Workgroup: drip Working Group
Internet-Draft: draft-ietf-drip-registries-00
Published: 27 January 2022
Intended Status: Standards Track
Expires: 31 July 2022
Authors: A. Wiethuechter
AX Enterprize, LLC
S. Card
AX Enterprize, LLC
R. Moskowitz
HTT Consulting
J. Reid
RTFM llp

DRIP Registries

Abstract

This document creates the DRIP DET registration and discovery ecosystem. This includes all components in the ecosystem (e.g., RAA, HDA, UA, GCS, USS). The registration process will use the Extensible Provisioning Protocol (EPP) and other protocols. The discovery process will leverage DNS and DNSSEC and related technology. The DETs can be registered with as their "raw public keys" or in X.509 certificates.

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 31 July 2022.

Copyright Notice

Table of Contents

1. Introduction

Registries are fundamental to RID. Only very limited information can be Broadcast, but extended information is sometimes needed. The most essential element of information sent is the UAS ID itself, the unique key for lookup of extended information in registries.

While it is expected that registry functions will be integrated with USS, who will provide them is not yet determined in most, and is expected to vary between, jurisdictions. However this evolves, the essential registry functions, starting with management of identifiers, are expected to remain the same, so are specified herein.

While most data to be sent via Broadcast or Network RID is public, much of the extended information in registries will be private. Thus AAA for registries is essential, not just to ensure that access is granted only to strongly authenticated, duly authorized parties, but also to support subsequent attribution of any leaks, audit of who accessed information when and for what purpose, etc. As specific AAA requirements will vary by jurisdictional regulation, provider philosophy, customer demand, etc., they are left to specification in policies, which should be human readable to facilitate analysis and discussion, and machine readable to enable automated enforcement, using a language amenable to both, e.g., XACML.

The intent of the negative and positive access control requirements on registries is to ensure that no member of the public would be hindered from accessing public information, while only duly authorized parties would be enabled to access private information. Mitigation of Denial of Service attacks and refusal to allow database mass scraping would be based on those behaviors, not on identity or role of the party submitting the query per se, but querant identity information might be gathered (by security systems protecting DRIP implementations) on such misbehavior.

Registration under DRIP is vital as the worry of collisions in the hash portion of the DET. Forgery of the DET is still possible, but including it as a part of a public registration mitigates a lot of the risk. This document creates the DRIP DET registration and discovery ecosystem. This includes all components in the ecosystem (e.g., RAA, HDA, UA, GCS, USS). The registration process will use the Extensible Provisioning Protocol (EPP) and other protocols. The discovery process will leverage DNS and DNSSEC and related

technology. The DETs can be registered with as their "raw public keys" or in X.509 certificates.

2. Terminology

2.1. Required Terminology

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

2.2. Definitions

See [drip-requirements] for common DRIP terms.

HDA: Hierarchial HIT Domain Authority. The 16 bit field identifying the HIT Domain Authority under a RAA.

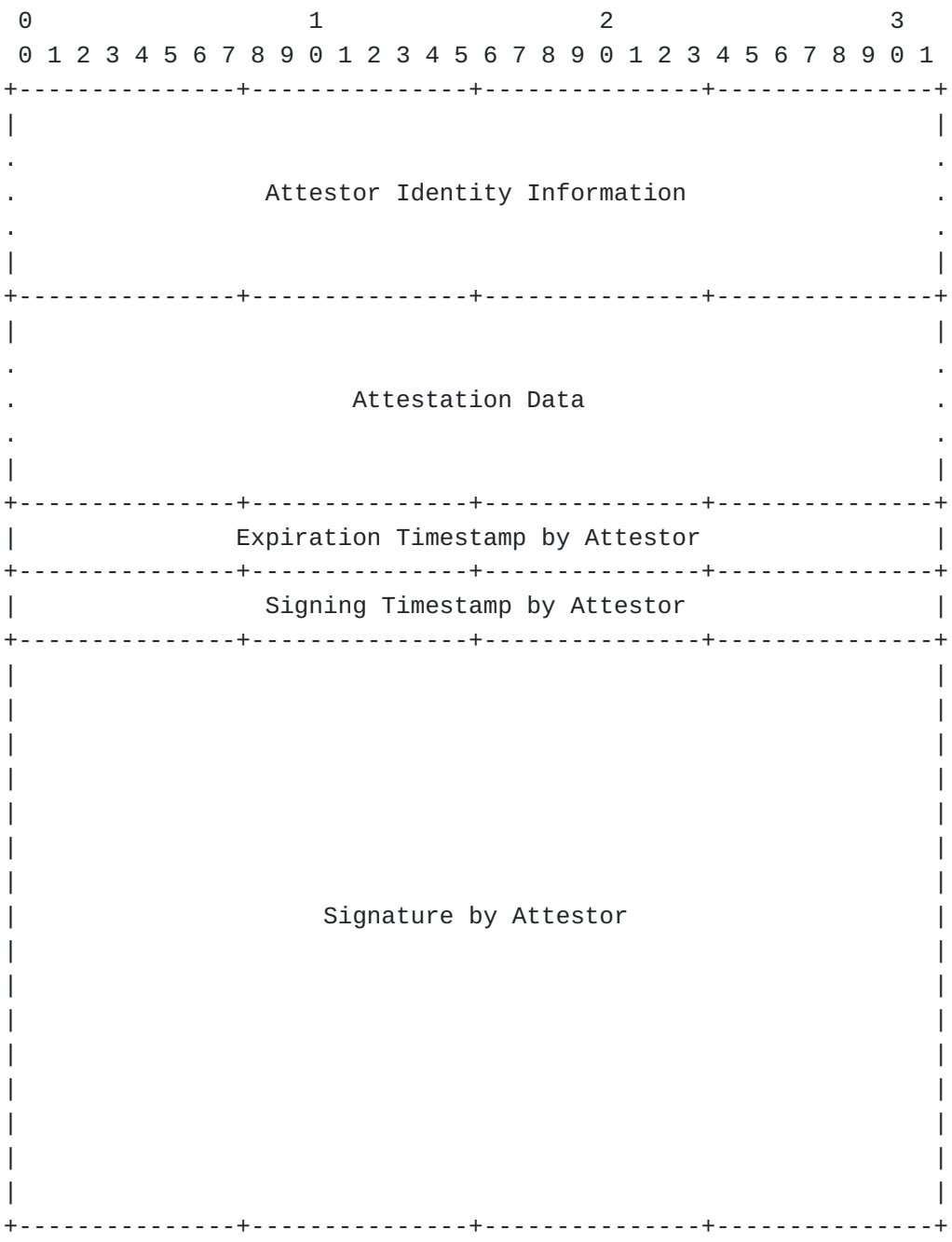
HID: Hierarchy ID. The 32 bit field providing the HIT Hierarchy ID.

