Operational Considerations for Manufacturer Authorized Signing Authority

draft-richardson-anima-masa-considerations-01

Abstract

This document describes a number of operational modes that a BRSKI Manufacturer Authorized Signing Authority (MASA) may take on.

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

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 June 6, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of
1. Introduction

[I-D.ietf-anima-bootstrapping-keyinfra] introduces a mechanism for new devices (called pledges) to be onboarded into a network without intervention from an expert operator.

This mechanism leverages the pre-existing relationship between a device and the manufacturer that built the device. There are two aspects to this relationship: the provision of an identity for the device by the manufacturer (the IDevID), and a mechanism which convinces the device to trust the new owner (the [RFC8366] voucher).

The manufacturer, or their designate, is involved in both aspects of this process. This requires the manufacturer to operate a significant process for each aspect.

This document offers a number of operational considerations for each aspect.
The first aspect is the device identity in the form of an [ieee802-1AR] certificate that is installed at manufacturing time in the device. The first section of this document deals with operational considerations of building this public key infrastructure.

The second aspect is the use of the Manufacturer Authorized Signing Authority (MASA), as described in [I-D.ietf-anima-bootstrapping-keyinfra] section 2.5.4. The device needs to have the MASA anchor built in; the exact nature of the anchor is subject to a many possibilities. The second section of this document deals with a number of options for architecting the security of the MASA relationship.

There are some additional considerations for a MASA that deals with constrained vouchers as described in [I-D.ietf-anima-constrained-voucher]. In particular in the COSE signed version, there are no PKI structure included in the voucher mechanism, so cryptographic hygiene needs a different set of tradeoffs.

2. Operational Considerations for Manufacturer IDevID Public Key Infrastructure

The manufacturer has the responsibility to provision a keypair into each device as part of the manufacturing process. There are a variety of mechanisms to accomplish this, which this document will overview.

There are two fundamental ways to generate IDevID certificates for devices:

1) generating a private key on the device, creating a Certificate Signing Request (or equivalent), and then returning a certificate to the device.

2) generating a private key outside the device, signing the certificate, and the installing both into the device.

There is a third situation where the IDevID is provided as part of a Trusted Platform Module (TPM), in which case the TPM vendor may be making the same tradeoffs. Or the mechanisms to install the certificate into the TPM will use TPM APIs.
2.1. On-device private key generation

Generating the key on-device has the advantage that the private key never leaves the device. The disadvantage is that the device may not have a verified random number generator.

There are a number of options of how to get the public key securely from the device to the certification authority. This transmission must be done in an integral manner, and must be securely associated with the assigned serial number. The serial number goes into the certificate, and the resulting certificate needs to be loaded into the manufacturer’s asset database. This asset database needs to be shared with the MASA.

One way to do the transmission is during a manufacturing during a Bed of Nails (see [BedOfNails]) or Boundary Scan. There are other ways that could be used where a certificate signing request is sent over a special network channel after the system has been started the first time. There are risks with these methods, as an attacker with physical access may be able to put device back into this mode afterwards.

2.2. Off-device private key generation

Generating the key off-device has the advantage that the randomness of the private key can be better analyzed. As the private key is available to the manufacturing infrastructure, the authenticity of the public key is well known ahead of time. A serial number for the device can be assigned and placed into a certificate. The private key and certificate could be programmed into the device along with the initial bootloader firmware in a single step.

The major downside to generating the private key off-device is that it could be seen by the manufacturing infrastructure. This makes the manufacturing infrastructure a high-value attack target, so mechanisms that keep the private key secure within the manufacturing process, yet allow the device to get access to the private key would be advantageous.

Per-device keys would be ideal, but that would require that an individual per-device key be provisioned in seperately. An alternate mechanism would be that a constant decryption key is kept in the firmware, but this provides little protection against an attack on the manufacturing infrastructure unless the firmware is loaded in a completely different place than the keypair.
2.3. Public Key infrastructure for IDevID

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

The root private key should be kept offline, quite probably in a Hardware Security Module if financially feasible. If not, then it should be secret-split across seven to nine people, with a threshold of four to five people. The split secrets should be kept in geographically diverse places if the manufacturer has operations in multiple places.

Ongoing access to the root-CA is important, but not as critical as access to the MASA key.

The root CA is then used to sign a number of intermediate entities. If manufacturing occurs in multiple factories, then an intermediate CA for each factory is appropriate. It is also reasonable to use different intermediate CAs for different product lines. It may also be valuable to split IDevID certificates across intermediate CAs in a round-robin fashion for products with high volumes.

Cycling the intermediate CAs after a period of a few months or so is a quite reasonable strategy. The intermediate CAs private key may be destroyed after it signed some number of IDevIDs, and a new key generated. The IDevID certificates have very long (ideally infinite) validity lifetimes for reasons that [ieee802-1AR] explains, but once the certificates have been created the intermediate CA has no further obligations as neither CRLs nor OCSP are appropriate.

In all cases the serialNumber embedded in the certificate must be unique across all products produced by the manufacturer. This suggests some amount of structure to the serialNumber, such that different intermediate CAs do not need to coordinate when issuing certificates.

3. Operational Considerations for Manufacturer Authorized Signing Authority (MASA)

The manufacturer needs to make a Signing Authority available to new owners so that they may obtain [RFC8366] format vouchers to prove ownership. This section initially assumes that the manufacturer will provide this Authority internally, but subsequent sections deal with some adjustments when the authority is externally run.
In any case, the manufacturer SHOULD plan for a situation where the manufacturer is no longer able or interested in running the Authority: this does not have to be an unhappy situation of the manufacturer going out of business. It could be a happy event where the manufacturer goes through a merge or acquisition and it makes sense to consolidate the Signing Authority in another part of the organization.

