ANIMA WG
Internet-Draft
Intended status: Informational
Expires: May 4, 2017

M. Pritikin
Cisco
M. Richardson
SSW
M. Behringer
S. Bjarnason
Cisco
K. Watsen
Juniper Networks
October 31, 2016

Bootstrapping Remote Secure Key Infrastructures (BRSKI)
draft-ietf-anima-bootstrapping-keyinfra-04

Abstract

This document specifies automated bootstrapping of a remote secure key infrastructure (BRSKI) using vendor installed X.509 certificate, in combination with a vendor authorized service on the Internet. Bootstrapping a new device can occur using a routable address and a cloud service, or using only link-local connectivity, or on limited/disconnected networks. Support for lower security models, including devices with minimal identity, is described for legacy reasons but not encouraged. Bootstrapping is complete when the cryptographic identity of the new key infrastructure is successfully deployed to the device but the established secure connection can be used to deploy a locally issued certificate to the device as well.

Table of Contents

1. Introduction

To literally "pull yourself up by the bootstraps" is an impossible action. Similarly the secure establishment of a key infrastructure without external help is also an impossibility. Today it is accepted that the initial connections between nodes are insecure, until key distribution is complete, or that domain-specific keying material is pre-provisioned on each new device in a costly and non-scalable manner. This document describes a zero-touch approach to bootstrapping an entity by securing the initial distribution of key material using third-party issued X.509 certificates and cryptographically signed "vouchers" issued by a new form of cloud service.

The two sides of an association being bootstrapped authenticate each other and then determine appropriate authorization. This process is described as four distinct steps between the existing domain and the device, or "pledge", being added:

- Pledge authentication: "Who is this? What is its identity?"
- Pledge authorization: "Is it mine? Do I want it? What are the chances it has been compromised?"
- Domain authentication: "What is this domain's claimed identity?"
- Domain authorization: "Should I join it?"

A precise answer to these questions can not be obtained without leveraging an established key infrastructure(s). The pledge's decisions are made according to verified communication with a trusted third-party. The domain's decisions are made by comparing the pledge's authenticated identity against domain information such as a configured list of purchased devices supplimented by information provided by a trusted third-party. The third-party is not required to provide sales channel ownership tracking nor is it required to authenticate the domain.

Optimal security is achieved with X.509 certificates on each Pledge, accompanied by a third-party (e.g., vendor, manufacturer or integrator) Internet based service for verification. Bootstrapping concepts run to completion with less requirements, but are then less secure. A domain can choose to accept lower levels of security when a trusted third-party is not available so that bootstrapping proceeds even at the risk of reduced security. Only the domain can make these decisions based on administrative input and known behavior of the pledge.

The result of bootstrapping is that a domain specific key infrastructure is deployed. Since X.509 PKI certificates are used for identifying the pledge, and the public key of the domain identity is leveraged during communications with an Internet based service, which is itself authenticated using HTTPS, bootstrapping of a domain specific Public Key Infrastructure (PKI) is described. Sufficient agility to support bootstrapping alternative key infrastructures (such as symmetric key solutions) is considered although no such alternate key infrastructure is described.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The following terms are defined for clarity:

DomainID: The domain identity is the 160-bit SHA-1 hash of the BIT STRING of the subjectPublicKey of the domain trust anchor that is stored by the Domain CA. This is consistent with the Certification Authority subject key identifier (Section 4.2.1.2 [RFC5280]) of the Domain CA's self signed root certificate. (A string value bound to the Domain CA's self signed root certificate subject and issuer fields is often colloquially used as a humanized identity value but during protocol discussions the more exact term as defined here is used).

drop ship: The physical distribution of equipment containing the "factory default" configuration to a final destination. In zero-touch scenarios there is no staging or pre-configuration during drop-ship.

imprint: The process where a device obtains the cryptographic key material to identify and trust future interactions with a network. This term is taken from Konrad Lorenz's work in biology with new ducklings: during a critical period, the duckling would assume that anything that looks like a mother duck is in fact their mother. An equivalent for a device is to obtain the fingerprint of the network's root certification authority certificate. A device that imprints on an attacker suffers a similar fate to a duckling that imprints on a hungry wolf. Securely imprinting is a primary focus of this document.[imprinting]. The analogy to Lorenz's work was first noted in [Stajano99theresurrecting].

enrollment: The process where a device presents key material to a network and acquires a network specific identity. For example when a certificate signing request is presented to a certification authority and a certificate is obtained in response.

Pledge: The prospective device, which has an identity installed by a third-party (e.g., vendor, manufacturer or integrator).

Voucher A signed statement from the MASA service that indicates to a Pledge the cryptographic identity of the Registrar it should trust. There are different types of vouchers depending on how that trust verified.

Audit Voucher: A voucher from the MASA service that indicates that the bootstrapping event has been successfully logged. The Registrar is primarily responsible for verifying the logs and ensuring domain network security.

Ownership Voucher: A voucher from the MASA service that indicates the explicit owner identity. The MASA is primarily responsible for tracking ownership using out-of-band sales channel integration (the definition of which is out-of-scope of this document). It is defined in [I-D.ietf-netconf-zerotouch].

Domain: The set of entities that trust a common key infrastructure trust anchor. This includes the Proxy, Registrar, Domain Certificate Authority, Management components and any existing entity that is already a member of the domain.

Domain CA: The domain Certification Authority (CA) provides certification functionalities to the domain. At a minimum it provides certification functionalities to a Registrar and stores the trust anchor that defines the domain. Optionally, it certifies all elements.

Registrar: A representative of the domain that is configured, perhaps autonomically, to decide whether a new device is allowed to join the domain. The administrator of the domain interfaces with a Registrar to control this process. Typically a Registrar is "inside" its domain.

Proxy: A domain entity that helps the pledge join the domain. A Proxy facilitates communication for devices that find themselves in an environment where they are not provided connectivity until after they are validated as members of the domain. The pledge is unaware that they are communicating with a proxy rather than directly with a Registrar.

MASA Service: A third-party Manufacturer Authorized Signing Authority (MASA) service on the global Internet. The MASA provides a repository for audit log information concerning privacy protected bootstrapping events. It does not track ownership.

Ownership Tracker An Ownership Tracker service on the global internet. The Ownership Tracker uses business processes to accurately track ownership of all devices shipped against domains that have purchased them. Although optional this component allows vendors to provide additional value in cases where their sales and distribution channels allow for accurately tracking of such ownership.

IDevID An Initial Device Identity X.509 certificate installed by the vendor on new equipment. The [IDevID] certificate format is the primary example. In particular the X.509 certificate needs to contain the device's serial number in a well known location in order to perform white list operations and in order to extract it for inclusion in messages to the MASA service. The subject field's DN encoding MUST include the "serialNumber" attribute with the device's unique serial number.

1.2. Scope of solution

Questions have been posed as to whether this solution is suitable in general for Internet of Things (IoT) networks. This depends on the capabilities of the devices in question. The terminology of [RFC7228] is best used to describe the boundaries.

The entire solution described in this document is aimed in general at non-constrained (i.e. class 2+) devices operating on a non-Challenged network. The entire solution described here is not intended to be useable as-is by constrained devices operating on challenged networks (such as 802.15.4 LLNs).

In many target applications, the systems involved are large router platforms with multi-gigabit inter-connections, mounted in controlled access data centers. But this solution is not exclusive to the large, it is intended to scale to thousands of devices located in hostile environments, such as ISP provided CPE devices which are drop-shipped to the end user. The situation where an order is fulfilled from distributed warehouse from a common stock and shipped directly to the target location at the request of the domain owner is explicitly supported. That stock ("SKU") could be provided to a number of potential domain owners, and the eventual domain owner will not know a-priori which device will go to which location.

The bootstraping process can take minutes to complete depending on the network infrastructure and device processing speed. The network communication itself is not optimized for speed; the discovery process allows for the Pledge to avoid broadcasting for privacy reasons. This protocol is not intended for low latency handoffs.

Specifically, there are protocol aspects described here which might result in congestion collapse or energy-exhaustion of intermediate battery powered routers in an LLN. Those types of networks SHOULD NOT use this solution. These limitations are predominately related to the large credential and key sizes required for device authentication. Defining symmetric key techniques that meet the operational requirements is out-of-scope but the underlying protocol

operations (TLS handshake and signing structures) have sufficient algorithm agility to support such techniques when defined.

The imprint protocol described here could, however, be used by non-energy constrained devices joining a non-constrained network (for instance, smart light bulbs are usually mains powered, and speak 802.11). It could also be used by non-constrained devices across a non-energy constrained, but challenged network (such as 802.15.4).

The use of an IDevID that is consistant with [IDevID] allows for alignment with 802.1X network access control methods which could need to complete before bootstrapping can be initiated. This document presumes that network access control has either already occured, is not required, or is integrated by the proxy and registrar in such a way that the device itself does not need to be aware of the details. Further integration is not in scope.

Some aspects are in scope for constrained devices on challenged networks: the certificate contents, and the process by which the four questions above are resolved is in scope. It is simply the actual on-the-wire imprint protocol which is likely inappropriate.

1.3. Trust bootstrap

The imprint protocol results in a secure relationship between a domain Registrar and the Pledge. If the new device is sufficiently constrained that the ACE protocol should be leveraged for operation, (see [I-D.ietf-ace-actors]), and the domain registrar is also the Client Authorization Server or the Authorization Server, then it may be appropriate to use this secure channel to exchange ACE tokens.

2. Architectural Overview

The logical elements of the bootstrapping framework are described in this section. Figure 1 provides a simplified overview of the components. Each component is logical and may be combined with other components as necessary.

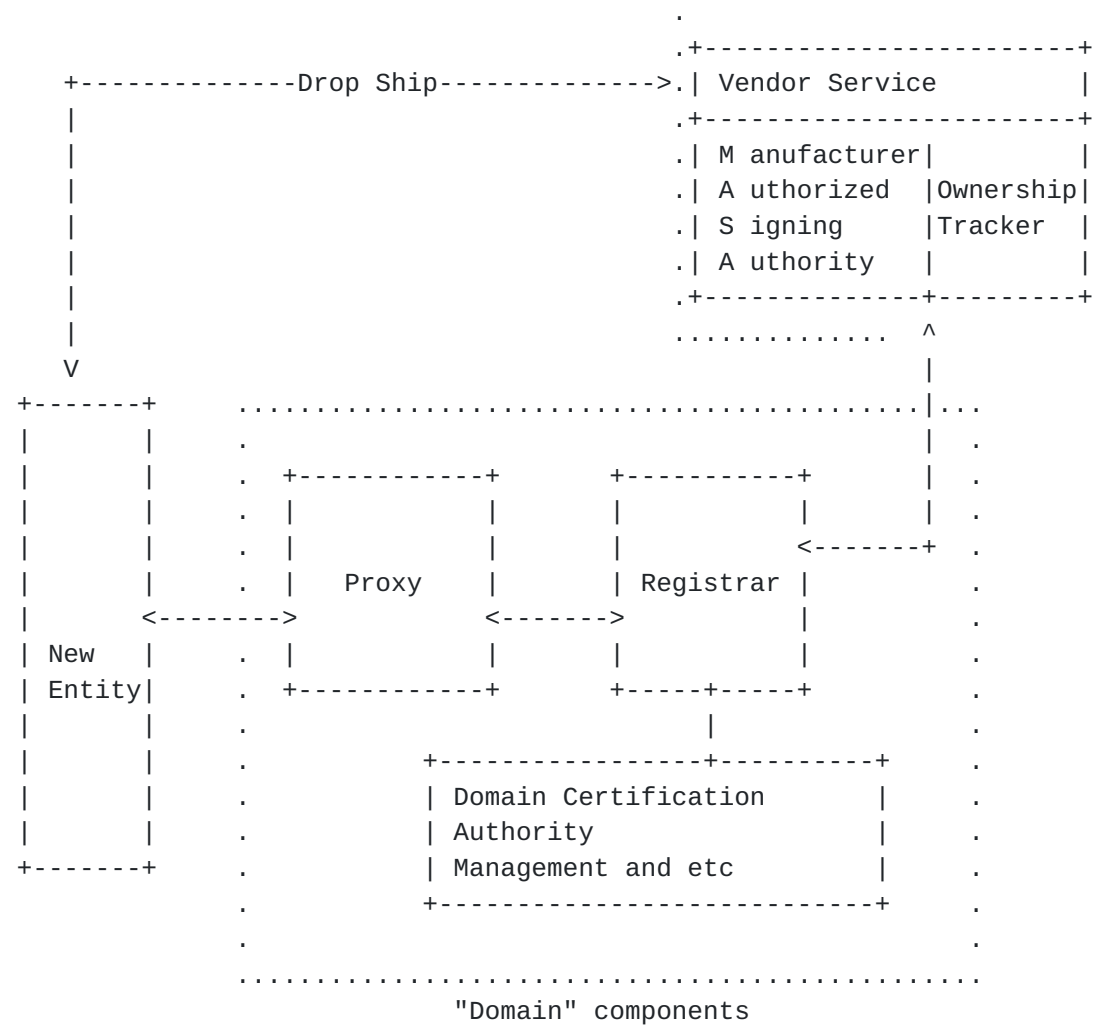


Figure 1

We assume a multi-vendor network. In such an environment there could be a MASA or Ownership Tracker for each vendor that supports devices following this document's specification, or an integrator could provide a MASA service for all devices. It is unlikely that an integrator could provide Ownership Tracking services for multiple vendors.

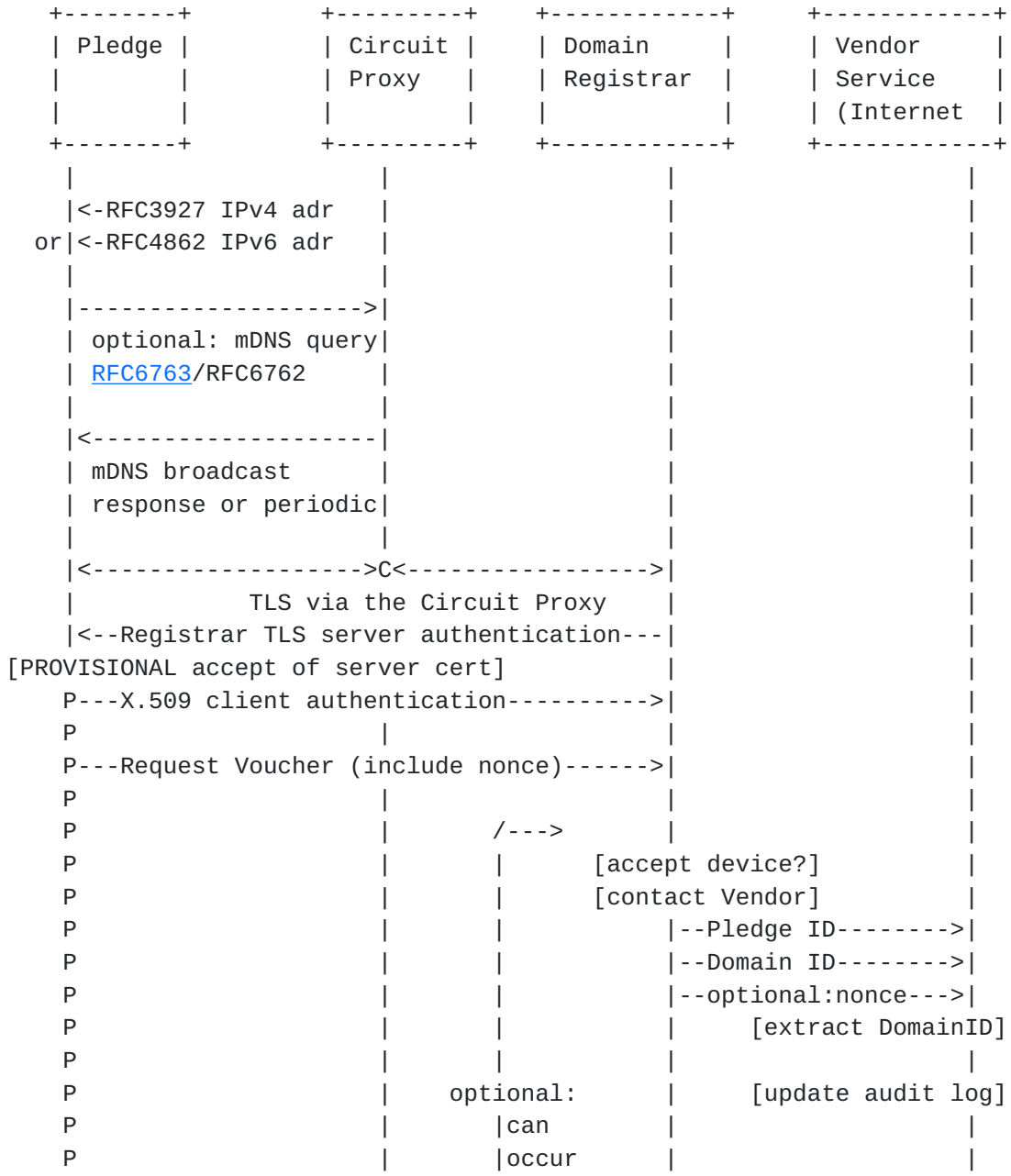
This document describes a secure zero-touch approach to bootstrapping a key infrastructure; if certain devices in a network do not support this approach, they can still be bootstrapped manually. Although manual deployment is not scalable and is not a focus of this document the necessary mechanisms are called out in this document to ensure such edge conditions are covered by the architectural and protocol models.

3. Functional Overview

Entities behave in an autonomic fashion. They discover each other and autonomically bootstrap into a key infrastructure delineating the autonomic domain. See [RFC7575] for more information.

This section details the state machine and operational flow for each of the main three entities. The pledge, the domain (primarily a Registrar) and the MASA service.

A representative flow is shown in Figure 2:



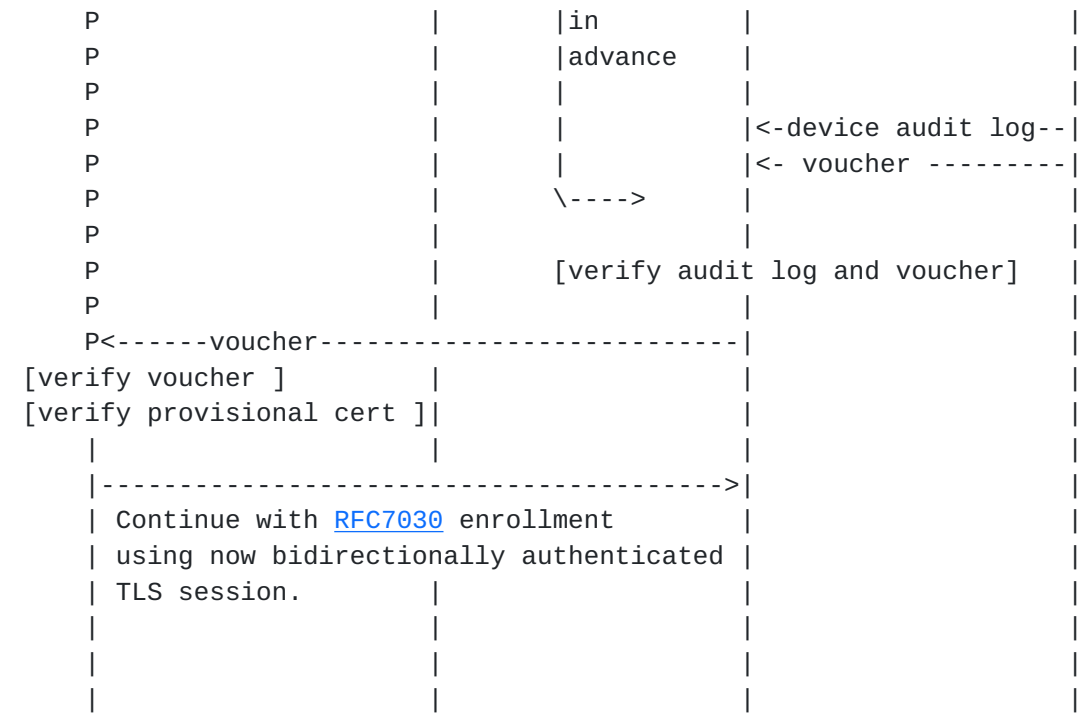


Figure 2

3.1. Behavior of a Pledge

A pledge that has not yet been bootstrapped attempts to find a local domain and join it. A pledge MUST NOT automatically initiate bootstrapping if it has already been configured or is in the process of being configured.

States of a pledge are as follows:

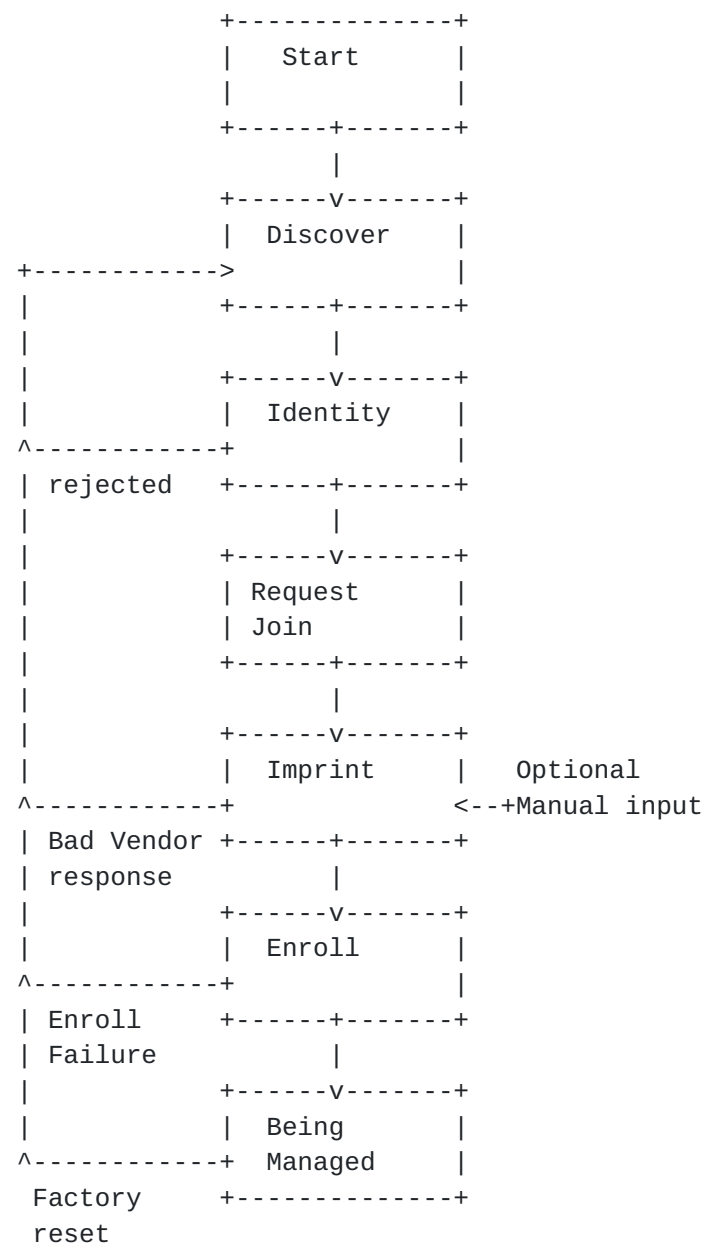


Figure 3

State descriptions for the pledge are as follows:

1. Discover a communication channel to a Registrar.

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

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

4. Imprint on the Registrar. This requires verification of the vendor service provided "Audit" or "Ownership" Voucher. Either of these responses contains sufficient information for the pledge to complete authentication of a Registrar. (The pledge can now finish authentication of the Registrar TLS server certificate)

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

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

The following sections describe each of these steps in more detail.

3.1.1. Discovery

The result of discovery is a logical communication with a Registrar, through a Proxy. The Proxy is transparent to the Pledge but is always assumed to exist.

To discover the Registrar the Pledge performs the following actions:

a. MUST: Obtains a local address using either IPv4 or IPv6 methods as described in [RFC4862] IPv6 Stateless Address AutoConfiguration or [RFC3927] Dynamic Configuration of IPv4 Link-Local Addresses. The Plege MAY obtain an IP address via DHCP [RFC2131]. The DHCP provided parameters for the Domain Name System can be used to perform step (d) DNS operations if all local discovery attempts fail (see below).

b. MUST: Performs DNS-based Service Discovery [RFC6763] over Multicast DNS [RFC6762] searching for the service "_bootstrapks._tcp.local.". To prevent unaccceptable levels of network traffic the congestion avoidance mechanisms specified in [RFC6762] section 7 MUST be followed. The Pledge SHOULD listen for an unsolicited broadcast response as described in [RFC6762]. This allows devices to avoid announcing their presence via mDNS broadcasts and instead silently join a network by watching for periodic unsolicited broadcast responses.

c. MAY: Performs DNS-based Service Discovery [RFC6763] over normal DNS operations. The service searched for is "_bootstrapks._tcp.example.com". In this case the domain "example.com" is discovered as described in [RFC6763] section 11.

d. MAY: If no local bootstrapks service is located using the DNS-based Service Discovery methods the Pledge contacts a well known vendor provided bootstrapping server by performing a DNS lookup using a well known URI such as "bootstrapks.vendor-example.com". The details of the URI are vendor specific. Vendors that leverage this method on the Pledge are responsible for providing the bootstrapks service.

DNS-based service discovery communicates the local proxy IPv4 or IPv6 address and port to the Pledge. Once a proxy is discovered the Pledge communicates with a Registrar through the proxy using the bootstrapping protocol defined in Section 5. The current DNS services returned during each query is maintained until bootstrapping is completed. If bootstrapping fails and the Pledge returns to the Discovery state it picks up where it left off and continues attempting bootstrapping. For example if the first Multicast DNS _bootstrapks._tcp.local response doesn't work then the second and third responses are tried. If these fail the Pledge moves on to normal DNS-based Service Discovery.

Each discovery method attempted SHOULD exponentially back-off attempts (to a maximum of one hour) to avoid overloading the network infrastructure with discovery. The back-off timer for each method MUST be independent of other methods. Methods SHOULD be run in parallel to avoid head of queue problems. Once a connection to a Registrar is established (e.g. establishment of a TLS session key) there are expectations of more timely responses, see Section 5.1.

Once all discovered services are attempted the device SHOULD return to Multicast DNS. It should periodically retry the vendor specific mechanisms. The Pledge may prioritize selection order as appropriate for the anticipated environment.

3.1.2. Identity

The Pledge identifies itself during the communication protocol handshake. If the client identity is rejected the Pledge repeats the Discovery process using the next proxy or discovery method available.

The bootstrapping protocol server is not initially authenticated. Thus the connection is provisional and all data received is untrusted until sufficiently validated even though it is over a TLS connection. This is aligned with the existing provisional mode of EST [RFC7030]

during s4.1.1 "Bootstrap Distribution of CA Certificates". See Section 5.3 for more information about when the TLS connection authenticated is completed.

All security associations established are between the new device and the Bootstrapping server regardless of proxy operations.

3.1.3. Request Join

The Pledge POSTs a request to join the domain to the Bootstrapping server. This request contains a Pledge generated nonce and informs the Bootstrapping server which imprint methods the Pledge will accept.

As indicated in EST [RFC7030] the bootstrapping server MAY redirect the client to an alternate server. This is most useful in the case where the Pledge has resorted to a well known vendor URI and is communicating with the vendor's Registrar directly. In this case the Pledge has authenticated the Registrar using the local Implicit Trust Anchor database and can therefore treat the redirect URI as a trusted URI which can also be validated using the Implicit Trust Anchor database. Since client authentication occurs during the TLS handshake the bootstrapping server has sufficient information to apply appropriate policy concerning which server to redirect to.

The nonce ensures the Pledge can verify that responses are specific to this bootstrapping attempt. This minimizes the use of global time and provides a substantial benefit for devices without a valid clock.

3.1.4. Imprint

The domain trust anchor is received by the Pledge during the bootstrapping protocol methods in the form of a voucher. The goal of the imprint state is to securely obtain a copy of this trust anchor without involving human interaction.

The enrollment protocol EST [RFC7030] details a set of non-autonomic bootstrapping methods such as:

- using the Implicit Trust Anchor database (not an autonomic solution because the URL must be securely distributed),

- engaging a human user to authorize the CA certificate using out-of-band data (not an autonomic solution because the human user is involved),

- using a configured Explicit TA database (not an autonomic solution because the distribution of an explicit TA database is not autonomic),

- and using a Certificate-Less TLS mutual authentication method (not an autonomic solution because the distribution of symmetric key material is not autonomic).

This document describes autonomic methods that MUST be supported by the Pledge:

Audit Voucher Audit Vouchers are obtained by a Registrar from the MASA service and presented to the Pledge for validation. These indicate to the Pledge that joining the domain has been logged by a logging service.

Ownership Voucher Ownership Vouchers are obtained by a Registrar from the MASA service and explicitly indicate the owner of the Pledge. The Ownership Voucher is defined in [I-D.ietf-netconf-zerotouch].

Since client authentication occurs during the TLS handshake the bootstrapping server has sufficient information to apply appropriate policy concerning which method to use.

The Audit Voucher contains the domain's public key material as provided to the MASA service by a Registrar. This provides sufficient information to the client to complete automated bootstrapping with the local key infrastructure. The Ownership Voucher contains the Owner Certificate which the Pledge uses to authenticate the TLS connection.

If the autonomic methods fail the Pledge returns to discovery state and attempts bootstrapping with the next available discovered Registrar.

3.1.5. Lack of realtime clock

Many devices when bootstrapping do not have knowledge of the current time. Mechanisms like Network Time Protocols can not be secured until bootstrapping is complete. Therefore bootstrapping is defined in a method that does not require knowledge of the current time.

Unfortunately there are moments during bootstrapping when certificates are verified, such as during the TLS handshake, where validity periods are confirmed. This paradoxical "catch-22" is resolved by the Pledge maintaining a concept of the current "window"

of presumed time validity that is continually refined throughout the bootstrapping process as follows:

- Initially the Pledge does not know the current time.

- During Pledge authentiation by the Registrar a realtime clock can be used by the Registrar. This bullet expands on a closely related issue regarding Pledge lifetimes. RFC5280 indicates that long lived Pledge certifiates "SHOULD be assigned the GeneralizedTime value of 99991231235959Z" [RFC5280] so the Registrar MUST support such lifetimes and SHOULD support ignoring Pledge lifetimes if they did not follow the RFC5280 recommendations.

- Once the Audit Voucher is accepted the validity period of the domainCAcert in the voucher (see Section 5.3) now describes a valid time window. Any subsequent certificate validity periods checked during RFC5280 path validation MUST occur within this window.

- When accepting an enrollment certificate the validity period within the new certificate is assumed to be valid by the Pledge. The Pledge is now willing to use this credential for client authentication.

Once in this state the Pledge has a valid trust anchor with the local domain and has a locally issued credential. These MAY be used to secure distribution of more accurate time information although specification of such a protocol is out-of-scope of this document.

The nonce included in join attempts provides an alternate mechanism for the Pledge to ensure Audit Voucher responses are associated with a particular bootstrapping attempt. Nonceless Audit Vouchers from the MASA server are always valid and thus time is not needed.

Ownership Vouchers include time information and MUST be validated using a realtime clock.

3.1.6. Enrollment

As the final step of bootstrapping a Registrar helps to issue a domain specific credential to the Pledge. For simplicity in this document, a Registrar primarily facilitates issuing a credential by acting as an RFC5280 Registration Authority for the Domain Certification Authority.

Enrollment proceeds as described in [RFC7030]. Authentication of the EST server is done using the Voucher rather than the methods defined in EST.

Once the Audit or Ownership Voucher is received, as specified in this document, the client has sufficient information to leverage the existing communication channel with a Registrar to continue an EST RFC7030 enrollment. Enrollment picks up at RFC7030 section 4.1.1. bootstrapping where the Audit Voucher provides the "out-of-band" CA certificate fingerprint (in this case the full CA certificate) such that the client can now complete the TLS server authentication. At this point the client continues with EST enrollment operations including "CA Certificates Request", "CSR Attributes" and "Client Certificate Request" or "Server-Side Key Generation".

3.1.7. Being Managed

Functionality to provide generic "configuration" information is supported. The parsing of this data and any subsequent use of the data, for example communications with a Network Management System is out of scope but is expected to occur after bootstrapping enrollment is complete. This ensures that all communications with management systems which can divulge local security information (e.g. network topology or raw key material) is secured using the local credentials issued during enrollment.

The Pledge uses bootstrapping to join only one domain. Management by multiple domains is out-of-scope of bootstrapping. After the device has successfully joined a domain and is being managed it is plausible that the domain can insert credentials for other domains depending on the device capabilities.

See Section 3.5.

3.2. Behavior of a Proxy

The role of the Proxy is to facilitate communications. The Proxy forwards packets between the Pledge and a Registrar that has been configured on the Proxy. The Proxy does not terminate the TLS handshake. A Proxy is always assumed even if directly integrated into a Registrar.

As a result of the Proxy Discovery process in section Section 3.1.1, the port number exposed by the proxy does not need to be well known, or require an IANA allocation.

If the Proxy joins an Autonomic Control Plane ([I-D.ietf-anima-autonomic-control-plane]) it SHOULD use Autonomic

Control Plane secured GRASP ([I-D.ietf-anima-grasp]) to discovery the Registrar address and port. For the IPIP encapsulation methods, the port announced by the Proxy MUST be the same as on the registrar in order for the proxy to remain stateless.

In order to permit the proxy functionality to be implemented on the maximum variety of devices the chosen mechanism SHOULD use the minimum amount of state on the proxy device. While many devices in the ANIMA target space will be rather large routers, the proxy function is likely to be implemented in the control plane CPU such a device, with available capabilities for the proxy function similar to many class 2 IoT devices.

The document [I-D.richardson-anima-state-for-joinrouter] provides a more extensive analysis of the alternative proxy methods.

3.2.1. CoAP connection to Registrar

The proxy MUST implement an IPIP (protocol 41) encapsulation function for CoAP traffic to the configured UDP port on the registrar. The proxy does not terminate the CoAP DTLS connection. [[EDNOTE: The choice of CoAP as the mandatory to implement protocol rather than HTTP maximizes code reuse on the smallest of devices. Unfortunately this means this document will have to include the EST over CoAP details as additional sections. The alternative is to make 'HTTPS proxy' method the mandatory to implement and provide a less friendly environment for the smallest of devices. This is a decision we'll have to see addressed by the broader team.]]

The IPIP encapsulation allows the proxy to forward traffic which is otherwise not to be forwarded, as the traffic between New Node and Proxy use IPv6 Link Local addresses.

If the Proxy device has more than one interface on which it offers the proxy function, then it must select a unique (ACP) IP address per interface in order so that the proxy can stateless return the (link-local) reply packets to the correct link.

3.2.2. HTTPS proxy connection to Registrar

The proxy SHOULD also provide one of: an IPIP encapsulation of HTTP traffic on TCP port TBD to the registrar, or a TCP circuit proxy that connects the Pledge to a Registrar.

When the Proxy provides a circuit proxy to a Registrar the Registrar MUST accept HTTPS connections.

When the Proxy provides a stateless IPIP encapsulation to a Registrar, then the Registrar will have to perform IPIP decapsulation, remembering the originating outer IPIP source address in order to qualify the inner link-local address. This is a kind of encapsulation and processing which is similar in many ways to how mobile IP works.

Being able to connect a TCP (HTTP) or UDP (CoAP) socket to a link-local address with an encapsulated IPIP header requires API extensions beyond [RFC3542] for UDP use, and requires a form of connection latching (see section 4.1 of [RFC5386] and all of [RFC5660], except that a simple IPIP tunnel is used rather than an IPsec tunnel).

3.3. Behavior of the Registrar

A Registrar listens for Pledges and determines if they can join the domain. A Registrar obtains a Voucher from the MASA service and delivers them to the Pledge as well as facilitating enrollment with the domain PKI.

A Registrar is typically configured manually. If the Registrar joins an Autonomic Control Plane ([I-D.ietf-anima-autonomic-control-plane]) it MUST use Autonomic Control Plane secured GRASP ([I-D.ietf-anima-grasp]) to broadcast the Registrar's address and port to potential Proxies.

Registrar behavior is as follows:

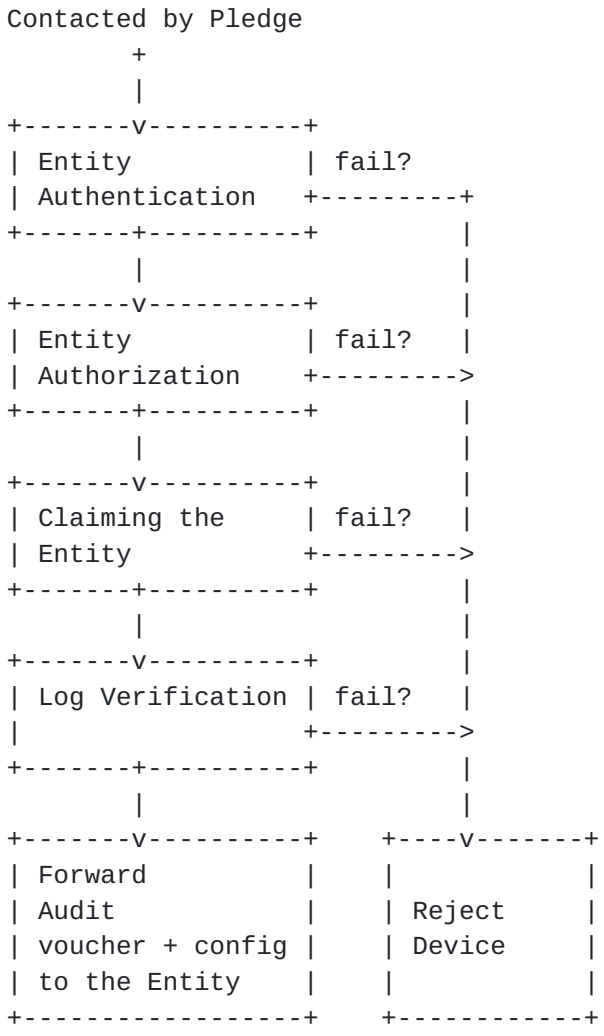


Figure 4

3.3.1. Pledge Authentication

The applicable authentication methods detailed in EST [RFC7030] are:

- the use of an IDevID X.509 credential during the TLS client authentication,

- or the use of a secret that is transmitted out of band between the Pledge and a Registrar (this use case is not autonomic).

In order to validate the IDevID X.509 credential a Registrar maintains a database of vendor trust anchors (e.g. vendor root certificates or keyIdentifiers for vendor root public keys). For user interface purposes this database can be mapped to colloquial vendor names. Registrars can be shipped with the trust anchors of a significant number of third-party vendors within the target market.

3.3.2. Pledge Authorization

In a fully automated network all devices must be securely identified and authorized to join the domain.

A Registrar accepts or declines a request to join the domain, based on the authenticated identity presented. Automated acceptance criteria include:

- allow any device of a specific type (as determined by the X.509 IDevID),
- allow any device from a specific vendor (as determined by the X.509 IDevID),
- allow a specific device from a vendor (as determined by the X.509 IDevID) against a domain white list. (The mechanism for checking a shared white list potentiatlly used by multiple Registrars is out of scope).

To look the Pledge up in a domain white list a consistent method for extracting device identity from the X.509 certificate is required. RFC6125 describes Domain-Based Application Service identity but here we require Vendor Device-Based identity. The subject field's DN encoding MUST include the "serialNumber" attribute with the device's unique serial number. In the language of RFC6125 this provides for a SERIALNUM-ID category of identifier that can be included in a certificate and therefore that can also be used for matching purposes. The SERIALNUM-ID whitelist is collated according to vendor trust anchor since serial numbers are not globally unique.

Since all Pledges accept Audit Vouchers a Registrar MUST use the vendor provided MASA service to verify that the device's history log does not include unexpected Registrars. If a device had previously registered with another domain, a Registrar of that domain would show in the log.

If a Pledge is accepted into the domain, it is expected to request a domain certificate through a certificate enrollment process. The result is a common trust anchor and device certificates for all autonomic devices in a domain (these certificates can be used for other methods, for example boundary detection, auto-securing protocols, etc.). The authorization performed during this phase is used for EST enrollment requests.

3.3.3. Claiming the New Entity

Claiming an entity establishes an audit log at the MASA server and provides a Registrar with proof, in the form of a MASA Audit Voucher, that the log entry has been inserted. As indicated in Section 3.1.4 a Pledge will only proceed with bootstrapping if a validated MASA Audit Voucher has been received. The Pledge therefore enforces that bootstrapping only occurs if the claim has been logged. There is no requirement for the vendor to definitively know that the device is owned by the Registrar.

Registrar's obtain the Vendor URI via static configuration or by extracting it from the X.509 IDevID credential. The imprint method supported by the Pledge is known from the X.509 IDevID credential. [[EDNOTE: An appropriate extension for indicating the Vendor URI and imprint method could be defined using the methods described in [I-D.lear-mud-framework]]].

During initial bootstrapping the Pledge provides a nonce specific to the particular bootstrapping attempt. The Registrar SHOULD include this nonce when claiming the Pledge from the MASA service. Claims from an unauthenticated Registrar are only serviced by the MASA resource if a nonce is provided.

The Registrar can claim a Pledge that is not online by forming the request using the entities unique identifier and not including a nonce in the claim request. Audit Voucher obtained in this way do not have a lifetime and they provide a permanent method for the domain to claim the device. Evidence of such a claim is provided in the audit log entries available to any future Registrar. Such claims reduce the ability for future domains to secure bootstrapping and therefore the Registrar MUST be authenticated by the MASA service although no requirement is implied that the MASA associates this authentication with ownership.

An Ownership Voucher requires the vendor to definitively know that a device is owned by a specific domain. The method used to "claim" this are out-of-scope. A MASA ignores or reports failures when an attempt is made to claim a device that has a an Ownership Voucher.

3.3.4. Log Verification

A Registrar requests the log information for the Pledge from the MASA service. The log is verified to confirm that the following is true to the satisfaction of a Registrar's configured policy:

- Any nonceless entries in the log are associated with domainIDs recognized by the registrar.

- Any nonce'd entries are older than when the domain is known to have physical possession of the Pledge or that the domainIDs are recognized by the registrar.

If any of these criteria are unacceptable to a Registrar the entity is rejected. A Registrar MAY be configured to ignore the history of the device but it is RECOMMENDED that this only be configured if hardware assisted NEA [RFC5209] is supported.

This document specifies a simple log format as provided by the MASA service to the registar. This format could be improved by distributed consensus technologies that integrate the Audit Voucher with a current technologies such as block-chain or hash trees or the like. Doing so is out of the scope of this document but are anticipated improvements for future work.

3.4. Behavior of the MASA Service

The MASA service is provided by the Factory provider on the global Internet. The URI of this service is well known. The URI SHOULD also be provided as an X.509 IDevID extension (a "MASA Audit Voucher Distribution Point" extension).

The MASA service provides the following functionalities to Registrars:

3.4.1. Issue Audit Voucher and Log the event

A Registrar POSTs a claim message optionally containing the bootstrap nonce to the MASA server.

If a nonce is provided the MASA service responds to all requests. The MASA service verifies the Registrar is representative of the domain and generates a privacy protected log entry before responding with the Audit Voucher. For the simple log format defined in this document using the DomainID is considered sufficient privacy. Future work to improve the logging mechanism could include additional privacy protections.

If a nonce is not provided then the MASA service MUST authenticate the Registrar as a valid customer. This prevents denial of service attacks.

3.4.2. Retrieve Audit Entries from Log

When determining if a Pledge should be accepted into a domain the Registrar retrieves a copy of the audit log from the MASA service. This contains a list of privacy protected domain identities that have

previously claimed the device. Included in the list is an indication of the time the entry was made and if the nonce was included.

3.5. Leveraging the new key infrastructure / next steps

As the devices have a common trust anchor, device identity can be securely established, making it possible to automatically deploy services across the domain in a secure manner.

Examples of services:

- Device management.
- Routing authentication.
- Service discovery.

3.5.1. Network boundaries

When a device has joined the domain, it can validate the domain membership of other devices. This makes it possible to create trust boundaries where domain members have higher level of trusted than external devices. Using the autonomic User Interface, specific devices can be grouped into to sub domains and specific trust levels can be implemented between those.

3.6. Interactions with Network Access Control

The assumption is that Network Access Control (NAC) completes using the Pledge 's X.509 IDevID credentials and results in the device having sufficient connectivity to discovery and communicate with the proxy. Any additional connectivity or quarantine behavior by the NAC infrastructure is out-of-scope. After the devices has completed bootstrapping the mechanism to trigger NAC to re-authenticate the device and provide updated network privileges is also out-of-scope.

This achieves the goal of a bootstrap architecture that can integrate with NAC but does not require NAC within the network where it wasn't previously required. Future optimizations can be achieved by integrating the bootstrapping protocol directly into an initial EAP exchange.

4. Domain Operator Activities

This section describes how an operator interacts with a domain that supports the bootstrapping as described in this document.

4.1. Instantiating the Domain Certification Authority

This is a one time step by the domain administrator. This is an "off the shelf" CA with the exception that it is designed to work as an integrated part of the security solution. This precludes the use of 3rd party certification authority services that do not provide support for delegation of certificate issuance decisions to a domain managed Registration Authority.

4.2. Instantiating the Registrar

This is a one time step by the domain administrator. One or more devices in the domain are configured take on a Registrar function.

A device can be configured to act as a Registrar or a device can auto-select itself to take on this function, using a detection mechanism to resolve potential conflicts and setup communication with the Domain Certification Authority. Automated Registrar selection is outside scope for this document.

4.3. Accepting New Entities

For each Pledge the Registrar is informed of the unique identifier (e.g. serial number) along with the manufacturer's identifying information (e.g. manufacturer root certificate). This can happen in different ways:

1. Default acceptance: In the simplest case, the new device asserts its unique identity to a Registrar. The registrar accepts all devices without authorization checks. This mode does not provide security against intruders and is not recommended.

2. Per device acceptance: The new device asserts its unique identity to a Registrar. A non-technical human validates the identity, for example by comparing the identity displayed by the registrar (for example using a smartphone app) with the identity shown on the packaging of the device. Acceptance may be triggered by a click on a smartphone app "accept this device", or by other forms of pairing. See also [I-D.behringer-homenet-trust-bootstrap] for how the approach could work in a homenet.

3. Whitelist acceptance: In larger networks, neither of the previous approaches is acceptable. Default acceptance is not secure, and a manual per device methods do not scale. Here, the registrar is provided a priori with a list of identifiers of devices that belong to the network. This list can be extracted from an inventory database, or sales records. If a device is detected

that is not on the list of known devices, it can still be manually accepted using the per device acceptance methods.

4. Automated Whitelist: an automated process that builds the necessary whitelists and inserts them into the larger network domain infrastructure is plausible. Once set up, no human intervention is required in this process. Defining the exact mechanisms for this is out of scope although the registrar authorization checks is identified as the logical integration point of any future work in this area.

None of these approaches require the network to have permanent Internet connectivity. Even when the Internet based MASA service is used, it is possible to pre-fetch the required information from the MASA a priori, for example at time of purchase such that devices can enroll later. This supports use cases where the domain network may be entirely isolated during device deployment.

Additional policy can be stored for future authorization decisions. For example an expected deployment time window or that a certain Proxy must be used.

4.4. Automatic Enrollment of Devices

The approach outlined in this document provides a secure zero-touch method to enroll new devices without any pre-staged configuration. New devices communicate with already enrolled devices of the domain, which proxy between the new device and a Registrar. As a result of this completely automatic operation, all devices obtain a domain based certificate.

4.5. Secure Network Operations

The certificate installed in the previous step can be used for all subsequent operations. For example, to determine the boundaries of the domain: If a neighbor has a certificate from the same trust anchor it can be assumed "inside" the same organization; if not, as outside. See also Section 3.5.1. The certificate can also be used to securely establish a connection between devices and central control functions. Also autonomic transactions can use the domain certificates to authenticate and/or encrypt direct interactions between devices. The usage of the domain certificates is outside scope for this document.

5. Protocol Details

A bootstrapping protocol could be implemented as an independent protocol from EST, but for simplicity and to reduce the number of TLS connections and crypto operations required on the Pledge, it is described specifically as extensions to EST. These extensions MUST be supported by the Registrar EST server within the same .well-known URI tree as the existing EST URIs as described in [RFC7030] section 3.2.2.

The Pledge establishes a TLS connection with the Registrar through the circuit proxy (see Section 3.2) but the TLS connection is with the Registar; so for this section the "Pledge" is the TLS client and the "Registrar" is the TLS server.

Establishment of the TLS connection for bootstrapping is as specified for EST [RFC7030]. In particular server identity and client identity are as described in EST [RFC7030] section 3.3. In EST [RFC7030] provisional server authentication for bootstrapping is described in section 4.1.1 wherein EST clients can "engage a human user to authorize the CA certificate using out-of-band data such as a CA certificate" or wherein a human user configures the URI of the EST server for Implicit TA based authentication. As described in this document, Section 5.3.1, a new method of bootstrapping now provides a completely automating method of bootstrapping PKI.

The extensions for the Pledge client are as follows:

- The Pledge provisionally accept the EST server certificate during the TLS handshake as detailed in Section 5.3.1.

- The Pledge requests and validates the Audit Voucher as described below. At this point the Pledge has sufficient information to validate domain credentials.

- The Pledge calls the EST defined /cacerts method to obtain the current CA certificate. These are validated using the Audit Voucher.

- The Pledge completes bootstrapping as detailed in EST section 4.1.1.

In order to obtain a validated Audit Voucher and Audit Log a Registrar contacts the MASA service Service using REST calls:

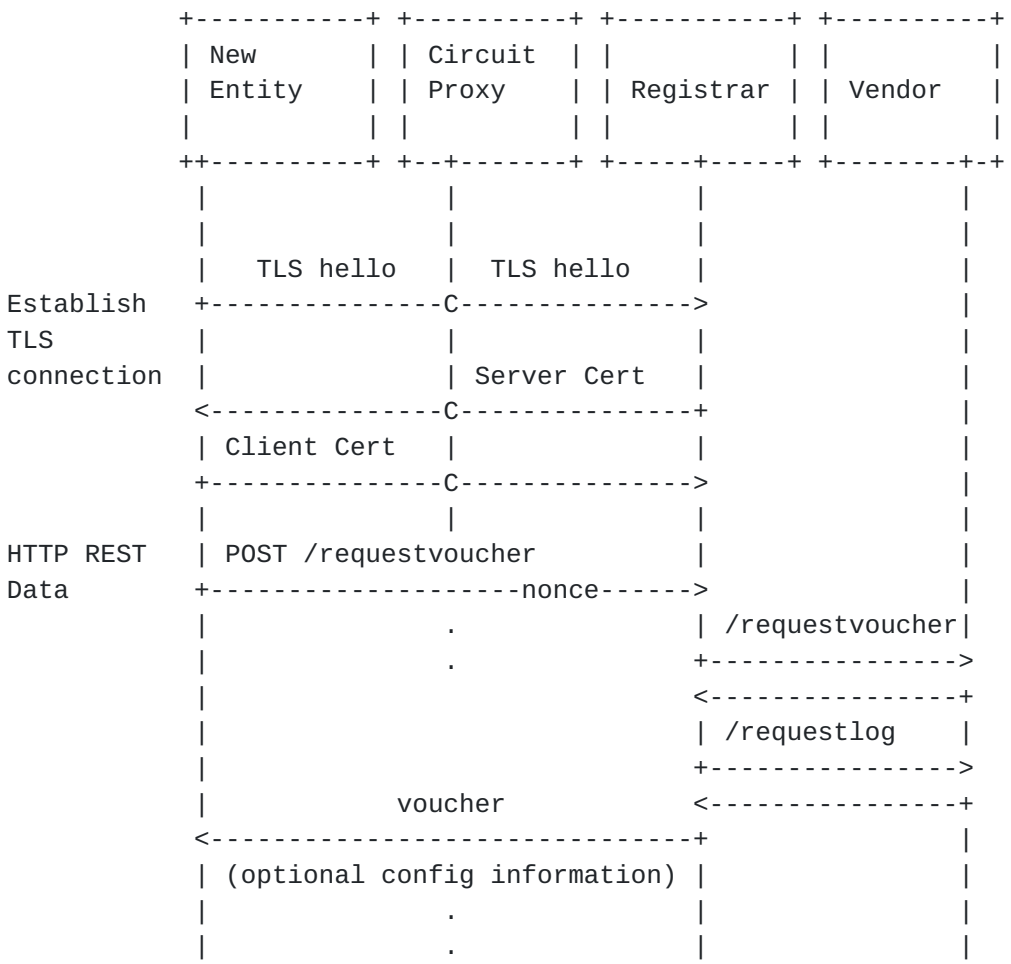


Figure 5

In some use cases the Registrar may need to contact the Vendor in advanced, for example when the target network is air-gapped. The nonceless request format is provided for this and the resulting flow is slightly different. The security differences associated with not knowing the nonce are discussed below:

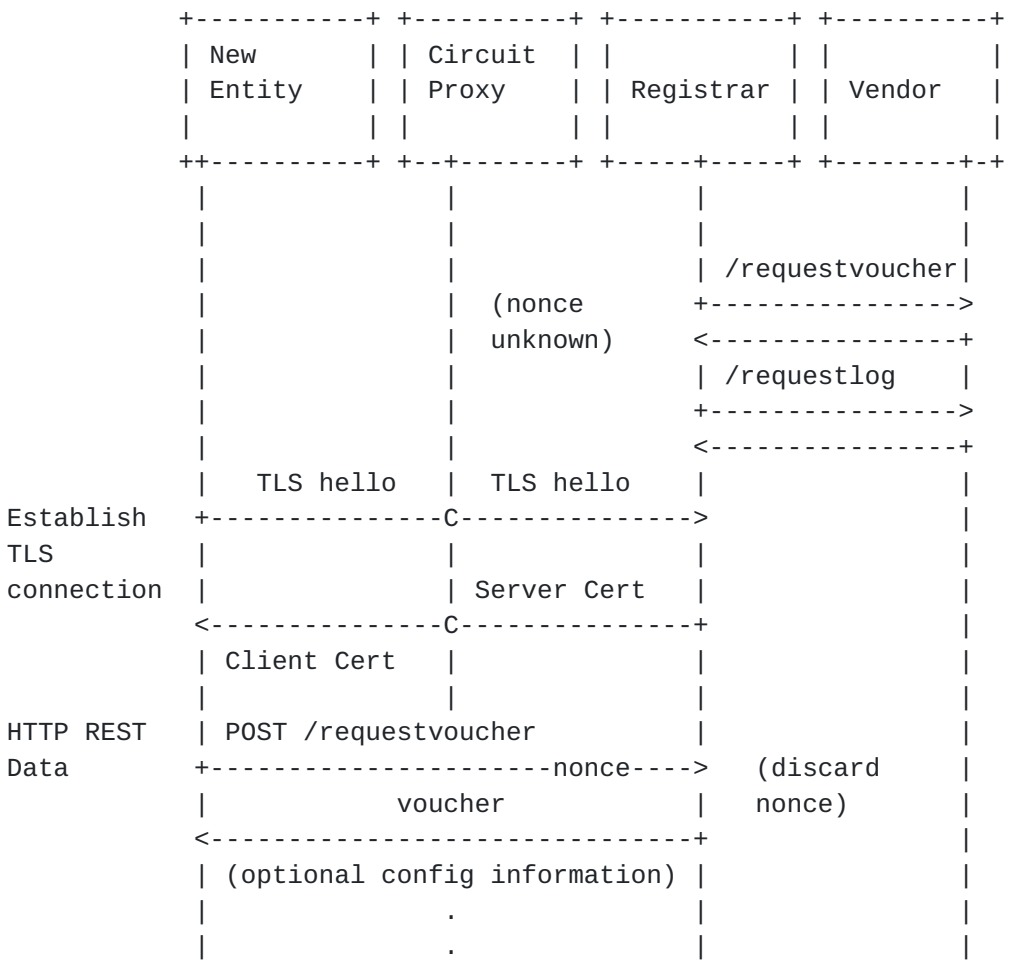


Figure 6

The extensions for a Registrar server are as follows:

- The Registrar requests and validates the Audit Voucher from the vendor authorized MASA service.

- The Registrar forwards the Audit Voucher to the Pledge when requested.

- The Registar performs log verifications in addition to local authorization checks before accepting the Pledge device.

5.1. Request Voucher from the Registrar

When the Pledge bootstraps it makes a request for a Voucher from a Registrar.

This is done with an HTTPS POST using the operation path value of "/requestvoucher".

The request format is JSON object containing a 64bit nonce generated by the client for each request. This nonce MUST be a cryptographically strong random or pseudo-random number that can not be easily predicted. The nonce MUST NOT be reused for multiple attempts to join a network domain. The nonce assures the Pledge that the Audit Voucher response is associated with this bootstrapping attempt and is not a replay.

Request media type: application/auditnonce

Request format: a JSON file with the following:

```
{
 "version":"1",
 "nonce":"<64bit nonce value>",
}
```

[[EDNOTE: Even if the nonce was signed it would provide no defense against rogue registrars; although it would assure the MASA that a certified Pledge exists. To protect against rogue registrars a nonce component generated by the MASA (a new round trip) would be required). Instead this is addressed by requiring MASA & Registrar authentications but it is worth exploring additional protections. This to be explored more at IETF96.]]

The Registrar validates the client identity as described in EST [RFC7030] section 3.3.2. The registrar performs authorization as detailed in Section 3.3.2. If authorization is successful the Registrar obtains an Voucher from the MASA service (see Section 5.2).

The received Voucher is forwarded to the Pledge.

As indicated in EST [RFC7030] the bootstrapping server can redirect the client to an alternate server. If the Pledge authenticated a Registrar using the well known URI method then the Pledge MUST follow the redirect automatically and authenticate the new Registrar against the redirect URI provided. If the Pledge had not yet authenticated a Registrar because it was discovered and was not a known-to-be-valid URI then the new Registrar must be authenticated using one of the two autonomic methods described in this document. Similarly the Registar MAY respond with an HTTP 202 ("the request has been accepted for processing, but the processing has not been completed") as described in EST [RFC7030] section 4.2.3.

Recall that during this communication with the Registar the TLS authentication is only provisional. The Pledge client MUST handle all data from the Registrar with upmost care. In particular the Pledge MUST only allow a single redirection and MUST only support a

delay of five seconds before declaring the Registrar a failure and moving on to the next discovered Registrar. As detailed in Section 3.1.1 if no suitable Registrar is found the Pledge restarts the state machine and tries again. So a Registrar that is unable to complete the transaction the first time will have future chances.

5.2. Request Voucher from MASA

A Registrar requests a Voucher from the MASA service using a REST interface. For simplicity this is defined as an optional EST message between a Registrar and an EST server running on the MASA service although the Registrar is not required to make use of any other EST functionality when communicating with the MASA service. (The MASA service MUST properly reject any EST functionality requests it does not wish to service; a requirement that holds for any REST interface).

This is done with an HTTP POST using the operation path value of "/requestvoucher".

The request format is a JSON object optionally containing the nonce value (as obtained from the bootstrap request) and the X.509 IDevID extracted serial number (the full certificate is not needed and no proof-of-possession information for the device identity is included). The AuthorityKeyIdentifier value from the certificate is included to ensure a statistically unique identity. The Pledge's serial number is extracted from the X.509 IDevID subject name id-at-serialNumber or it is the base64 encoded RFC4108 hardwareModuleName hwSerialNum:

```
{
 "version":"1",
 "nonce":"<64bit nonce value>",
 "IDevIDAuthorityKeyIdentifier":"<base64 encoded keyIdentifier">,
 "DevIDSerialNumber":"<id-at-serialNumber or base64 encoded
                      hardwareModuleName hwSerialNum>",
}
```

A Registrar MAY exclude the nonce from the request. Doing so allows the Registrar to request a Voucher when the Pledge is not online, or when the target bootstrapping environment is not on the same network as the MASA server (this requires the Registrar to learn the appropriate DevIDSerialNumber field from the physical device labeling or from the sales channel -- how this occurs is out-of-scope of this document). If a nonce is not provided the MASA server MUST authenticate the client as described in EST [RFC7030] section 3.3.2 to reduce the risk of DDoS attacks. A Registrar performs authorization as detailed in Section 3.3.2. If authorization is

successful the Registrar obtains an Voucher from the MASA service (see Section 5.2).

The JSON message information is encapsulated in a [RFC5652] Signed-data that is signed by the Registrar. The entire certificate chain, up to and including the Domain CA, MUST be included in the CertificateSet structure. The MASA service checks the internal consistency of the CMS but does not authenticate the domain identity information. The domain is not know to the MASA server in advance and a shared trust anchor is not implied. The MASA server MUST verify that the CMS is signed by a Registrar certificate (by checking for the cmc-idRA field) that was issued by a the root certificate included in the CMS. This ensures that the Registrar making the claim is an authorized Registrar of the unauthenticated domain. The EST style client authentication (TLS and HTTP) is used to provide a DDoS prevention strategy.

The root certificate is extracted and used to populate the Audit Voucher. The domain ID (e.g. hash of the public key of the domain) is extracted from the root certificate and is used to update the audit log.

5.3. Audit Voucher Response

The voucher response to requests from the device and requests from a Registrar are in the same format. A Registrar either caches prior MASA responses or dynamically requests a new Voucher based on local policy.

If the the join operation is successful, the server response MUST contain an HTTP 200 response code with a content-type of "application/authorizationvoucher". The server MUST answer with a suitable 4xx or 5xx HTTP [RFC2616] error code when a problem occurs. The response data from the MASA server MUST be a plaintext human-readable error message containing explanatory information describing why the request was rejected.

The Audit Voucher consists of the nonce, if supplied, the serial number information identifying the device and the domain CA certificate extracted from the request:

```
{
 "version":"1",
 "nonce":"<64bit nonce value>",
 "IDevIDAuthorityKeyIdentifier":"<base64 encoded keyIdentifier>",
 "DevIDSerialNumber":"<id-at-serialNumber>",
 "domainCAcert":"<the base64 encoded domain CA's certificate>"
}
```

The Audit Voucher response is encapsulated in a [RFC5652] Signed-data that is signed by the MASA server. The Pledge verifies this signed message using the manufacturer installed trust anchor assocaited with the X.509 IDevID. [[EDNOTE: As detailed in netconf-zerotouch this might be a distinct trust anchor rather than re-using the trust anchor for the IDevID. This concept will need to be detailed in this document as well.]]

[[EDNOTE: Using CMS is consistent with the alignment of this bootstrapping document with EST, a PKIX enrollment protocol that includes Certificate Management over CMS. An alternative format would be the RFC7515 JSON Web Signature (JWS), which would allow clients that do not use fullCMC messages to avoid CMS entirely. Use of JWS would likely include a discussion of CBOR in order ensure the base64 expansions of the certs and signatures within the JWS message are of minimal size -- it is not yet clear to this author how that would work out]]

The 'domainCAcert' element of this message contains the domain CA's public key. This is specific to bootstrapping a public key infrastructure. To support bootstrapping other key infrastructures additional domain identity types might be defined in the future. Clients MUST be prepared to ignore additional fields they do not recognize. Clients MUST be prepared to parse and fail gracefully from an Audit Voucher response that does not contain a 'domainCAcert' field at all.

To minimize the size of the Audit Voucher response message the domainCAcert is not a complete distribution of the EST section 4.1.3 CA Certificate Response.

The Pledge installs the domainCAcert trust anchor. As indicated in Section 3.1.2 the newly installed trust anchor is used as an EST RFC7030 Explicit Trust Anchor. The Pledge MUST use the domainCAcert trust anchor to immediately validate the currently provisional TLS connection to a Registrar.

5.3.1. Completing authentication of Provisional TLS connection

If a Registrar's credential can not be verified using the domainCAcert trust anchor the TLS connection is immediately discarded and the Pledge abandons attempts to bootstrap with this discovered registrar.

The following behaviors on a Registrar and Pledge are in addition to normal PKIX operations:

- The EST server MUST use a certificate that chains to the domainCAcert. This means that when the EST server obtains renewed credentials the credentials included in the Section 5.2 request match the chain used in the current provisional TLS connection.

- The Pledge PKIX path validation of a Registrar validity period information is as described in Section 3.1.5.

Because the domainCAcert trust anchor is installed as an Explicit Trust Anchor it can be used to authenticate any dynamically discovered EST server that contain the id-kp-cmcRA extended key usage extension as detailed in EST RFC7030 section 3.6.1; but to reduce system complexity the Pledge SHOULD avoid additional discovery operations. Instead the Pledge SHOULD communicate directly with the Registrar as the EST server to complete PKI local certificate enrollment. Additionally the Pledge SHOULD use the existing TLS connection to proceed with EST enrollment, thus reducing the total amount of cryptographic and round trip operations required during bootstrapping. [[EDNOTE: It is reasonable to mandate that the existing TLS connection be re-used? e.g. MUST >> SHOULD?]]

5.4. Voucher Status Telemetry

For automated bootstrapping of devices the adminstrative elements providing bootstrapping also provide indications to the system administrators concerning device lifecycle status. To facilitate this those elements need telemetry information concerning the device's status.

To indicate Pledge status regarding the Audit Voucher the client SHOULD post a status message.

The client HTTP POSTs the following to the server at the EST well known URI /voucher_status. The Status field indicates if the Voucher was acceptable. If it was not acceptable the Reason string indicates why. In the failure case this message is being sent to an unauthenticated, potentially malicious Registrar and therefore the Reason string SHOULD NOT provide information beneficial to an attacker. The operational benefit of this telemetry information is balanced against the operational costs of not recording that an Voucher was ignored by a client the registar expected to continue joining the domain.

```
{
  "version":"1",
  "Status":FALSE /* TRUE=Success, FALSE=Fail"
  "Reason":"Informative human readable message"
}
```

The server SHOULD respond with an HTTP 200 but MAY simply fail with an HTTP 404 error. The client ignores any response. Within the server logs the server SHOULD capture this telemetry information.

5.5. MASA authorization log Request

A registrar requests the MASA authorization log from the MASA service using this EST extension.

This is done with an HTTP GET using the operation path value of "/requestauditlog".

The client HTTP POSTs the same Voucher Request as for requesting an audit token but now posts it to the /requestauditlog URI instead. The IDevIDAuthorityKeyIdentifier and DevIDSerialNumber informs the MASA server which log is requested so the appropriate log can be prepared for the response.

5.6. MASA authorization log Response

A log data file is returned consisting of all log entries. For example:

```
{
  "version":"1",
  "events":[
    {
     "date":"<date/time of the entry>",
     "domainID":"<domainID as extracted from the domain CA certificate
                 within the CMS of the audit voucher request>",
     "nonce":"<any nonce if supplied (or the exact string 'NULL')>"
    },
    {
     "date":"<date/time of the entry>",
     "domainID":"<domainID as extracted from the domain CA certificate
                 within the CMS of the audit voucher request>",
     "nonce":"<any nonce if supplied (or the exact string 'NULL')>"
    }
  ]
}
```

Distribution of a large log is less than ideal. This structure can be optimized as follows: All nonce-less entries for the same domainID MAY be condensed into the single most recent nonceless entry.

A Registrar uses this log information to make an informed decision regarding the continued bootstrapping of the Pledge. For example if the log includes unexpected domainIDs this is indicative of

problematic imprints by the Pledge. If the log includes nonce-less entries this is indicative of the permanent ability for the indicated domain to trigger a reset of the device and take over management of it. Equipment that is purchased pre-owned can be expected to have an extensive history.

Log entries containing the Domain's ID can be compared against local history logs in search of discrepancies.

5.7. EST Integration for PKI bootstrapping

The prior sections describe EST extensions necessary to enable fully automated bootstrapping. Although the Audit Voucher request/response structure members IDevIDAuthorityKeyIdentifier and DevIDSerialNumber are specific to PKI bootstrapping these are the only PKI specific aspects of the extensions and future work might replace them with non-PKI structures.

The prior sections provide functionality for the Pledge to obtain a trust anchor representative of the Domain. The following section describe using EST to obtain a locally issued PKI certificate. The Pledge SHOULD leverage the discovered Registrar to proceed with certificate enrollment and, if they do, MUST implement the EST options described in this section. The Pledge MAY perform alternative enrollment methods including discovering an alternate EST server, or proceed to use its IDevID credential indefinitely.

5.7.1. EST Distribution of CA Certificates

The Pledge MUST request the full EST Distribution of CA Certificates message. See RFC7030, section 4.1.

This ensures that the Pledge has the complete set of current CA certificates beyond the domainCAcert (see Section 5.3 for a discussion of the limitations). Although these restrictions are acceptable for a Registrar integrated with initial bootstrapping they are not appropriate for ongoing PKIX end entity certificate validation.

5.7.2. EST CSR Attributes

Automated bootstrapping occurs without local administrative configuration of the Pledge. In some deployments its plausible that the Pledge generates a certificate request containing only identity information known to the Pledge (essentially the IDevID information) and ultimately receives a certificate containing domain specific identity information. Conceptually the CA has complete control over all fields issued in the end entity certificate. Realistically this

is operationally difficult with the current status of PKI certificate authority deployments where the CSR is submitted to the CA via a number of non-standard protocols.

To alleviate operational difficulty the Pledge MUST request the EST "CSR Attributes" from the EST server. This allows the local infrastructure to inform the Pledge of the proper fields to include in the generated CSR.

[[EDNOTE: The following is specific to anima purposes and should be moved to an appropriate anima document so as to keep bootstrapping as generic as possible: What we want are a 'domain name' stored in [TBD] and an 'ACP IPv6 address' stored in the iPAddress field as specified in RFC5208 s4.2.1.6. ref ACP draft where certificate verification [TBD]. These should go into the subjectaltname in the [TBD] fields.]]. If the hardwareModuleName in the IDevID is populated then it SHOULD by default be propagated to the LDevID along with the hwSerialNum. The registar SHOULD support local policy concerning this functionality. [[EDNOTE: extensive use of EST CSR Attributes might need an new OID definition]].]]

The Registar MUST also confirm the resulting CSR is formatted as indicated before forwarding the request to a CA. If the Registar is communicating with the CA using a protocol like full CMC which provides mechanisms to override the CSR attributes, then these mechanisms MAY be used even if the client ignores CSR Attribute guidance.

5.7.3. EST Client Certificate Request

The Pledge MUST request a new client certificate. See RFC7030, section 4.2.

5.7.4. Enrollment Status Telemetry

For automated bootstrapping of devices the adminstrative elements providing bootstrapping also provide indications to the system administrators concerning device lifecycle status. This might include information concerning attempted bootstrapping messages seen by the client, MASA provides logs and status of credential enrollment. The EST protocol assumes an end user and therefore does not include a final success indication back to the server. This is insufficient for automated use cases.

To indicate successful enrollment the client SHOULD re-negotiate the EST TLS session using the newly obtained credentials. This occurs by the client initiating a new TLS ClientHello message on the existing

TLS connection. The client MAY simply close the old TLS session and start a new one. The server MUST support either model.

In the case of a failure the Reason string indicates why the most recent enrollment failed. The SubjectKeyIdentifier field MUST be included if the enrollment attempt was for a keypair that is locally known to the client. If EST /serverkeygen was used and failed then the this field is ommited from the status telemetry.

The client HTTP POSTs the following to the server at the new EST well known URI /enrollstatus.

```
{
  "version":"1",
  "Status":TRUE /* TRUE=Success, FALSE=Fail"
  "Reason":"Informative human readable message"
  "SubjectKeyIdentifier":"<base64 encoded subjectkeyidentifier for the
                           enrollment that failed>"
}
```

The server SHOULD respond with an HTTP 200 but MAY simply fail with an HTTP 404 error.

Within the server logs the server MUST capture if this message was recieved over an TLS session with a matching client certificate. This allows for clients that wish to minimize their crypto operations to simpy POST this response without renegotiating the TLS session - at the cost of the server not being able to accurately verify that enrollment was truly successful.

5.7.5. EST over CoAP

[[EDNOTE: In order to support smaller devices the above section on Proxy behavior introduces mandatory to implement support for CoAP support by the Proxy. This implies similar support by the Pledge and Registrar and means that the EST protocol operation encapsulation into CoAP needs to be described. EST is HTTP based and "CoaP is designed to easily interface with HTTP for integration" [RFC7252]. Use of CoAP implies Datagram TLS (DTLS) wherever this document describes TLS handshake specifics. A complexity is that the large message sizes necessary for bootstrapping will require support for [draft-ietf-core-block].]]

6. Reduced security operational modes

A common requirement of bootstrapping is to support less secure operational modes for support specific use cases. The following

sections detail specific ways that the Pledge, Registrar and MASA can be configured to run in a less secure mode for the indicated reasons.

6.1. Trust Model

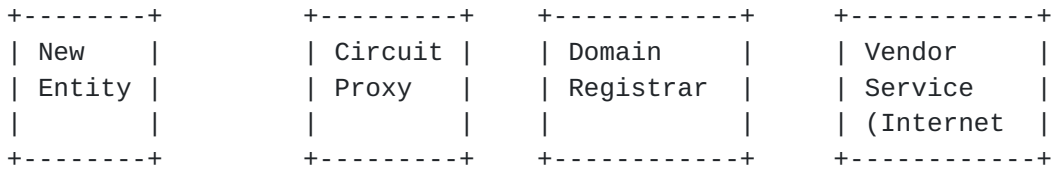


Figure 7

Pledge: The Pledge could be compromised and providing an attack vector for malware. The entity is trusted to only imprint using secure methods described in this document. Additional endpoint assessment techniques are RECOMMENDED but are out-of-scope of this document.

Proxy: Provides proxy functionalities but is not involved in security considerations.

Registrar: When interacting with a MASA server a Registrar makes all decisions. When Ownership Vouchers are involved a Registrar is only a conduit and all security decisions are made on the vendor service.

Vendor Service, MASA: This form of vendor service is trusted to accurately log all claim attempts and to provide authoritative log information to Registrars. The MASA does not know which devices are associated with which domains. These claims could be strengthened by using cryptographic log techniques to provide append only, cryptographic assured, publicly auditable logs. Current text provides only for a trusted vendor.

Vendor Service, Ownership Validation: This form of vendor service is trusted to accurately know which device is owned by which domain.

6.2. New Entity security reductions

The Pledge MAY support "trust on first use" on physical interfaces but MUST NOT support "trust on first use" on network interfaces. This is because "trust on first use" permanently degrades the security for all other use cases.

The Pledge MAY have an operational mode where it skips Voucher validation one time. For example if a physical button is depressed during the bootstrapping operation. This can be useful if the vendor

service is unavailable. This behavior SHOULD be available via local configuration or physical presence methods to ensure new entities can always be deployed even when autonomic methods fail. This allows for unsecured imprint.

It is RECOMMENDED that this only be available if hardware assisted NEA [RFC5209] is supported.

6.3. Registrar security reductions

A Registrar can choose to accept devices using less secure methods. These methods are acceptable when low security models are needed, as the security decisions are being made by the local administrator, but they MUST NOT be the default behavior:

1. A registrar MAY choose to accept all devices, or all devices of a particular type, at the administrator's discretion. This could occur when informing all Registrars of unique identifiers of new entities might be operationally difficult.

2. A registrar MAY choose to accept devices that claim a unique identity without the benefit of authenticating that claimed identity. This could occur when the Pledge does not include an X.509 IDevID factory installed credential. New Entities without an IDevID credential MAY form the Section 5.1 request using the Section 5.2 format to ensure the Pledge's serial number information is provided to the Registar (this includes the IDevIDAuthorityKeyIdentifier value which would be statically configured on the Pledge). The Pledge MAY refused to provide a TLS client certificate (as one is not available). The Pledge SHOULD support HTTP-based or certificate-less TLS authentication as described in EST RFC7030 section 3.3.2. A Registrar MUST NOT accept unauthenticated New Entities unless it has been configured to do so by an administrator that has verified that only expected new entities can communicate with a Registrar (presumably via a physically secured perimeter).

3. A Registrar MAY request nonce-less Audit Vouchers from the MASA service (by not including a nonce in the request). These Audit Vouchers can then be transmitted to the Registrar and stored until they are needed during bootstrapping operations. This is for use cases where target network is protected by an air gap and therefore can not contact the MASA service during Pledge deployment.

4. A registrar MAY ignore unrecognized nonce-less Audit Log entries. This could occur when used equipment is purchased with a valid

history being deployed in air gap networks that required permanent Audit Vouchers.

These modes are not available for devices that require a vendor Ownership Voucher. The methods vendors use to determine which devices are owned by which domains is out-of-scope.

6.4. MASA security reductions

Lower security modes chosen by the MASA service effect all device deployments unless bound to the specific device identities. In which case these modes can be provided as additional features for specific customers. The MASA service can choose to run in less secure modes by:

1. Not enforcing that a Nonce is in the Audit Voucher. This results in distribution of Audit Voucher that never expire and in effect makes the Domain an always trusted entity to the Pledge during any subsequent bootstrapping attempts. That this occurred is captured in the log information so that the Domain registrar can make appropriate security decisions when a Pledge joins the Domain. This is useful to support use cases where Registrars might not be online during actual device deployment. Because this results in long lived Audit Voucher and do not require the proof that the device is online this is only accepted when the Registrar is authenticated by the MASA server and authorized to provide this functionality. The MASA server is RECOMMENDED to use this functionality only in concert with Ownership Validation tracking.

2. Not verifying ownership before responding with an Audit Voucher. This is expected to be a common operational model because doing so relieves the vendor providing MASA services from having to tracking ownership during shipping and supply chain and allows for a very low overhead MASA service. A Registrar uses the audit log information as a defense in depth strategy to ensure that this does not occur unexpectedly (for example when purchasing new equipment the Registrar would throw an error if any audit log information is reported).

7. Security Considerations

In order to support a wide variety of use cases, devices can be claimed by a registrar without proving possession of the device in question. This would result in a nonceless, and thus always valid, claim. Or would result in an invalid nonce being associated with a claim. The MASA service is required to authenticate such Registrars but no programmatic method is provided to ensure good behavior by the

MASA service. Nonceless entries into the audit log therefore permanently reduce the value of a device because future Registrars, during future bootstrap attempts, would now have to be configured with policy to ignore previously (and potentially unknown) domains.

Future registrars are recommended to take the audit history of a device into account when deciding to join such devices into their network. If the MASA server were to have allowed a significantly large number of claims this might become onerous to the MASA server which must maintain all the extra log entries. Ensuring a Registrar is representative of a valid customer domain even without validating ownership helps to mitigate this.

It is possible for an attacker to send an authorization request to the MASA service directly after the real Registrar obtains an authorization log. If the attacker could also force the bootstrapping protocol to reset there is a theoretical opportunity for the attacker to use the Audit Voucher to take control of the Pledge but then proceed to enroll with the target domain. Possible prevention mechanisms include:

- Per device rate limits on the MASA service ensure such timing attacks are difficult.

- In the advent of an unexpectedly lost bootstrapping connection the Registrar repeats the request for audit log information.

To facilitate logging and administrative oversight the Pledge reports on Audit Voucher parsing status to the Registrar. In the case of a failure this information is informative to a potentially malicious Registar but this is RECOMMENDED anyway because of the operational benefits of an informed administrator in cases where the failure is indicative of a problem.

As indicated in EST [RFC7030] the connection is provisional and untrusted until the server is successfully authorized. If the server provides a redirect response the client MUST follow the redirect but the connection remains provisional. If the client uses a well known URI for contacting a well known Registrar the EST Implicit Trust Anchor database is used as is described in RFC6125 to authenticate the well known URI. In this case the connection is not provisional and RFC6125 methods can be used for each subsequent redirection.

To facilitate truely limited clients EST RFC7030 section 3.3.2 requirements that the client MUST support a client authentication model have been reduced in Section 6 to a statement that clients only "SHOULD" support such a model. This reflects current (not great) practices but is NOT RECOMMENDED.

The MASA service could lock a claim and refuse to issue a new voucher or the MASA service could go offline (for example if a vendor went out of business). This functionality provides benefits such as theft resistance, but it also implies an operational risk to the Domain that Vendor behavior could limit future bootstrapping of the device by the Domain. This can be mitigated by Registrars that request nonce-less Audit Vouchers.

7.1. Security concerns with discovery process

7.1.1. Discovery of Registrar by Proxy

As described in section Section 3.2, the RECOMMENDED mechanism is for the proxy to discover the address of the registrar via GRASP [I-D.ietf-anima-grasp]

GRASP is intended to run over a secured, and private Autonomic Control Plan [I-D.ietf-anima-autonomic-control-plane]. This discovery is between the already registered Registrar, and the already registered Proxy. There are no GRASP security issues with this part, as both entities will have already joined the secured ACP.

7.1.2. Discovery of Proxy by New Entity

[[EDNOTE: To be discussed]]

8. Acknowledgements

We would like to thank the various reviewers for their input, in particular Markus Stenberg, Brian Carpenter, Fuyu Eleven, Toerless Eckert, Eliot Lear and Sergey Kasatkin.

9. References

9.1. Normative References

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

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

[RFC3542] Stevens, W., Thomas, M., Nordmark, E., and T. Jinmei, "Advanced Sockets Application Program Interface (API) for IPv6", RFC 3542, DOI 10.17487/RFC3542, May 2003, <http://www.rfc-editor.org/info/rfc3542>.

[RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, DOI 10.17487/RFC3927, May 2005, <http://www.rfc-editor.org/info/rfc3927>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[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, <http://www.rfc-editor.org/info/rfc5280>.

[RFC5386] Williams, N. and M. Richardson, "Better-Than-Nothing Security: An Unauthenticated Mode of IPsec", RFC 5386, DOI 10.17487/RFC5386, November 2008, <http://www.rfc-editor.org/info/rfc5386>.

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

[RFC5660] Williams, N., "IPsec Channels: Connection Latching", RFC 5660, DOI 10.17487/RFC5660, October 2009, <http://www.rfc-editor.org/info/rfc5660>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <http://www.rfc-editor.org/info/rfc6762>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <http://www.rfc-editor.org/info/rfc6763>.

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

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

9.2. Informative References

[I-D.behringer-homenet-trust-bootstrap]
Behringer, M., Pritikin, M., and S. Bjarnason, "Bootstrapping Trust on a Homenet", draft-behringer-homenet-trust-bootstrap-02 (work in progress), February 2014.

[I-D.ietf-ace-actors]
Gerdes, S., Seitz, L., Selander, G., and C. Bormann, "An architecture for authorization in constrained environments", draft-ietf-ace-actors-04 (work in progress), September 2016.

[I-D.ietf-anima-autonomic-control-plane]
Behringer, M., Eckert, T., and S. Bjarnason, "An Autonomic Control Plane", draft-ietf-anima-autonomic-control-plane-03 (work in progress), July 2016.

[I-D.ietf-anima-grasp]
Bormann, C., Carpenter, B., and B. Liu, "A Generic Autonomic Signaling Protocol (GRASP)", draft-ietf-anima-grasp-08 (work in progress), October 2016.

[I-D.ietf-netconf-zerotouch]
Watsen, K. and M. Abrahamsson, "Zero Touch Provisioning for NETCONF or RESTCONF based Management", draft-ietf-netconf-zerotouch-09 (work in progress), July 2016.

[I-D.lear-mud-framework]
Lear, E., "Manufacturer Usage Description Framework", draft-lear-mud-framework-00 (work in progress), January 2016.

[I-D.richardson-anima-state-for-joinrouter]
Richardson, M., "Considerations for stateful vs stateless join router in ANIMA bootstrap", draft-richardson-anima-state-for-joinrouter-01 (work in progress), July 2016.

[imprinting]
Wikipedia, , "Wikipedia article: Imprinting", July 2015, <https://en.wikipedia.org/wiki/Imprinting_(psychology)>.

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

[RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals", RFC 7575, DOI 10.17487/RFC7575, June 2015, <http://www.rfc-editor.org/info/rfc7575>.

[Stajano99theresurrecting] Stajano, F. and R. Anderson, "The resurrecting duckling: security issues for ad-hoc wireless networks", 1999, <https://www.cl.cam.ac.uk/~fms27/papers/1999-StajanoAnd-duckling.pdf>.

Authors' Addresses

Max Pritikin
Cisco

Email: pritikin@cisco.com

Michael C. Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca
URI: http://www.sandelman.ca/

Michael H. Behringer
Cisco

Email: mbehring@cisco.com

Steinthor Bjarnason
Cisco

Email: sbjarnas@cisco.com

Kent Watsen
Juniper Networks

Email: kwatsen@juniper.net