RAA: Registered Assigning Authority. The 16 bit field identifying the Hierarchical HIT Assigning Authority.

3. DRIP Attestations & Certificates

3.1. Attestation Structure

All Attestations and Certificates under DRIP share the following format:



Attestor Identity Information: (0, 16-bytes or 120-bytes)
    Field containing Attestor Identity Information in various forms.

Attestation Data:
    A field of variable length containing the attestation data.

Expiration Timestamp by Attestor (4 bytes):
    Timestamp denoting recommended time to trust data to.

Signing Timestamp by Attestor (4 bytes):
    Current time at signing.

```
Attestor Signature (64 bytes):
    Signature over preceding fields using the keypair of
    the Attestor.
```

Figure 1: Attestation Structure

3.1.1. Attestor Identity Information

This can be any one of the following:

1. None
2. Attestor HHIT: 16-bytes
3. Attestor SelfAttestation: 120-bytes

A specific definition of an Attestation or Certificate defines which of these are used.

Two Attestation's remove this field: MutualAttestation Section 3.2.4 and LinkAttestation Section 3.2.5 as their definition clearly states that the signer is the second party with their HHIT or SelfAttestation already embedded in the Attestation Data.

3.1.2. Attestation Data

The data being attested to. It can be one of the following forms:

1. Claims
2. Assertions
3. Attestations

This field is variable length with no limit and specific definitions of an Attestation or Certificate indicate the fields, size and ordering of any subfields.

3.1.3. Expiration Timestamp

A UTC timestamp set some time into the future to indicate a point the Attestation Structure should not be trusted.

3.1.4. Signing Timestamp

A UTC timestamp set to the time when the Attestation Structure was signed.

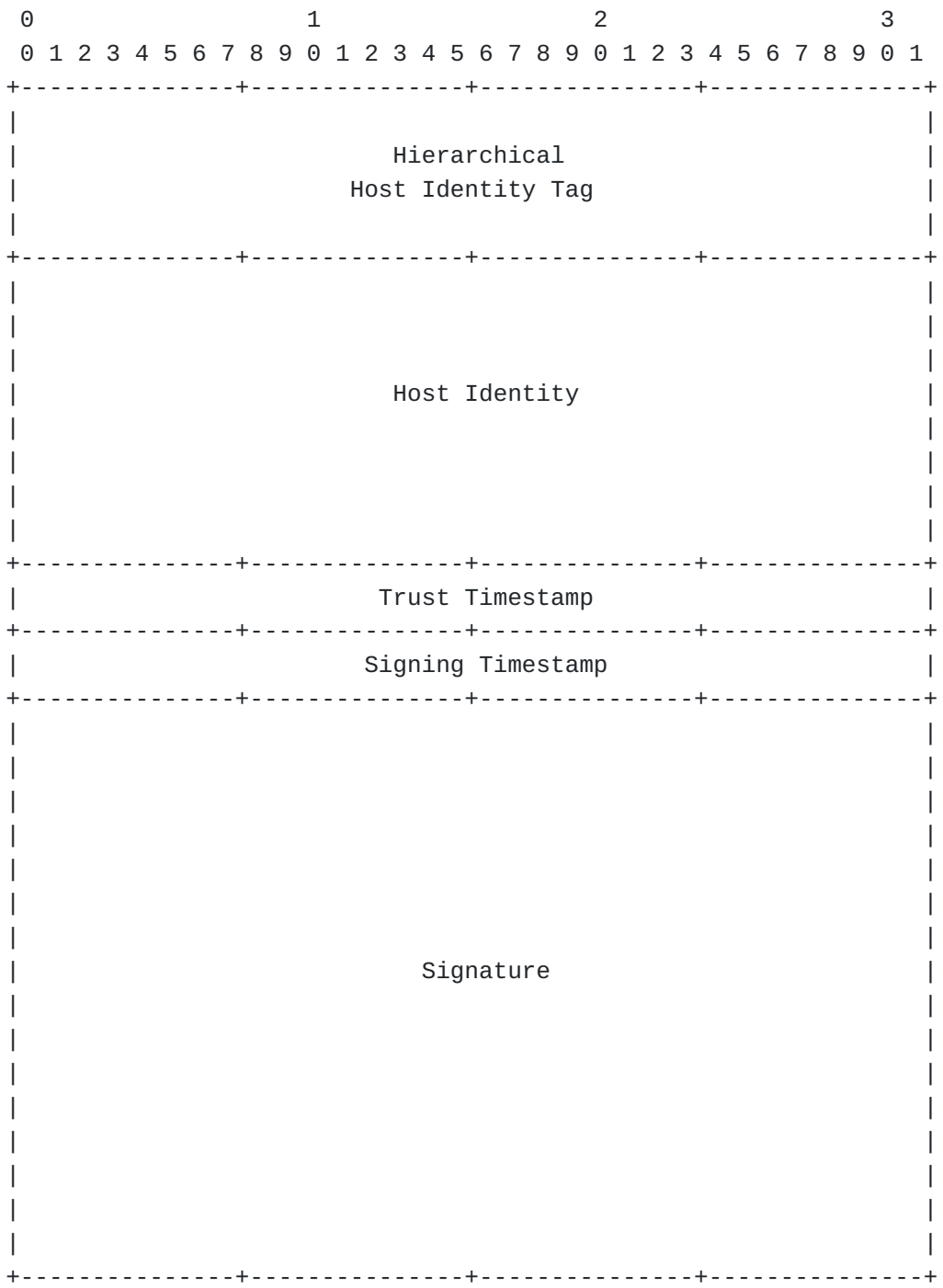
3.1.5. Signature

An EdDSA25519 signature using the signing parties private key over the preceding fields in the Attestation Structure.

3.2. Attestations

3.2.1. Self-Attestation (SA-xx)

The only attestation to use a claim (the Host Identity) in the Attestation Data with the HHIT acting as the Attestor Identity Information.

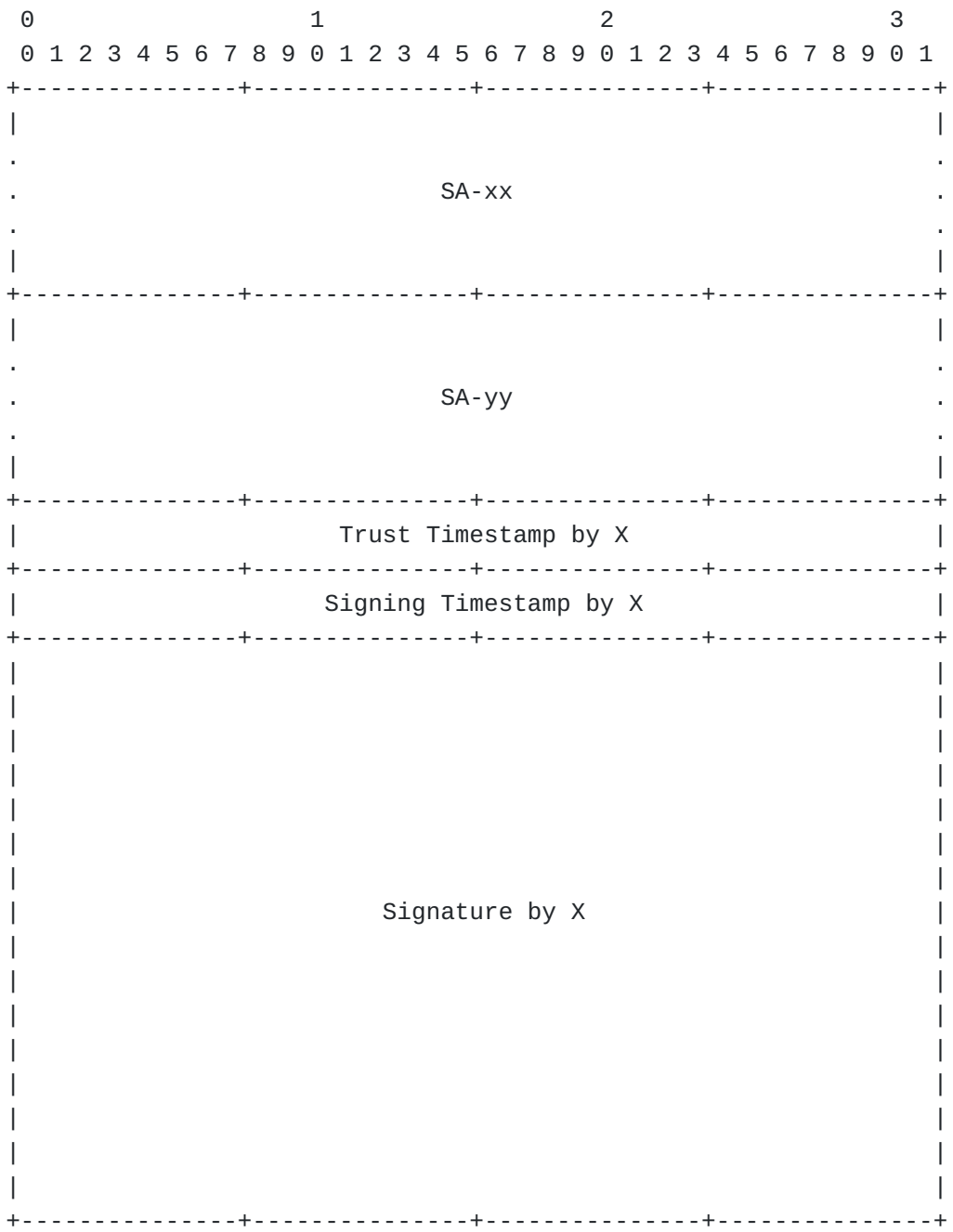


Length = 120-bytes

Figure 2: DRIP Self-Attestation

3.2.2. Attestation (A-xy)

(Editors Note: blurb here?)



Length = 312-bytes

Figure 3: DRIP Attestation

3.2.3. Concise Attestation (CA-xy)

In constrained environments and when there is the guarantee of being able to lookup the HHITs to obtain HIs this attestation can be used.

Length = 104-bytes

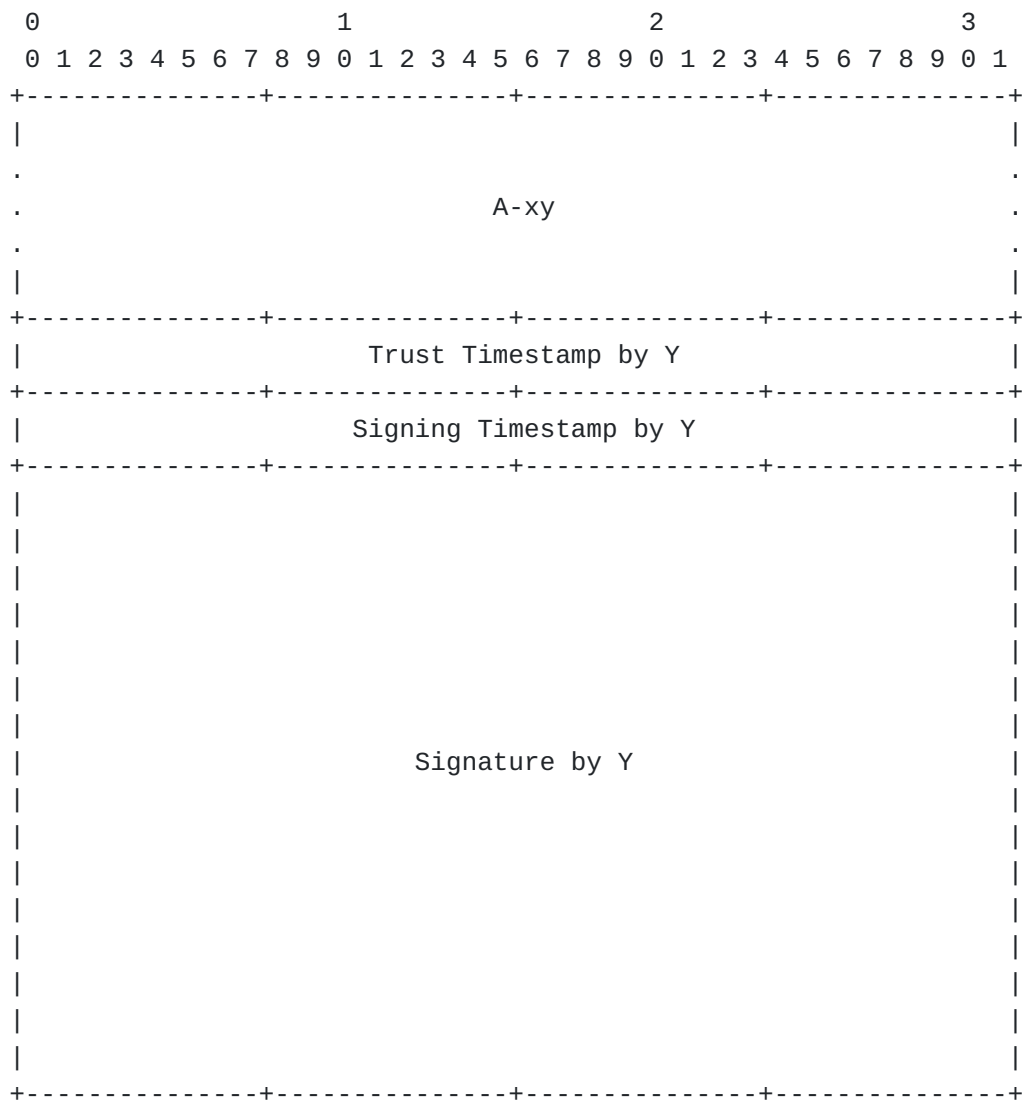
Figure 4: DRIP Concise Attestation

3.2.4. Mutual Attestation (MA-xy)

An attestation that perform a sign over an existing Attestation where the signer is the second party of the embedded attestation.

This Attestation is one of two that does not fill in the Attestor Identity Information (Section 3.1.1) as the data is already present in the Attestation Data (Section 3.1.2) in the form of Y's SelfAttestation.

The unique size of this attestation (384-bytes) allows for easy detection and subsequent decoding without issue.



Length = 384-bytes

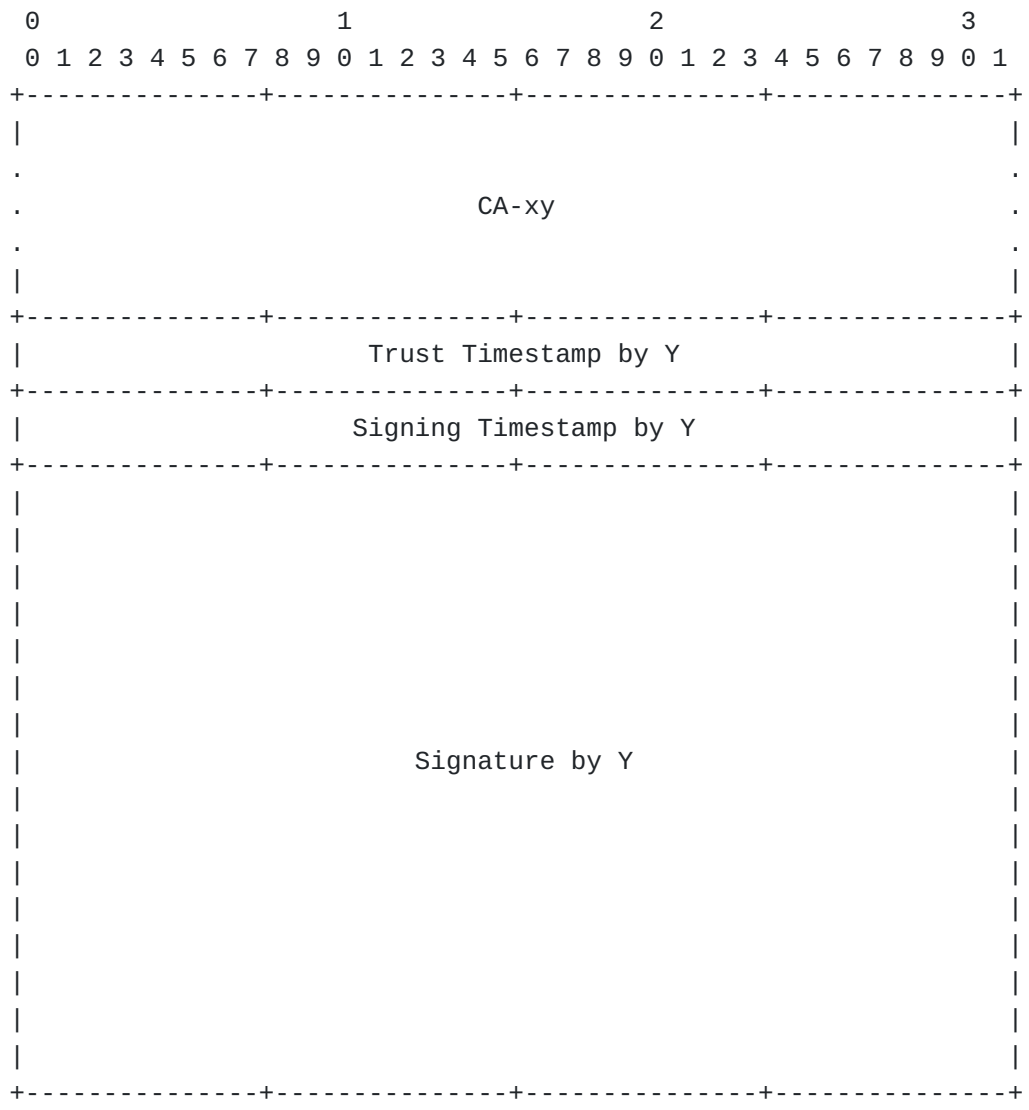
Figure 5: DRIP Mutual Attestation

3.2.5. Link Attestation (LA-xy)

An attestations that perform a sign over an existing ConciseAttestation where the signer is the second party of the embedded attestation.

This Attestation is one of two that does not fill in the Attestor Identity Information (Section 3.1.1) as the data is already present in the Attestation Data (Section 3.1.2) in the form of Y's HHIT.

The unique size of this attestation (176-bytes) allows for easy detection and subsequent decoding without issue.



Length = 176-bytes

Figure 6: DRIP Link Attestation

3.2.6. Broadcast Attestation (BA-xy)

Required by DRIP Authentication Formats for Broadcast RID (Editor Note: add link to draft here) to satisfy [drip-requirements] GEN-1 and GEN-3.

Length = 136-bytes

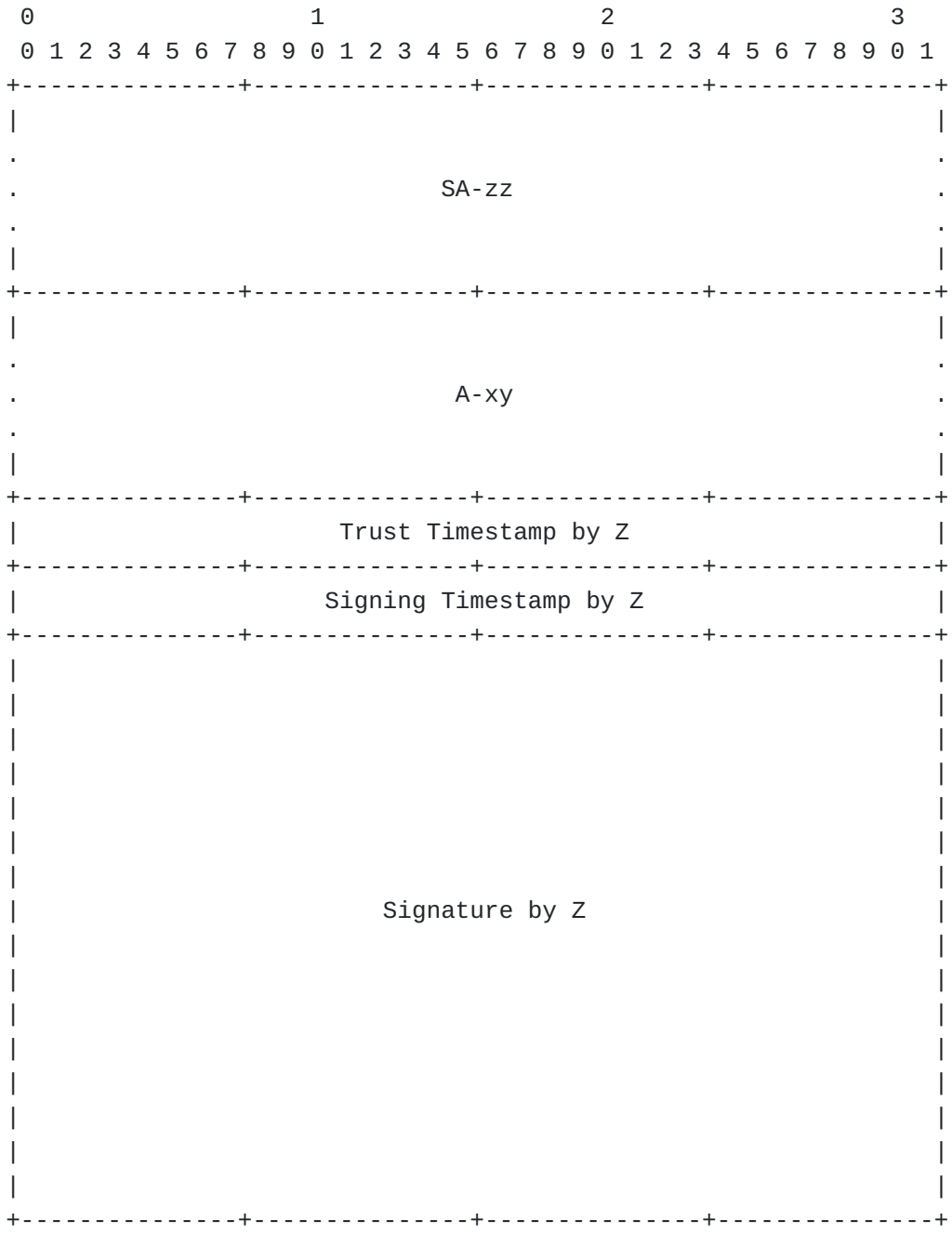
Figure 7: DRIP Broadcast Attestation

3.3. Certificates

In DRIP certificates are signed by a third party that has no stake in the claims/assertions/attestations being attested to.

It is analogous to a third party in legal system that signs a document as a "witness" and bears no responsibility in the document.

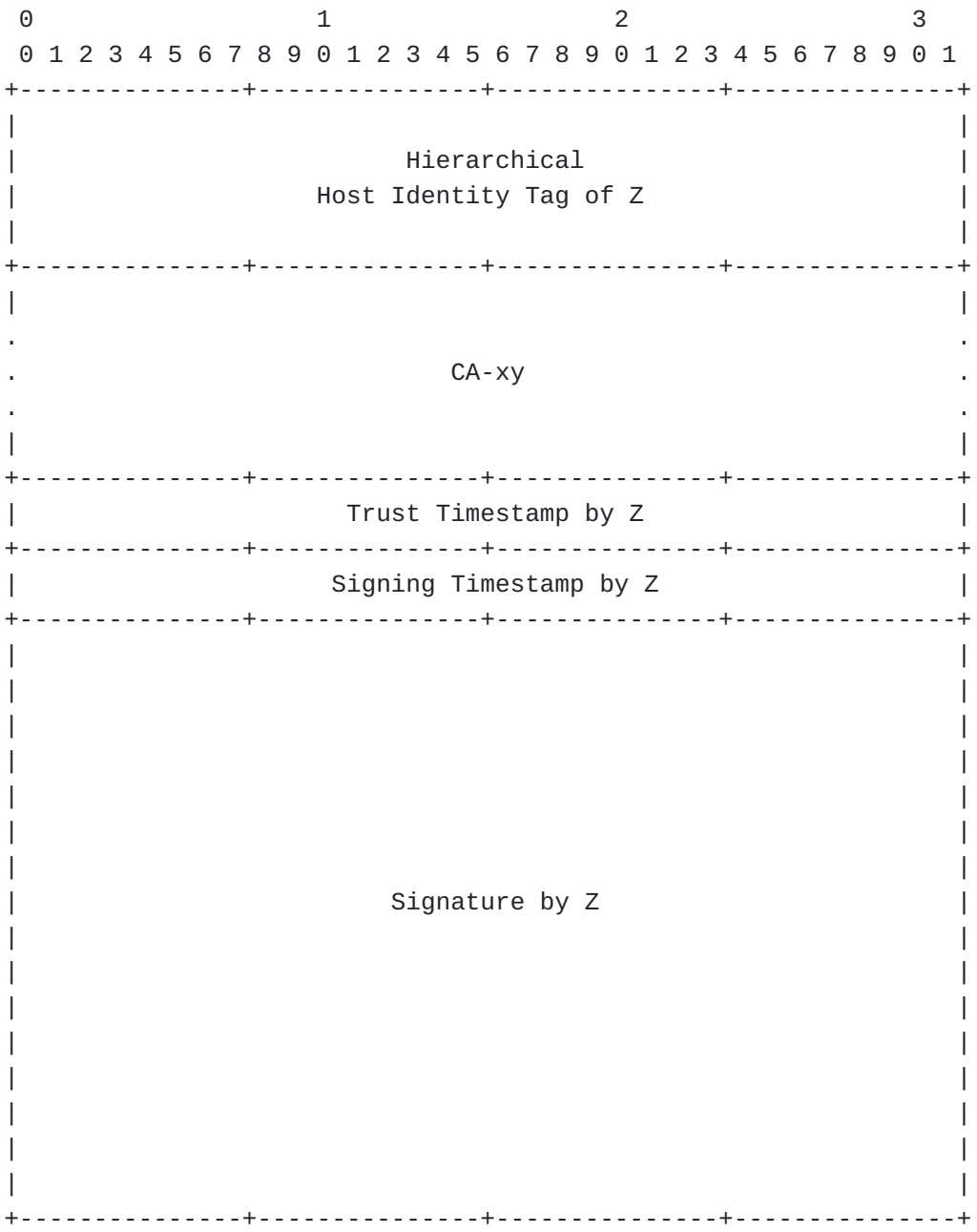
3.3.1. Attestation Certificate (AC-zxy)