A plan to escrow the signing keys SHOULD be detailed, and it is likely that customers will want to review it for high-value products.

The anchors for the MASA need to be "baked-in" to the device firmware so that they are always available to validate vouchers. In order to avoid locking down a single validation key, a PKI infrastructure is appropriate. Note that constrained devices without code space to parse and validate a public key certificate chain require different considerations, and this document does not (yet) provide that consideration.

There are many ways to construct a resilient PKI to sign vouchers.

3.1. Self-contained multi-product MASA

The most obvious is to just create a new offline CA, have it periodically sign a new End-Entity (EE) Certificate with an online private key, and use that to sign voucher requests. The entity used to sign [RFC8366] format vouchers does not need to be a certificate authority.

The public key of the offline CA is then built-in to the firmware of the device, providing a trust anchor with which to validate vouchers. The End-Entity certificate used to sign the voucher is included in the certificate set in the CMS structure that is used to sign the voucher. The root CA’s trust anchor should _also_ be included, even though it is self-signed, as this permits auditing elements in a Registrar to validate the End-Entity Certificate.

The inclusion of the full chain also supports a Trust-on-First-Use (TOFU) workflow for the manager of the Registrar: they can see the trust anchor chain and can compare a fingerprint displayed on their screen with one that could be included in packaging or other sales channel information.

When building the MASA public key into a device, only the public key contents matter, not the structure of the self-signed certificate itself. Using only the public key enables a MASA architecture to evolve from a single self-contained system into a more complex architecture later on.
3.2. Self-contained per-product MASA

A simple enhancement to the previous scenario is to have a unique MASA offline key for each product line. This has a few advantages:

- if the private keys are kept separately (under different encryption keys), then compromise of a single product line's MASA does not compromise all products.

- if a product line is sold to another entity, or if it has to go through an escrow process due to the product going out of production, then the process affects only a single product line.

- it is safe to have serialNumber duplicated among different product lines since a voucher for one product line would not validate on another product line.

The disadvantage is that it requires a private key to be stored per product line, and most large OEMs have many dozens of product lines. If the keys are stored in a single Hardware Security Module (HSM), with the access to it split across the same parties, then some of the cryptographic advantages of different private keys go away, as a compromise of one key likely compromises them all. Given a HSM, the most likely way a key is compromised is by an attacker getting authorization on the HSM through theft or coercion.

The use of per-product MASA signing keys is encouraged.

3.3. Per-product MASA keys intertwined with IDevID PKI

The IDevID certificate chain (the intermediate CA and root CA that signed the IDevID certificate) should be included in the device firmware so that they can be communicated during the BRSKI-EST exchange.

Since they are already present, why not use of them as the MASA trust anchor?

A voucher needs to be signed by recognized End-Entity, which has been authorized to be a voucher signer. The challenge with combining it into the IDevID PKI is making sure that only an authorized entity can sign the vouchers. This suggests that it can not be the same intermediate CA that is used to sign the IDevID, since that CA should have the authority to sign vouchers. If it did, then the End-Entities that it created, the IDevID for devices, would then be able to sign vouchers, which would not be an appropriate authorization.
The PKI root CA therefore needs to sign an intermediate CA, or End-Entity certificate with an extension OID that is specific for Voucher Authorization. This is easy to do as policy OIDs can be created from Private Enterprise Numbers. There is no need for standardization, as the entity doing the signing is also creating the verification code. If the entire PKI operation was outsource, then there would be a benefit for standardization.

3.4. Rotating MASA authorization keys

As a variation of the scenario described in Section 3.2, there could be multiple Signing Authority keys per product line. They could be rotated though in some deterministic order. For instance, serial numbers ending in 0 would have MASA key 0 embedded in them at manufacturing time. The asset database would have to know which key that corresponded to, and it would have to produce vouchers using that key.

There are significant downsides to this mechanism:

- all of the MASA signing keys need to be online and available in order to respond to any voucher request
- it is necessary to keep track of which device trust which key in the asset database

There is no obvious advantage to doing this if all the MASA signing private keys are kept in the same device, under control of the same managers. But if the keys are spread out to multiple locations and are under control of different people, then there may be some advantage. A single MASA signing authority key compromise does not cause a recall of all devices, but only the portion that had that key embedded in it.

The relationship between signing key and device could be temporal: all devices made on Tuesday could have the same key, there could be hundreds of keys, each one used only for a few hundred devices. There are many variations possible.

The major advantage comes with the COSE signed constrained-vouchers described in [I-D.ietf-anima-constrained-voucher]. In this context there isn’t space in the voucher for a certificate chain, nor is there code space in the device to validate a certificate chain. The (public) key used to sign is embedded directly in the firmware of each device without the benefit of any public key infrastructure to allow indirection of the key.
4. Operational Considerations for Constrained MASA
   TBD

5. Operational Considerations for creating Nonceless vouchers
   TBD

6. Privacy Considerations
   YYY

7. Security Considerations
   ZZZ

8. IANA Considerations
   This document makes no IANA requests.

9. Acknowledgements
   Hello.

10. Changelog

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-30 (work in progress), November 2019.

[I-D.ietf-anima-constrained-voucher]
   Richardson, M., Stok, P., and P. Kampanakis, "Constrained
   Voucher Artifacts for Bootstrapping Protocols", draft-
   ietf-anima-constrained-voucher-05 (work in progress), July
   2019.

[ieee802-1AR]
   IEEE Standard, ., "IEEE 802.1AR Secure Device Identifier",

Richardson
Expires June 6, 2020
11.2. Informative References


"Enrollment over Secure Transport", RFC 7030,
DOI 10.17487/RFC7030, October 2013,

Author’s Address

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca