PKI) and also to the connectivity between pledge and registrar. For this it enhances the use of authenticated self-contained objects in BRSKI also for request and distribution of deployment domain specific device certificates. The defined approach is agnostic regarding the utilized enrollment protocol allowing the application of existing and potentially new certificate management protocols.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 11, 2021.

Table of Contents

1. Introduction

BRSKI as defined in [I-D.ietf-anima-bootstrapping-keyinfra] specifies a solution for secure zero-touch (automated) bootstrapping of devices (pledges) in a target deployment domain. This includes the discovery of network elements in the deployment domain, time synchronization, and the exchange of security information necessary to establish trust between a pledge and the domain and to adopt a pledge as new network and application element. Security information about the deployment domain, specifically the deployment domain certificate (domain root certificate), is exchanged utilizing voucher objects as defined in [RFC8366]. These vouchers are authenticated self-contained (signed) objects, which may be provided online (synchronous) or offline (asynchronous) via the domain registrar to the pledge and originate from a Manufacturer's Authorized Signing Authority (MASA). The MASA signed voucher contains the target domain certificate and can be verified by the pledge due to the possession of a manufacturer root certificate. It facilitates the enrollment of the pledge in the deployment domain and is used to establish trust from the pledge to the domain.

For the enrollment of devices BRSKI relies on EST [RFC7030] to request and distribute deployment domain specific device certificates. EST in turn relies on a binding of the certification request to an underlying TLS connection between the EST client and the EST server. According to BRSKI the domain registrar acts as EST server and is also acting as registration authority (RA) or local registration authority (LRA). The binding to TLS is used to protect the exchange of a certification request (for an LDevID certificate) and to provide data origin authentication to support the authorization decision for processing the certification request. The TLS connection is mutually authenticated and the client side authentication utilizes the pledge's manufacturer issued device certificate (IDevID certificate). This approach requires an on-site availability of a local asset or inventory management system performing the authorization decision based on tuple of the certification request and the pledge authentication using the IDevID certificate, to issue a domain specific certificate to the pledge. The reason bases on the EST server (the domain registrar) terminating the security association with the pledge and thus the local binding between the certification request and the authentication of the pledge. This type of enrollment utilizing an online connection to the PKI is considered as synchronous enrollment.

For certain use cases on-site support of a RA/CA component and/or an asset management is not available and rather provided by an operator's backend and may be provided timely limited or completely through offline interactions. This may be due to higher security requirements for operating the certification authority. The authorization of a certification request based on an asset management in this case will not / can not be performed on-site at enrollment time. Enrollment, which cannot be performed in a (timely) consistent fashion is considered as asynchronous enrollment in this document. It requires the support of a store and forward functionality of certification request together with the requester authentication information. This enables processing of the request at a later point in time. A similar situation may occur through network segmentation, which is utilized in industrial systems to separate domains with different security needs. Here, a similar requirement arises if the communication channel carrying the requester authentication is terminated before the RA/CA authorization handling of the certification request. If a second communication channel is opened to forward the certification request to the issuing RA/ CA, the requester authentication information needs to be retained and ideally bound to the certification request. This uses case is independent from timely limitations of the first use case. For both cases, it is assumed that the requester authentication information is utilized in the process of authorization of a certification request. There are different options to perform store and forward of certification requests including the requester authentication information:

- Providing a trusted component (e.g., an LRA) in the deployment domain, which stores the certification request combined with the requester authentication information (based on the IDevID) and potentially the information about a successful proof of possession (of the corresponding private key) in a way prohibiting changes to the combined information. Note that the assumption is that the information elements may not be cryptographically bound together. Once connectivity to the backend is available, the trusted component forwards the certification request together with the requester information (authentication and proof of possession) to the off-site PKI for further processing. It is assumed that the off-site PKI in this case relies on the local pledge authentication result and thus performs the authorization and issues the requested certificate. In BRSKI the trusted component may be the EST server residing co-located with the registrar in the deployment domain.

- Utilization of authenticated self-contained objects for the enrollment, binding the certification request and the requester authentication in a cryptographic way. This approach reduces the necessary trust in a domain component to storage and delivery.

Unauthorized modifications of the requester information (request and authentication) can be detected during the verification of the authenticated self-contained object. An example for such an object is a signed CMS wrapped object (as the voucher).

This targets environments, in which connectivity to a PKI is only temporary or not directly available, by specifying support for handling authenticated self-contained objects for enrollment. As it is intended to enhance BRSKI it is named BRSKI-AE, where AE stands for asynchronous enrollment. As BRSKI, BRSKI-AE results in the pledge storing a X.509 root certificate sufficient for verifying the domain registrar / proxy identity (LDevID CA Certificate) as well as an domain specific X.509 device certificate (LDevID EE certificate).

Based on the proposed approach, a second set of scenarios can be addressed, in which the pledge has no direct communication path to the domain registrar, e.g., due to no network connectivity or a different technology stack as the domain registrar, but is considered to be managed by the domain registrar regarding the pledge domain credentials. For this, an additional component is introduced acting as an agent for the pledge towards the domain registrar, e.g., a commissioning tool. In contrast to BRSKI here the credentials may be pushed to the pledge instead of the pull approach taken by BRSKI.

The goal is to enhance BRSKI to either allow other existing certificate management protocols supporting authenticated self-contained objects to be applied or to allow other types of encoding for the certificate management information exchange. This is addressed by

- enhancing the well-known URI approach with additional path' for the utilized enrollment protocol.
- defining a certificate waiting indication and handling, if the certifying component is (temporarily) not available.
- allowing to utilize credentials different from the pledge's IDevID to establish a connection to the domain registrar.

Note that in contrast to BRSKI, BRSKI-AE assumes support of multiple enrollment protocols on the infrastructure side, allowing the pledge manufacturer to select the most appropriate. Thus, BRSKI-AE can be applied for both, asynchronous and synchronous enrollment.

2. 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 [RFC2119].

This document relies on the terminology defined in [I-D.ietf-anima-bootstrapping-keyinfra]. The following terms are defined additionally:

CA: Certification authority, issues certificates.

RA: Registration authority, an optional system component to which a CA delegates certificate management functions such as authorization checks.

LRA: Local registration authority, an optional RA system component with proximity to end entities.

IED: Intelligent Electronic Device (in essence a pledge).

on-site: Describes a component or service or functionality available in the target deployment domain.

off-site: Describes a component or service or functionality available in an operator domain different from the target deployment domain. This may be a central side, to which only a temporarily connection is available, or which is in a different administrative domain.

asynchronous communication: Describes a timely interrupted communication between an end entity and a PKI component.

synchronous communication: Describes a timely uninterrupted communication between an end entity and a PKI component.

authenticated self-contained object: Describes an object, which is cryptographically bound to the IDevID EE credential of a pledge. The binding is assumed to be provided through a digital signature using the corresponding private key of the IDevID to wrap the actual object. Note that depending on the availability of a LDevID EE credential, the binding may also be achieved using corresponding private key of the LDevID. This can be utilized in for instance in the context of an initial certification request or a certificate update.

3. Scope of solution

3.1. Supported environment

This solution is intended to be used in domains with limited support of on-site PKI services and comprises use cases in which:

- there is no registration authority available in the deployment domain. The connectivity to the backend RA may only be temporarily available. A local store and forward device is used for the communication with the backend services.

- authoritative actions of a LRA are limited and may not comprise authorization of certification requests of pledges. Final authorization is done at the RA residing in the backend operator domain.

- the target deployment domain already uses a certificate management approach that shall be reused to be consistent throughout the life cycle.

In addition, the solution is intended to be applicable in domains in which pledges have no direct connection to the domain registrar, but are expected to be managed by the registrar. This can be motivated by pledges featuring a different technology stack or by pledges without an existing connection to the domain registrar during onboarding.

3.2. Application Examples

The following examples are intended to motivate the support of different enrollment approaches in general and asynchronous enrollment specifically, by introducing industrial applications cases, which could leverage BRSKI as such but also require support of asynchronous operation as intended with BRSKI-AE.

3.2.1. Rolling stock

Rolling stock or railroad cars contain a variety of sensors, actuators, and controller, which communicate within the railroad car but also exchange information between railroad cars building a train or with a backend. These devices are typically unaware of backend connectivity. Managing certificates may be done during maintenance cycles of the railroad car, but can already be prepared during operation. The preparation may comprise the generation of certification requests by the components, which are collected and forwarded for processing once the railroad car is connected to the operator backend. The authorization of the certification request is

then done based on the operator's asset/inventory information in the backend.

3.2.2. Building automation

In building automation a use case can be described by a detached building or the basement of a building equipped with sensor, actuators, and controllers connected, but with only limited or no connection to the centralized building management system. This limited connectivity may be during the installation time but also during operation time. During the installation in the basement, a service technician collects the necessary information from the basement network and provides them to the central building management system, e.g., using a laptop or even a mobile phone to transport the information. This information may comprise parameters and settings required in the operational phase of the sensors/actuators, like a certificate issued by the operator to authenticate against other components and services.

The collected information may be provided by a domain registrar already existing in the installation network. In this case connectivity to the backend PKI may be facilitated by the service technician's laptop. Contrary, the information can also be collected from the pledges directly and provided to a domain registrar deployed in the main network. In this cases connectivity to the domain registrar may be facilitated by the service technician's laptop.

3.2.3. Substation automation

In substation automation a control center typically hosts PKI services to issue certificates for Intelligent Electronic Devices (IED)s in a substation. Communication between the substation and control center is done through a proxy/gateway/DMZ, which terminates protocol flows. Note that NERC CIP-005-5 [NERC-CIP-005-5] requires inspection of protocols at the boundary of a security perimeter (the substation in this case). In addition, security management in substation automation assumes central support of different enrollment protocols to facilitate the capabilities of IEDs from different vendors. The IEC standard IEC62351-9 [IEC-62351-9] specifies the mandatory support of two enrollment protocols, SCEP [I-D.gutmann-scep] and EST [RFC7030] for the infrastructure side, while the IED must only support one of the two.

3.2.4. Electric vehicle charging infrastructure

For the electric vehicle charging infrastructure protocols have been defined for the interaction between the electric vehicle (EV) and the charging point (e.g., ISO 15118-2 [ISO-IEC-15118-2]) as well as

between the charging point and the charging point operator (e.g. OCPP [OCPP]). Depending on the authentication model, unilateral or mutual authentication is required. In both cases the charging point authenticates uses an X.509 certificate to authenticate in the context of a TLS connection between the EV and the charging point. The management of this certificate depends (beyond others) on the selected backend connectivity protocol. Specifically, in case of OCPP it is intended as single communication protocol between the charging point and the backend carrying all information to control the charging operations and maintain the charging point itself. This means that the certificate management is intended to be handled in-band of OCPP. This requires to be able to encapsulate the certificate management exchanges in a transport independent way. Authenticated self-containment will ease this by allowing the transport without a separate communication protocol. For the purpose of certificate management CMP [RFC4210] is intended to be used.

3.2.5. Infrastructure isolation policy

This refers to any case in which network infrastructure is normally isolated from the Internet as a matter of policy, most likely for security reasons. In such a case, limited access to external PKI resources will be allowed in carefully controlled short periods of time, for example when a batch of new devices are deployed, but impossible at other times.

3.2.6. Less operational security in the deployment domain

The registration point performing the authorization of a certificate request is a critical PKI component and therefore implicates higher operational security than other components utilizing the issued certificates for their security features. CAs may also demand higher security in the registration procedures. Especially the CA/Browser forum currently increases the security requirements in the certificate issuance procedures for publicly trusted certificates. There may be the situation that the deployment domain does not offer enough security to operate a registration point and therefore wants to transfer this service to a backend.

4. Requirement discussion and mapping to solution elements

For the requirements discussion it is assumed that the domain registrar receiving a certification request as authenticated self-contained object is not the authorization point for this certification request. If the domain registrar is the authorization point, BRSKI can be used directly. Note that BRSKI-AE could also be used in this case.

Based on the intended deployment environment described in Section 3.1 and the motivated application examples described in Section 3.2 the following base requirements are derived to support authenticated self-contained objects as container carrying the certification request and further information to support asynchronous operation.

At least the following properties are required:

- Proof of Possession: utilizing the private key corresponding to the public key contained in the certification request.

- Proof of Identity: utilizing an existing IDevID credential bound to the certification request. Certificate updates may utilize the LDevID credential.

Solution examples (not complete) based on existing technology are provided with the focus on existing IETF documents:

- Certification request objects: Certification requests are structures protecting only the integrity of the contained data providing a proof-of-private-key-possession for locally generated key pairs. Examples for certification requests are:

  * PKCS#10 [RFC2986]: Defines a structure for a certification request. The structure is signed to ensure integrity protection and proof of possession of the private key of the requester that corresponds to the contained public key.

  * CRMF [RFC4211]: Defines a structure for the certification request message. The structure supports integrity protection and proof of possession, through a signature generated over parts of the structure by using the private key corresponding to the contained public key.

  Note that the integrity of the certification request is bound to the public key contained in the certification request by performing the signature operation with the corresponding private key. In the considered application examples, this is not sufficient and needs to be bound to the existing credential of the pledge (IDevID) additionally. This binding supports the authorization decision for the certification request through the provisioning of a proof of identity. The binding of data origin authentication to the certification request may be delegated to the protocol used for certificate management.

- Proof of Identity options: The certification request should be bound to an existing credential (here IDevID) to enable a proof of identity and based on it an the authorization of the certification

request. The binding may be realized through a security options in an underlying transport protocol if the authorization of the the certification request is done at the next communication hop. Alternatively, this binding can be done by a wrapping signature employing an existing credential (initial: IDevID, renewal: LDevID). This requirement is addressed by existing enrollment protocols in different ways, for instance:

* EST [RFC7030]: Utilizes PKCS#10 to encode the certification request. The Certificate Signing Request (CSR) may contain a binding to the underlying TLS by including the tls-unique value in the self-signed CSR structure. The tls-unique value is one result of the TLS handshake. As the TLS handshake is performed mutually authenticated and the pledge utilized its IDevID for it, the proof of identity can be provided by the binding to the TLS session. This is supported in EST using simpleenroll. To avoid the binding to the underlying authentication in the transport layer EST offers the support of a wrapping the CSR with an existing certificate by using fullcmc.

* SCEP [I-D.gutmann-scep]: Provides the option to utilize either an existing secret (password) or an existing certificate to protect the CSR based on SCEP Secure Message Objects using CMS wrapping ([RFC5652]). Note that the wrapping using an existing IDevID credential in SCEP is referred to as renewal. SCEP therefore does not rely on the security of an underlying transport.

* CMP [RFC4210] Provides the option to utilize either an existing secret (password) or an existing certificate to protect the PKIMessage containing the certification request. The certification request is encoded utilizing CRMF. PKCS#10 is optionally supported. The proof of identity of the PKIMessage containing the certification request can be achieved by using IDevID credentials to calculate a signature over the header and the body of the PKIMessage utilizing the protectionAlg signaled in the PKIMessage header and the PKIProtection carrying the actual signature value. CMP therefore does not rely on the security of an underlying transport.

* CMC [RFC5272] Provides the option to utilize either an existing secret (password) or an existing certificate to protect the certification request (either in CRMF or PKCS#10) based on CMS [RFC5652]). Here a FullCMCRequest can be used, which allows signing with an existing IDevID credential to provide a proof of identity. CMC therefore does not rely on the security of an underlying transport.

Note that besides the already existing enrollment protocols there ongoing work in the ACE WG to define an encapsulation of EST in OSCORE to result in a TLS independent way of protecting EST. This approach [I-D.selander-ace-coap-est-oscore] may be considered as further variant.

5. Architectural Overview and Communication Exchanges

To support asynchronous enrollment, the base system architecture defined in BRSKI [I-D.ietf-anima-bootstrapping-keyinfra] is enhanced to facilitate the two target use cases.

- Use case 1 (PULL case): the pledge requests certificates from a PKI operated off-site via the domain registrar.

- Use case 2 (PUSH/PULL case): allows delayed (delegated) onboarding using a pledge-agent instead a direct connection to the domain registrar. The communication model between pledge-agent and pledge depends on the specified interface and may use a PULL or PUSH approach. This interaction in terms of a protocol specification is out of scope of this document.

Note that the terminology PUSH and PULL relates to the pledge behavior. In PULL the pledge requests data objects as in BRSKI, while in the PUSH case the pledge may be provisioned with the necessary data objects. The pledge-agent as it represents the pledge always acts in a PULL mode to the domain registrar. Both use cases are described in the next subsections. They utilize the existing BRSKI architecture elements as much as possible. Necessary enhancements to support authenticated self-contained objects for certificate enrollment are kept on a minimum to ensure reuse of already defined architecture elements and interactions.

For the authenticated self-contained objects used for the certification request, BRSKI-AE relies on the defined message wrapping mechanisms of the enrollment protocols stated in Section 4 above.

5.1. Use Case 1: Support of off-site PKI service

One assumption of BRSKI-AE is that the authorization of a certification request is performed based on an authenticated self-contained object, binding the certification request to the authentication using the IDevID. This supports interaction with off-site or off-line PKI (RA/CA) components. In addition, the authorization of the certification request may not be done by the domain registrar but by a PKI residing in the backend of the domain operator (off-site) as described in Section 3.1. This leads to

changes in the placement or enhancements of the logical elements as shown in Figure 1.

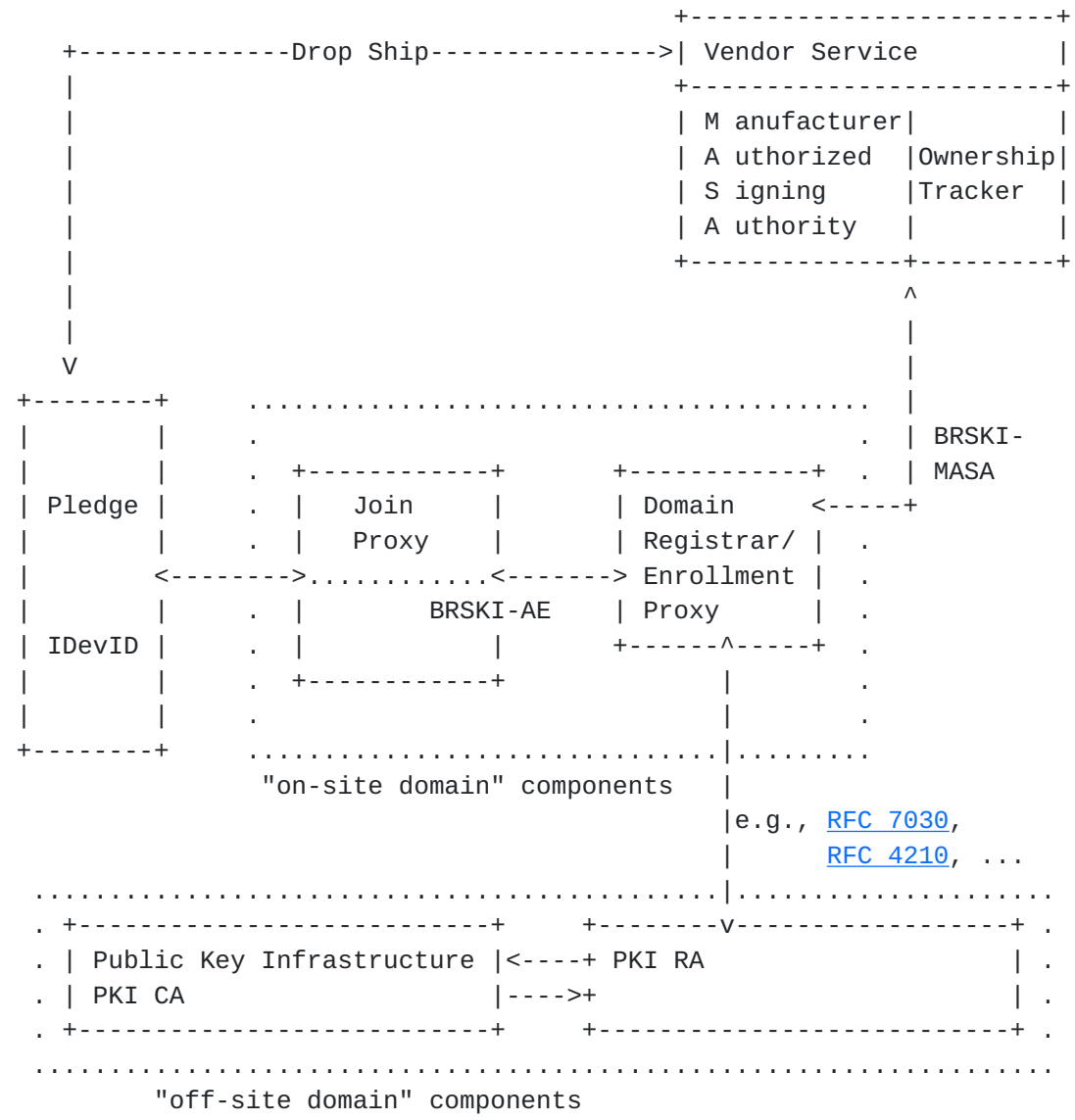


Figure 1: Architecture overview using off-site PKI components

The architecture overview in Figure 1 utilizes the same logical elements as BRSKI but with a different placement in the deployment architecture for some of the elements. The main difference is the placement of the PKI RA/CA component, which is performing the authorization decision for the certification request message. It is placed in the off-site domain of the operator (not the deployment site directly), which may have no or only temporary connectivity to the deployment or on-site domain of the pledge. This is to underline the authorization decision for the certification request in the

backend rather than on-site. The following list describes the components in the deployment domain:

- Join Proxy: same functionality as described in BRSKI.

- Domain Registrar / Enrollment Proxy: In general the domain registrar proxy has a similar functionality regarding the imprinting of the pledge in the deployment domain to facilitate the communication of the pledge with the MASA and the PKI. Different is the authorization of the certification request. BRSKI-AE allows to perform this in the operators backend (off-site), and not directly at the domain registrar.

  * Voucher exchange: The voucher exchange with the MASA via the domain registrar is performed as described in BRSKI [I-D.ietf-anima-bootstrapping-keyinfra] .

  * Certificate enrollment: For the pledge enrollment the domain registrar in the deployment domain supports the adoption of the pledge in the domain based on the voucher request. Nevertheless, it may not have sufficient information for authorizing the certification request. If the authorization is done in the off-site domain, the domain registrar forwards the certification request to the RA to perform the authorization. The domain registrar in this acts as an enrollment proxy or local registration authority. It is also able to handle the case having temporarily no connection to an off-site PKI by storing the certification request and forwarding it to the RA upon regaining connectivity. As authenticated self-contained objects are used, it requires an enhancement of the domain registrar. This is done by supporting alternative enrollment approaches (protocol options, protocols, encoding) by enhancing the addressing scheme to communicate with the domain registrar (see Section 5.1.5) and also by providing a discover scheme to allow the pledge to enumerate the supported enrollment options (see Section 5.3).

The following list describes the vendor related components/service outside the deployment domain:

- MASA: general functionality as described in BRSKI. Assumption that the interaction with the MASA may be synchronous (voucher request with nonce) or asynchronous (voucher request without nonce).

- Ownership tracker: as defined in BRSKI.

The following list describes the operator related components/service operated in the backend:

- PKI RA: Performs certificate management functions (validation of certification requests, interaction with inventory/asset management for authorization of certification requests, etc.) for issuing, updating, and revoking certificates for a domain as a centralized infrastructure for the domain operator. The inventory (asset) management may be a separate component or integrated into the RA directly.

- PKI CA: Performs certificate generation by signing the certificate structure provided in the certification request.

Based on BRSKI and the architectural changes the original protocol flow is divided into three phases showing commonalities and differences to the original approach as depicted in the following.

- Discovery phase (same as BRSKI)

- Voucher exchange with deployment domain registrar (same as BRSKI).

- Enrollment phase (changed to accompany the application of authenticated self-contained objects).

5.1.1. Behavior of a pledge

The behavior of a pledge as described in [I-D.ietf-anima-bootstrapping-keyinfra] is kept with one exception. After finishing the imprinting phase (4) the enrollment phase (5) is performed with a method supporting authenticated self-contained objects. Using EST with simpleenroll cannot be applied here, as it binds the pledge authentication with the existing IDevID to the transport channel (TLS) rather than to the certification request object directly. This authentication in the transport layer is not visible / verifiable at the authorization point in the off-site domain. Section 6 discusses potential enrollment protocols and options applicable.

5.1.2. Pledge - Registrar discovery and voucher exchange

The discovery phase is applied as specified in [I-D.ietf-anima-bootstrapping-keyinfra].

5.1.3. Registrar - MASA voucher exchange

The voucher exchange is performed as specified in [I-D.ietf-anima-bootstrapping-keyinfra].

5.1.4. Pledge - Registrar - RA/CA certificate enrollment

As stated in Section 4 the enrollment shall be performed using an authenticated self-contained object providing:

- Proof of Possession: utilizing the private key corresponding to the public key contained in the certification request.

- Proof of Identity: utilizing the existing IDevID credential to generate a signature of the initial certification request. Certificate updates may utilize the LDevID credential.

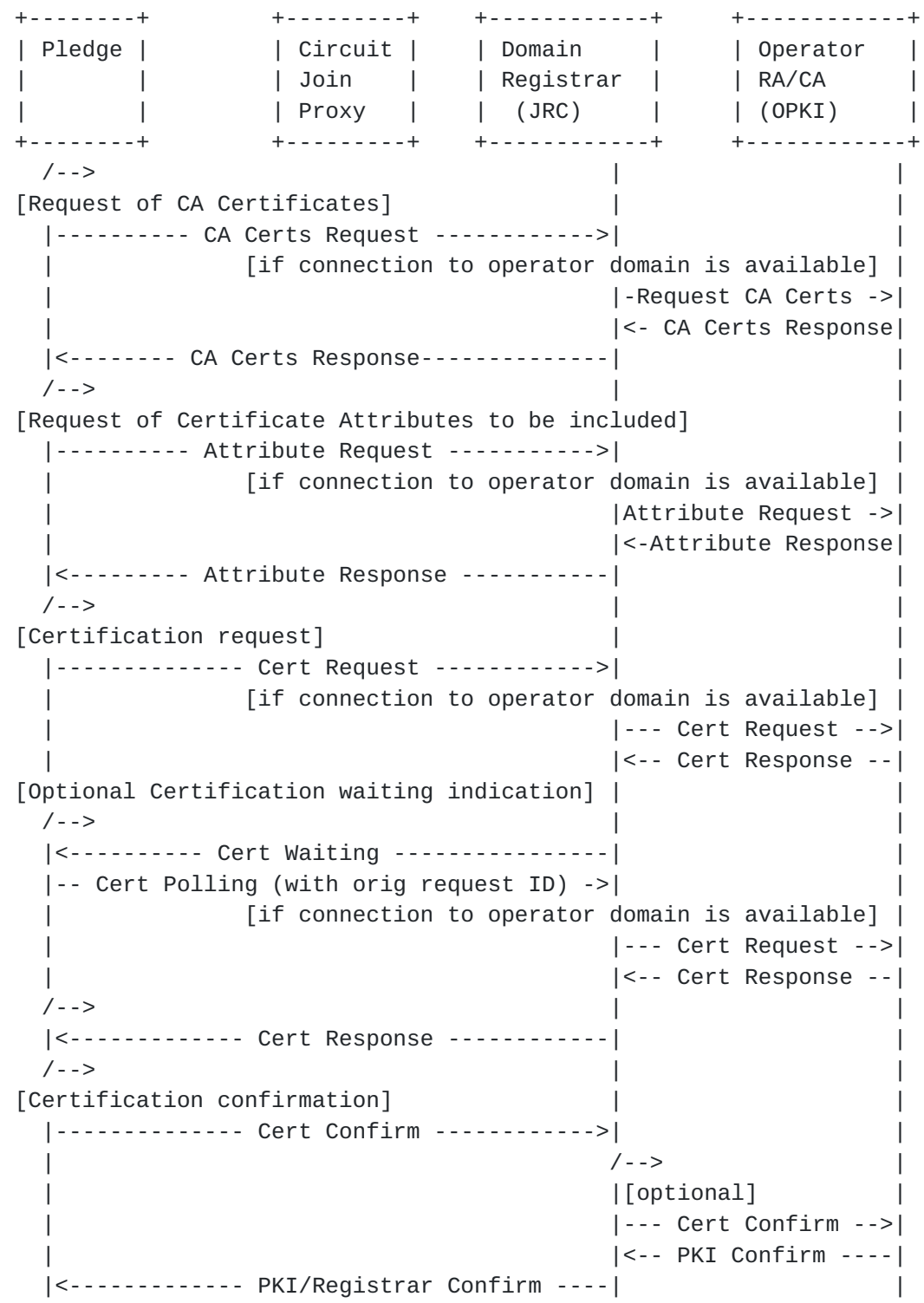


Figure 2: Certificate enrollment

The following list provides an abstract description of the flow depicted in Figure 2.

- CA Cert Request: The pledge SHOULD request the full distribution of CA Certificates. This ensures that the pledge has the complete set of current CA certificates beyond the pinned-domain-cert.

- CA Cert Response: Contains at least one CA certificate of the issuing CA.

- Attribute Request: Typically, the automated bootstrapping occurs without local administrative configuration of the pledge. Nevertheless, there are cases, in which the pledge may also include additional attributes specific to the deployment domain into the certification request. To get these attributes in advance, the attribute request SHOULD be used.

- Attribute Response: Contains the attributes to be included in the certification request message.

- Cert Request: Depending on the utilized enrollment protocol, this certification request contains the authenticated self-contained object ensuring both, proof-of-possession of the corresponding private key and proof-of-identity of the requester.

- Cert Response: certification response message containing the requested certificate and potentially further information like certificates of intermediary CAs on the certification path.

- Cert Waiting: waiting indication for the pledge to retry after a given time. For this a request identifier is necessary. This request identifier may bei either part of the enrollment protocol or build based on the certification request.

- Cert Polling: querying the registrar, if the certificate request was already processed; can be answered either with another Cert Waiting, or a Cert Response.

- Cert Confirm: confirmation message from pledge after receiving and verifying the certificate.

- PKI/Registrar Confirm: confirmation message from PKI/registrar about reception of the pledge's certificate confirmation.

[RFC Editor: please delete] /*

Open Issues:

- Description of certificate waiting and retries.

- Message exchange description is expected to be done by the utilized enrollment protocol based on the addressing scheme (see also Section 6.

- Handling of certificate/PKI confirmation message between pledge and domain registrar and PKI (treated optional?).

*/

5.1.5. Addressing Scheme Enhancements

BRSKI-AE requires enhancements to the addressing scheme defined in [I-D.ietf-anima-bootstrapping-keyinfra] to accommodate the additional handling of authenticated self-contained objects for the certification request. As this is supported by different enrollment protocols, they can be directly employed (see also Section 6). For the support of different enrollment options at the domain registrar, the addressing approach of BRSKI using a "/.well-known" tree from [RFC5785] is enhanced.

The current addressing scheme in BRSKI for the client certificate request function during the enrollment is using the definition from EST [RFC7030], here on the example on simple enroll: "/.well-known/est/simpleenroll" This approach is generalized to the following notation: "/.well-known/enrollment-protocol/request" in which enrollment-protocol may be an already existing protocol or a newly defined approach. Note that enrollment is considered here as a sequence of at least a certification request and a certification response. In case of existing enrollment protocols the following notation is used proving compatibility to BRSKI:

- enrollment-protocol: references either EST [RFC7030] as in BRSKI or CMP, CMC, SCEP, or newly defined approaches as alternatives. Note: the IANA registration of the well-known URI is expected to be done by the enrollment protocol. For CMP a lightweight profile is defined, which provides the definition of the well-known URI in Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

- request: depending on the utilized enrollment protocol, the request describes the required operation at the registrar side. Enrollment protocols are expected to define the request endpoints as done by existing protocols (see also Section 6).

5.2. Use Case 2: pledge-agent

To support mutual trust establishment of pledges, not directly connected to the domain registrar, a similar approach is applied as discussed for the use case 1. It relies on the exchange of

authenticated self-contained objects (the voucher request/response objects as known from BRSKI and the certification request/response objects as introduced by BRSKI-AE). This allows independence from the protection provided by the underlying transport.

In contrast to BRSKI, the exchange of these objects is performed with the help of a pledge-agent, supporting the interaction of the pledge with the domain registrar. It may be an integrated functionality of a commissioning tool. This leads to enhancements of the logical elements in the BRSKI architecture as shown in Figure 3. The pledge-agent provides an interface to the pledge to enable creation or consumption of required data objects, which are exchanged with the domain registrar. Moreover, the addition of the pledge-agent also influences the sequences for the data exchange between the pledge and the domain registrar described in [I-D.ietf-anima-bootstrapping-keyinfra]. The general goal for the pledge-agent application is the reuse of already defined endpoints on the domain registrar side. The behavior of the endpoint may need to be adapted.

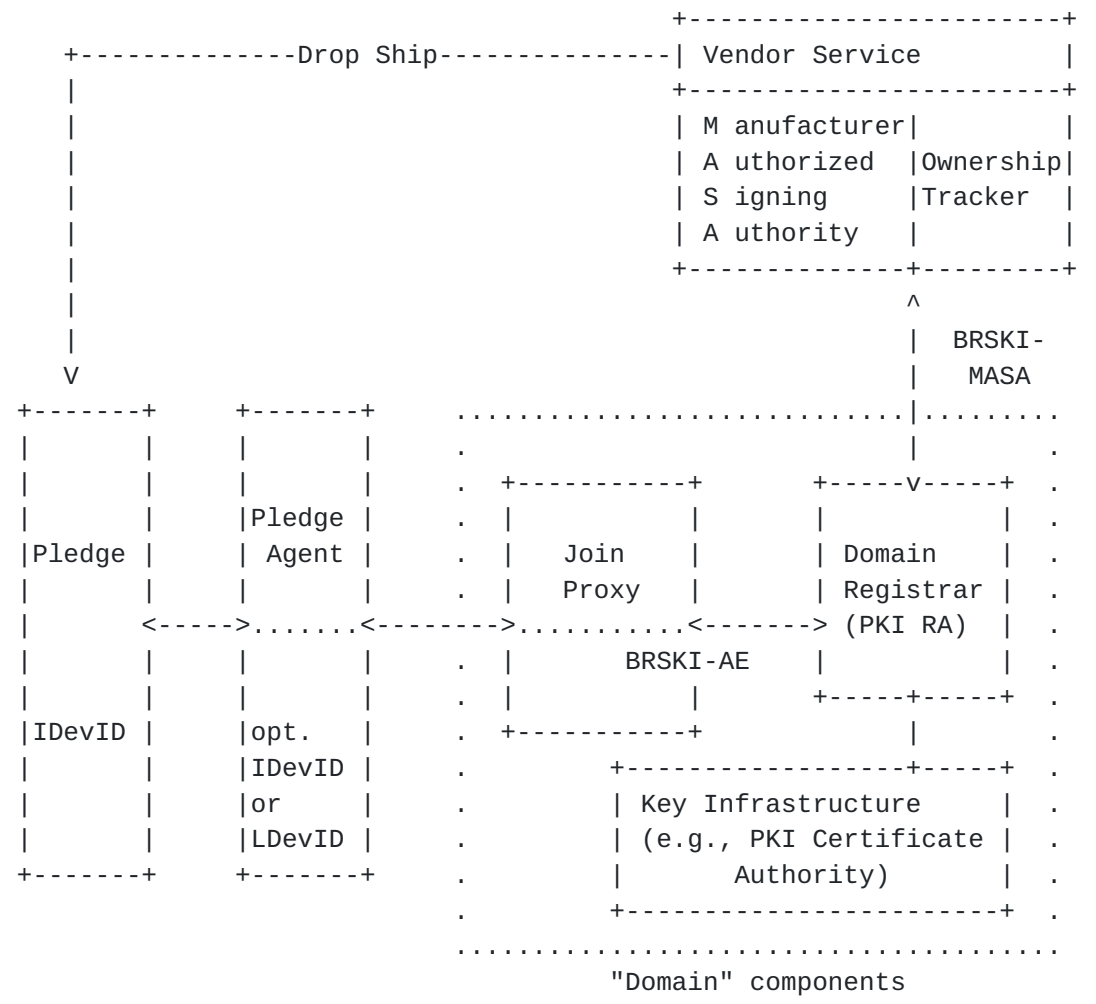


Figure 3: Architecture overview using a pledge-agent

The architecture overview in Figure 3 utilizes the same logical elements as BRSKI with the addition of the pledge-agent. The pledge-agent, may originate from the pledge manufacturer and may have either an own IDevID credential issued by the manufacturer or an LDevID issued already by the deployment (on-site) domain.

For authentication towards the domain registrar, the pledge-agent may use the IDevID or LDevID credentials, which are verified by the domain registrar as part of the TLS establishment. The provisioning of this credential to the pledge-agent is out of scope for this specification. Alternatively, the domain registrar may authenticate the user operating the pledge-agent to perform authorization of pledge onboarding. Examples for such a user level authentication are the application of HTTP authentication or the usage of SAML tokens or the application of a user related certificates in the TLS handshake or other. If the pledge-agent utilizes a certificate, the domain

registrar must be able to verify the certificate by possessing the corresponding root certificate.

The following list describes the components in the deployment domain:

- Pledge: The pledge is expected to communicate with the pledge-agent for providing the necessary data objects for onboarding. The exact protocol used between the pledge and the pledge-agent is out of scope for this document but may consider: If the pledge is triggered/PUSHED by the pledge-agent, it becomes a callee. There are some differences to BRSKI:

  * Discovery of the domain registrar will be omitted as the pledge is expected to be triggered by the pledge-agent.

  * The pledge-agent is expected to provide an option to trigger the onboarding by pushing data objects to the pledge.

  * Order of exchanges in the call flow is different as the pledge-agent collects both voucher request objects and certification request objects at once.

  * The data objects utilized are the same objects already applied in use case 1 Section 5.1.

- Pledge-Agent: provides a communication path to exchange data objects between the pledge and the domain registrar. The pledge-agent facilitates situations, in which the domain registrar is not directly reachable by the pledge, either due to a different technology stack or due to missing connectivity (e.g., if the domain registrar resides in the cloud and the pledge has no connectivity, yet). The pledge-agent in this cases can easily collect voucher request objects and certification request objects from one or multiple pledges at once and perform a bulk onboarding based on the collected data. The pledge-agent may be configured with the domain registrar information or may use the discovery mechanism.

- Join Proxy: same functionality as described in BRSKI.

- Domain Registrar: In general the domain registrar fulfills the same functionality regarding the onboarding of the pledge in the deployment domain by facilitating the communication of the pledge with the MASA and the PKI. In contrast to BRSKI, the domain registrar does not interact with a pledge directly but through the pledge-agent. This prohibits a pledge authentication using its IDevID during TLS establishment towards the registrar. If the pledge-agent has an IDevID or is already possessing a LDevID valid

in the deployment domain, it is expected to use this authentication towards the domain registrar.

The manufacturer provided components/services (MASA and Ownership tracker) are used as defined in BRSKI.

5.2.1. Behavior of a pledge

The behavior of a pledge as described for use case 1 Section 5.1 is basically kept regarding the generation of voucher request/response objects and certificate request/response objects. Due to the use of the pledge-agent, the interaction with the domain registrar is changed as shown in Figure 4.

The interaction of the pledge with the pledge-agent in terms of utilized protocols or discovery options is out of scope of this document. This document concentrates on the exchanged data objects between the pledge and the domain registrar via the pledge-agent.

The pledge-agent should be able to authenticate the pledge-agent either based on security mechanisms as part of the communication channel between the pledge and the pledge-agent or based on the data (request) objects.

The pledge-agent should provide the proximity-registrar-cert to the pledge to enable embedding in the voucher request object. The registrar certificate may be configured at the pledge-agent or may be fetched by the pledge-agent based on the TLS connection establishment with the domain registrar.

The pledge interacts with the pledge-agent, to generate a voucher request object (VouReq) and a certification request object (CR), which are provided to the domain registrar through the pledge-agent.

The pledge shall generate the voucher request object as described in [I-D.ietf-anima-bootstrapping-keyinfra] and provide this information to the pledge-agent.

After the voucher request exchange the pledge will be triggered by to generate a certification request object. For this, the pledge-agent may have been pre-configured with the certification request attributes, that it may provide to the pledge. The certification request is generated as authenticated self-signed object, which assures proof of possession of the private key corresponding to the contained public key in the certification request as well as a proof of identity, based on the IDevID of the pledge. This is done as described for use case 1 Section 5.1.

5.2.2. Behavior of a pledge-agent

The pledge-agent is a new component in the BRSKI context. It provides connectivity between the pledge and the domain registrar and utilizes the endpoints already specified in [I-D.ietf-anima-bootstrapping-keyinfra]. The pledge-agent is expected to interact with the pledge independent of the domain registrar. As stated before, data exchange is only defined based on the data objects, which are the voucher request/response objects and the certification request/response objects. The transport mechanism is out of scope here. This changes the general interaction as shown in Figure 4.

The pledge-agent may have an own IDevID or a deployment domain issued LDevID to be utilized in the TLS communication establishment towards the domain registrar. Note that the pledge-agent may also be used without client side authentication if no suitable credential is available on transport layer. As BRSKI-AE utilizes authenticated self-contained data objects, which bind the pledge authentication (proof of identity) directly to the objects (voucher request and certification request), the TLS client authentication may be neglected. This is a deviation from the BRSKI approach in which the pledge's IDevID credential is used to perform TLS client authentication. According to [I-D.ietf-anima-bootstrapping-keyinfra] section 5.3, the domain registrar performs the pledge authorization for onboarding within his domain based on the provided voucher request.

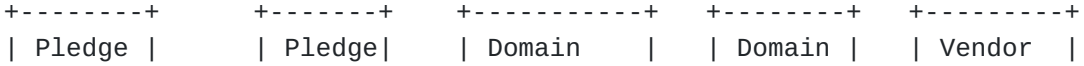
5.2.3. Registrar discovery

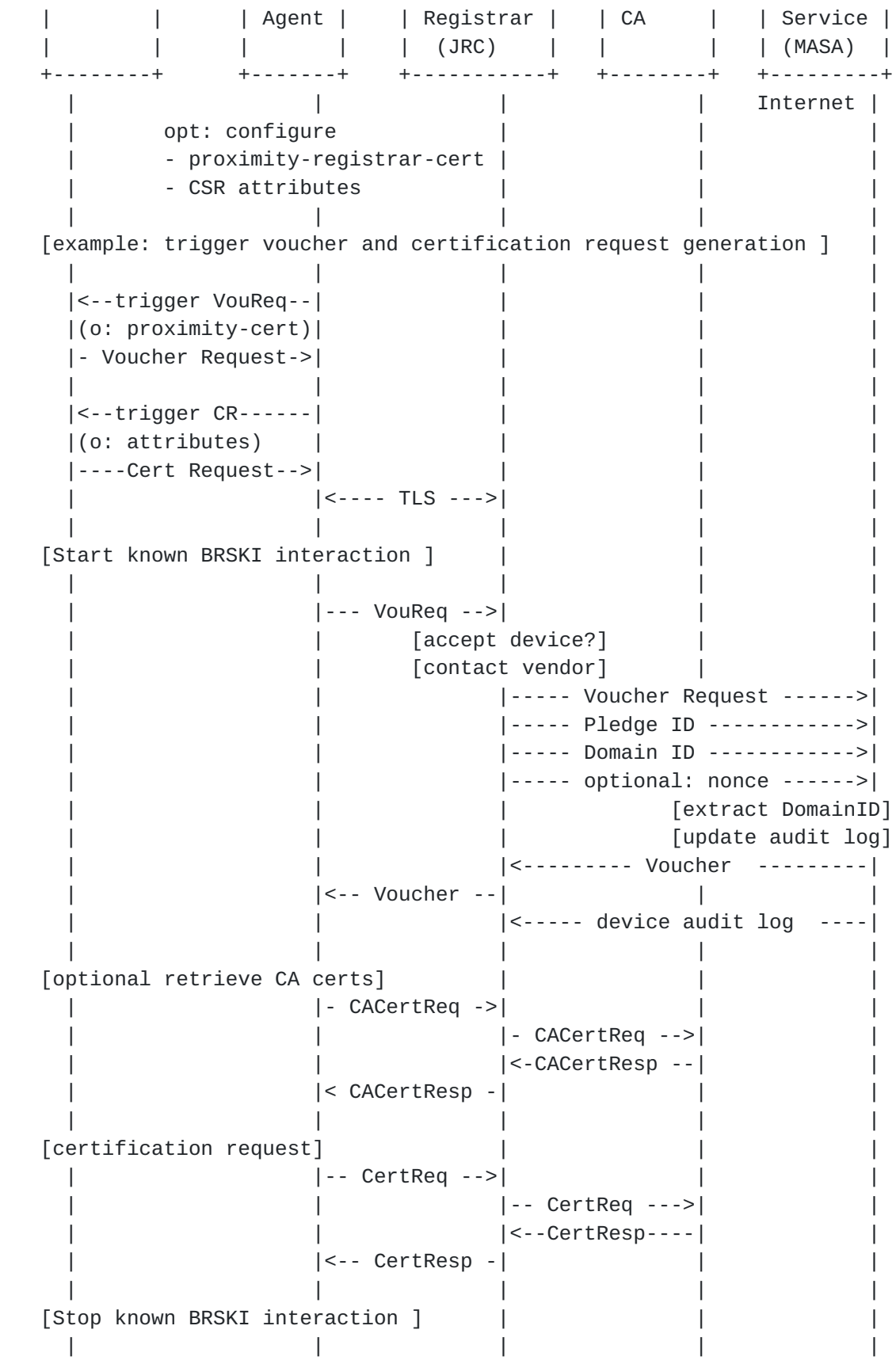
The discovery phase may be applied as specified in [I-D.ietf-anima-bootstrapping-keyinfra] with the deviation that it is done between the pledge-agent and the domain registrar. Alternatively, the domain registrar may be configured in the pledge-agent.

The discovery of the pledge-agent by the pledge belongs to the communication between the two instances and is out of scope for this specification.

5.2.4. Handling voucher request and certification requests

The BRSKI-AE exchange of voucher requests and certification requests utilizes authenticated self-contained objects independent of transport protection.




| | | Agent | | Registrar | | CA | | Service |
| | | | | (JRC) | | | | (MASA) |
Internet
opt: configure
- proximity-registrar-cert
- CSR attributes
[example: trigger voucher and certification request generation]
<--trigger VouReq--
(o: proximity-cert)
- Voucher Request->
<--trigger CR------
(o: attributes)
----Cert Request-->
<---- TLS --->
[Start known BRSKI interaction]
--- VouReq -->
[accept device?]
[contact vendor]
----- Voucher Request ------>
----- Pledge ID ------------>
----- Domain ID ------------>
----- optional: nonce ------>
[extract DomainID]
[update audit log]
<--------- Voucher ---------
<-- Voucher --
<----- device audit log ----
[optional retrieve CA certs]
- CACertReq ->
- CACertReq -->
<-CACertResp --
< CACertResp -
[certification request]
-- CertReq -->
-- CertReq --->
<--CertResp----
<-- CertResp -
[Stop known BRSKI interaction]

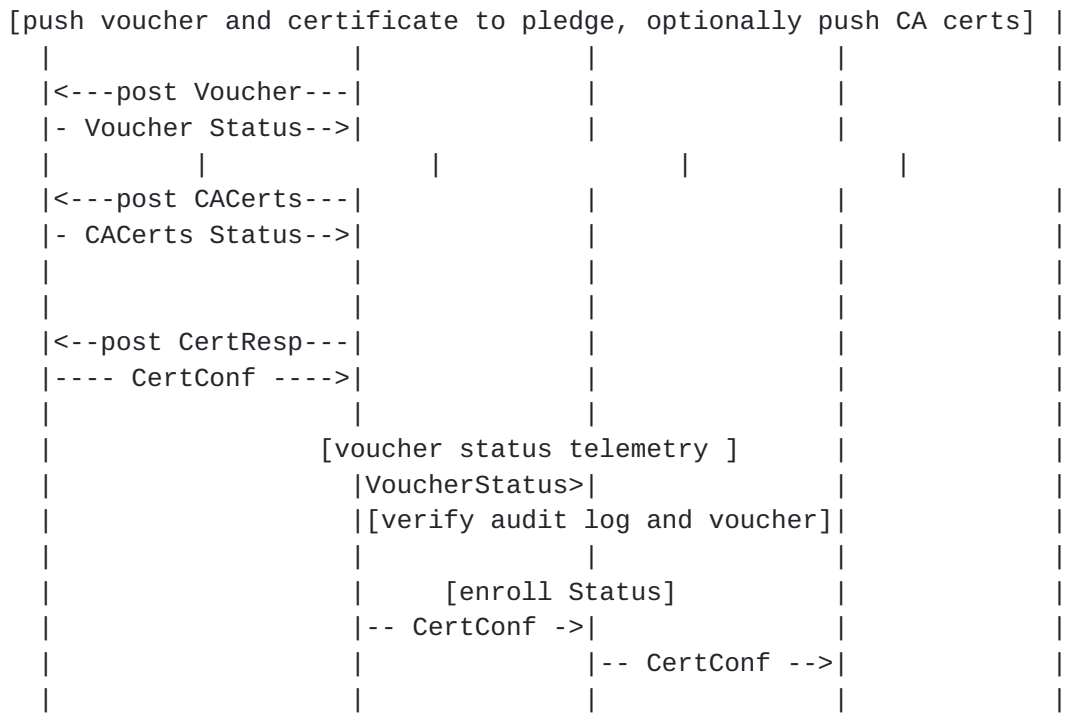


Figure 4: Request handling of the pledge using a pledge-agent

As shown in Figure 4 the pledge-agent collects the voucher request and certification request objects from a pledge. As the pledge-agent (e.g., as part of a commissioning tool) is intended to work between the pledge and the domain registrar, a collection of requests from multiple pledges is possible, allowing a bulk onboarding of multiple pledges using the connection between the pledge-agent and the domain registrar.

The information exchange between the pledge-agent and the domain registrar resembles the exchanges between the pledge and the domain registrar from BRSKI with one exception. As authenticated self-contained objects are used consequently, the authentication of the pledge-agent to the domain registrar may be neglected. Note that this allows to employ simple applications as pledge-agent. The authentication of the pledge-agent is recommended if it is desired to perform the onboarding with an authorized pledge-agent or to support advanced auditing in case a user based authentication is done. As stated above, the authentication may be realized by device (IDevID or LDevID) or user related credentials in the context of the TLS handshake, HTTP based authentication, SAML tokens or other.

[RFC Editor: please delete] /* to be discussed: Description on how the registrar makes the decision if he is connected with pledge directly or with a pledge-agent. This may result in a case statement (client side authentication in TLS, user authentication above TLS,

etc.) for the TLS connection establishment in the original BRSKI document in section 5.1 */

Once the pledge-agent has finished the exchanges with the domain registrar to get the voucher and the certificate object, it can close the TLS connection to the domain registrar and provide the objects to the pledge(s). The transport of the objects to the pledge is out of scope. The content of the response objects is defined through the voucher [RFC8366] and the certificate [RFC5280].

5.3. Discovery of supported enrollment options at domain registrar

Well-know URIs for different endpoints on the domain registrar are already defined as part of the base BRSKI specification. In addition, this document utilizes well-known URIs to allow for alternative enrollment options at the domain registrar. The discovery of supported endpoints will therefore provide the information to the pledge, how to contact the domain registrar.

Querying the registrar, the pledge will get a list of potential endpoints supported by the domain registrar. To allow for a BRSKI specific discovery of endpoints/resources, this document specifies a new URI for the discovery as "/.well-known/brski".

Performing a GET on "/.well-known/brski" to the default port returns a set of links to endpoints available from the server. In addition to the link also the expected format of the data object is provided as content type (ct).

The following provides an illustrative example for a domain registrar supporting different options for EST as well as CMP to be used in BRSKI-AE. The listing contains the supported endpoints for the onboarding:

REQ: GET /.well-known/brski

```
RES: Content
  </brski/voucherrequest>,ct=voucher-cms+json
  </brski/voucher_status>,ct=json
  </brski/requestauditlog>,ct=json
  </brski/enrollstatus>,ct=json
  </est/cacerts>;ct=pkcs7-mime
  </est/cacerts>;ct=pkcs7-mime
  </est/simpleenroll>;ct=pkcs7-mime
  </est/simplereenroll>;ct=pkcs7-mime
  </est/fullcmc>;ct=pkcs7-mime
  </est/serverkeygen>;ct= pkcs7-mime
  </est/csrattrs>;ct=pkcs7-mime
  </cmp/initialization>;ct=pkixcmp
  </cmp/certification>;ct=pkixcmp
  </cmp/keyupdate>;ct=pkixcmp
  </cmp/p10>;ct=pkixcmp
  </cmp/getCAcert>;ct=pkixcmp
  </cmp/getCSRparam>;ct=pkixcmp
```

[RFC Editor: please delete] /*

Open Issues:

- Change path from /est to /brski to be protocol agnostic

- Define new well-know URI as above or reuse core approach as described in RFC 6690 with /.well-known/core and the already defined functionality?

- In addition to the current content types, we may specify that the response provide information about different content types as multiple values. This would allow to further adopt the encoding of the objects exchanges (ASN.1, JSON, CBOR, ...).

*/

6. Example mappings to existing enrollment protocols

This sections maps the requirements to support proof of possession and proof of identity to selected existing enrollment protocols. Note that that the work in the ACE WG described in [I-D.selander-ace-coap-est-oscore] may be considered here as well, as it also addresses the encapsulation of EST in a way to make it independent from the underlying TLS using OSCORE resulting in an authenticated self-contained object.

6.1. EST Handling

When using EST [RFC7030], the following constrains should be considered:

- Proof of possession is provided by using the specified PKCS#10 structure in the request.

- Proof of identity is achieved by signing the certification request object, which is only supported when the /fullcmc endpoint is used. This contains sufficient information for the RA to make an authorization decision on the received certification request. Note: EST references CMC [RFC5272] for the definition of the Full PKI Request. For proof of identity, the signature of the SignedData of the Full PKI Request would be calculated using the IDevID credential of the pledge.

- [RFC Editor: please delete] /* TBD: in this case the binding to the underlying TLS connection is not be necessary. */

- When the RA is not available, as per [RFC7030] Section 4.2.3, a 202 return code should be returned by the Registrar. The pledge in this case would retry a simpleenroll with a PKCS#10 request. Note that if the TLS connection is teared down for the waiting time, the PKCS#10 request would need to be rebuild if it contains the unique identifier (tls_unique) from the underlying TLS connection for the binding.

- [RFC Editor: please delete] /* TBD: clarification of retry for fullcmc is necessary as not specified in the context of EST */

6.2. Lightweight CMP Handling

Instead of using CMP [RFC4210], this specification refers to the lightweight CMP profile [I-D.ietf-lamps-lightweight-cmp-profile], as it restricts the full featured CMP to the functionality needed here. For this, the following constrains should be observed:

- For proof of possession, the defined approach in Lightweight CMP section 5.1.1 (based on CRMF) and 5.1.5 based on PCKS#10 should be supported.

- Proof of identity can be provided by using the signatures to protect the certificate request message as outlined in section 4.2.

- When the RA/CA is not available, a waiting indication should be returned in the PKIStatus by the Registrar. The pledge in this

case would retry using the PollReqContent with a request identifier certReqId provided in the initial CertRequest message as specified in section 6.1.4 with delayed enrollemnt.

7. IANA Considerations

This document requires the following IANA actions:

[RFC Editor: please delete] /* to be done: IANA consideration to be included for the defined namespaces in Section 5.1.5 and Section 5.3 . */

8. Privacy Considerations

[RFC Editor: please delete] /* to be done: clarification necessary */

9. Security Considerations

[RFC Editor: please delete] /* to be done: clarification necessary */

10. Acknowledgments

We would like to thank the various reviewers for their input, in particular Brian E. Carpenter, Giorgio Romanenghi, Oskar Camenzind, for their input and discussion on use cases and call flows.

11. References

11.1. Normative References

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-41 (work in progress), April 2020.

[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>.

[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>.

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

11.2. Informative References

[I-D.gutmann-scep]
Gutmann, P., "Simple Certificate Enrolment Protocol", draft-gutmann-scep-16 (work in progress), March 2020.

[I-D.ietf-lamps-lightweight-cmp-profile]
Brockhaus, H., Fries, S., and D. Oheimb, "Lightweight CMP Profile", draft-ietf-lamps-lightweight-cmp-profile-01 (work in progress), March 2020.

[I-D.selander-ace-coap-est-oscore]
Selander, G., Raza, S., Furuhed, M., Vucinic, M., and T. Claeys, "Protecting EST Payloads with OSCORE", draft-selander-ace-coap-est-oscore-03 (work in progress), March 2020.

[IEC-62351-9]
International Electrotechnical Commission, "IEC 62351 - Power systems management and associated information exchange - Data and communications security - Part 9: Cyber security key management for power system equipment", IEC 62351-9 , May 2017.

[ISO-IEC-15118-2]
International Standardization Organization / International Electrotechnical Commission, "ISO/IEC 15118-2 Road vehicles - Vehicle-to-Grid Communication Interface - Part 2: Network and application protocol requirements", ISO/IEC 15118 , April 2014.

[NERC-CIP-005-5]
North American Reliability Council, "Cyber Security - Electronic Security Perimeter", CIP 005-5, December 2013.

[OCPP] Open Charge Alliance, "Open Charge Point Protocol 2.0 (Draft)", April 2018.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, DOI 10.17487/RFC2986, November 2000, <https://www.rfc-editor.org/info/rfc2986>.

[RFC4210] Adams, C., Farrell, S., Kause, T., and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", RFC 4210, DOI 10.17487/RFC4210, September 2005, <https://www.rfc-editor.org/info/rfc4210>.

[RFC4211] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, DOI 10.17487/RFC4211, September 2005, <https://www.rfc-editor.org/info/rfc4211>.

[RFC5272] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC)", RFC 5272, DOI 10.17487/RFC5272, June 2008, <https://www.rfc-editor.org/info/rfc5272>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, DOI 10.17487/RFC5785, April 2010, <https://www.rfc-editor.org/info/rfc5785>.

Appendix A. History of changes [RFC Editor: please delete]

From individual version 03 -> IETF draft 00:

- Inclusion of discovery options of enrollment endpoints at the domain registrar based on well-known endpoints in Section 5.3 as replacement of section 5.1.3 in the individual draft. This is intended to support both use cases in the document. An illustrative example is provided.

- Missing details provided for the description and call flow in pledge-agent use case Section 5.2, e.g. to accommodate distribution of CA certificates.

- Updated CMP example in Section 6 to use lightweight CMP instead of CMP, as the draft already provides the necessary /.well-known endpoints.

- Requirements discussion moved to separate section in Section 4. Shortened description of proof of identity binding and mapping to existing protocols.

- Removal of copied call flows for voucher exchange and registrar discovery flow from [I-D.ietf-anima-bootstrapping-keyinfra] in Section 5.1 to avoid doubling or text or inconsistencies.

- Reworked abstract and introduction to be more crisp regarding the targeted solution. Several structural changes in the document to have a better distinction between requirements, use case description, and solution description as separate sections. History moved to appendix.

From individual version 02 -> 03:

- Update of terminology from self-contained to authenticated self-contained object to be consistent in the wording and to underline the protection of the object with an existing credential. Note that the naming of this object may be discussed. An alternative name may be attestation object.

- Simplification of the architecture approach for the initial use case having an offsite PKI.

- Introduction of a new use case utilizing authenticated self-contain objects to onboard a pledge using a commissioning tool containing a pledge-agent. This requires additional changes in the BRSKI call flow sequence and led to changes in the introduction, the application example,and also in the related BRSKI-AE call flow.

- Update of provided examples of the addressing approach used in BRSKI to allow for support of multiple enrollment protocols in Section 5.1.5.

From individual version 01 -> 02:

- Update of introduction text to clearly relate to the usage of IDevID and LDevID.

- Definition of the addressing approach used in BRSKI to allow for support of multiple enrollment protocols in Section 5.1.5. This section also contains a first discussion of an optional discovery mechanism to address situations in which the registrar supports more than one enrollment approach. Discovery should avoid that the pledge performs a trial and error of enrollment protocols.

- Update of description of architecture elements and changes to BRSKI in Section 5.

- Enhanced consideration of existing enrollment protocols in the context of mapping the requirements to existing solutions in Section 4 and in Section 6.

From individual version 00 -> 01:

- Update of examples, specifically for building automation as well as two new application use cases in Section 3.2.

- Deletion of asynchronous interaction with MASA to not complicate the use case. Note that the voucher exchange can already be handled in an asynchronous manner and is therefore not considered further. This resulted in removal of the alternative path the MASA in Figure 1 and the associated description in Section 5.

- Enhancement of description of architecture elements and changes to BRSKI in Section 5.

- Consideration of existing enrollment protocols in the context of mapping the requirements to existing solutions in Section 4.

- New section starting Section 6 with the mapping to existing enrollment protocols by collecting boundary conditions.

Authors' Addresses

Steffen Fries
Siemens AG
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: steffen.fries@siemens.com
URI: http://www.siemens.com/

Hendrik Brockhaus
Siemens AG
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: hendrik.brockhaus@siemens.com
URI: http://www.siemens.com/

Eliot Lear
Cisco Systems
Richtistrasse 7
Wallisellen CH-8304
Switzerland

Phone: +41 44 878 9200
Email: lear@cisco.com