Length = 504-bytes

Figure 8: DRIP Attestation Certificate

3.3.2. Concise Certificate (CC-zxy)



Length = 192-bytes

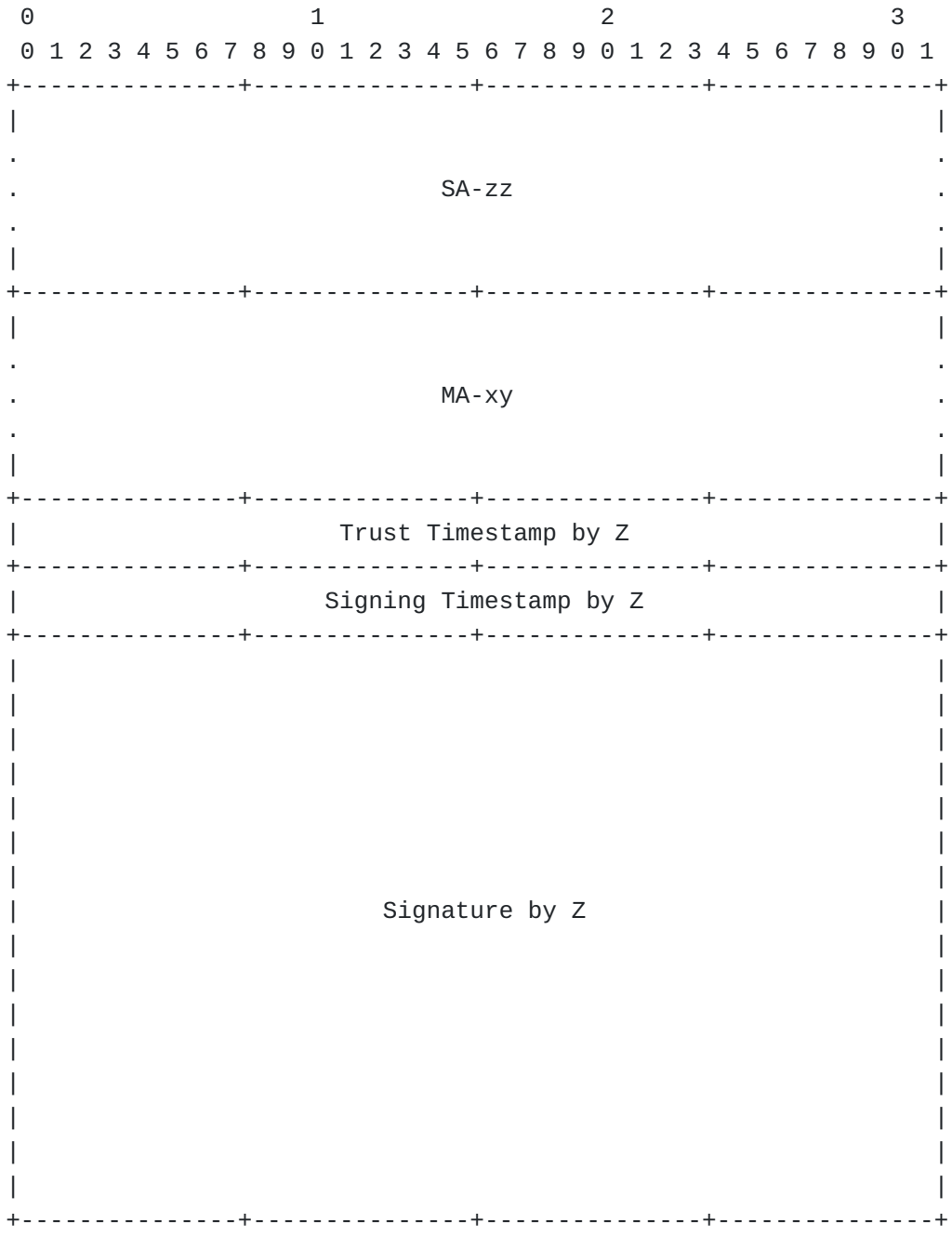
Figure 9: DRIP Concise Certificate

3.3.3. Link Certificate (LC-zxy)

Length = 300-bytes

Figure 10: DRIP Link Certificate

3.3.4. Mutual Certificate (MC-zxy)



Length = 576-bytes

Figure 11: DRIP Mutual Certificate

4. Registries

4.1. Classes

Under DRIP there 3 classes of registries, with specific variants in each.

4.1.1. Root

This is a special registry holding the RAA value of 0 and HDA value of 0. It delegates out RAA values only to registries that wish to act as an RAA.

(Editors Note: we contemplate this is ICAO running this server or federation of them)

4.1.2. Registered Assigning Authorities

RAA's are the upper hierarchy in DRIP. Most are contemplated to be Civil Aviation Authorities (CAAs) then delegate HDAs to manage their NAS. This is does not preclude other entities to operate an RAA if the Root server allows it.

All RAA's use an HDA value of 0 and have their RAA value delegated to them by the Root.

4.1.2.1. ICAO Registry of Manufacturer's (IRM)

A special RAA that hands out HDA values to participating Manufacturer's that hold an ICAO Manufacturer Code used in ANSI CTA2063-A Serial Numbers.

It is holds the RAA value of 1 and HDA value of 0.

(Editors Note: we contemplate this is ICAO running this server or federation of them)

4.1.3. Hierarchial HIT Domain Authorities

4.1.3.1. Manufacturer's Registry of Aircraft (MRA)

A registry (HDA) run by a manufacturer of UAS systems that participate in Remote ID. Stores UAS Serial Numbers under a specific ICAO Manufacturer Code (assigned to the manufacturer by ICAO).

A DET can be encoded into a Serial Number (Editor Note: link to -uas-rid) and when done so this registry would hold a mapping from the Serial Number to the DET and its artifacts.

Hold RAA value of 1 and HDA values of 1+.

4.1.3.2. Remote ID Registries (RIDR)

Registry that holds the binding between a UAS Session ID (for DRIP the DET) and the UA Serial Number. The Serial Number MUST have its access protected to allow only authorized parties to obtain. The

Serial Number SHOULD be encrypted in a way only the authorized party can decrypt.

As part of the UTM system they also hold a binding between a UAS ID (Serial Number or Session ID) and an Operational Intent.

(Editors Note: these are contemplated to be part of a USS as a function or a standalone SDSP in the UTM system)

Hold RAA values of 2+ and HDA values of 1+.

4.2. Federation

(Editors Note: Due to nature of HHIT we could have multiple registries with same RAA/HDA pairings running and being federated together. How do we handle this?)

5. DRIP Fully Qualified Domain Names

Under DRIP there are a number of FQDN forms used to allow lookups to take place.

(Editor Note: copy in DET Section 5 here)

5.1. Serial Number

Serial Number: 8653FZ2T7B8RA85D19LX
ICAO Mfr Code: 8653
Length Code: F
ID: FZ2T7B8RA85D19LX
FQDN: Z2T7B8RA85D19LX.F.8653.mfr.uas.icao.int

5.2. Reverse SN

(Editors Note: convert SN to DET format then perform reverse DET?)

5.3. DET

DET: 2001:0030:00a0:0145:a3ad:1952:0ad0:a69e
ID: a3ad:1952:0ad0:a69e
OGA: 5
HDA: 0014 = 20
RAA: 000a = 10
Prefix: 20010030
FQDN: a3ad19520ad0a69e.5.20.10.20010030.det.uas.icao.int

When building a DET FQDN the following two things must be done:

1. The RAA and HDA values MUST be converted from hexadecimal to decimal form

2. The FQDN must be built using the expanded form of the IPv6 address

The prefix is included in the FQDN form to support other potential prefixes being used.

5.4. Reverse DET

```
$ORIGIN  5.4.1.0.0.a.0.0.0.3.0.0.1.0.0.2.ip6.arpa.
e.9.6.a.0.d.a.0.2.5.9.1.d.a.3.a    IN   PTR
```

6. Supported DNS Records

DRIP requires a number of resource records, some specific to certain registries to function.

6.1. HIP RR

All registries will have their own DET associated with them and their respective DNS server will hold a HIP RR that is pointed to by their DET FQDN.

MRA and RIDR servers will also have HIP RRs for their registered parties (aircraft and operators).

6.2. CERT RR

Most attestations can be placed into DNS. An exception to this is the AttestationCertificate made during Session ID registration.

6.3. NS RR

Along with their associated "glue" record (A/AAAA) supports the traversal in DNS across the tree.

1. <mfr.remoteid.aero> on Root points to specific DET FQDN of IRM

2. <icao_mfr_code>.mfr.remoteid.aero on IRM points to specific DET FQDN of MRA

3. <raa_value>.det.remoteid.aero on Root pointing to DET FQDN of matching RAA

4. <hda_value>.<raa_value>.det.remoteid.aero on RAA Registry pointing to DET FQDN of matching HDA

6.4. AAAA RR

DRIP requires the use of IPv6.

6.5. SVR RR

TODO - points to server for RDAP stuff

6.6. TLSA RR

Raw key format here; for DTLS support. RFC6698

7. Registry Operations

(Editors Note: General processing instructions here?)

As a general rule the following processing performed for any registration operation:

1. Verify SelfAttestation of registering party
2. Populate DNS with required/optional records
3. Populate Database with PII and other info
4. Generate and return required/optional Attestations

7.1. Registering an RAA

Specifically handled by the Root Registry (Section 4.1.1).

7.1.1. Inputs

Required:

1. SelfAttestation of RAA
2. IP Address of RAA

7.1.2. DNS Entries

Required on Root:

NS RR = <raa_value>.det.remoteid.aero NS <raa_det_fqdn>

AAAA RR = <raa_det_fqdn> AAAA ...

CERT RR = ???

Required on RAA:

HIP RR = <raa_det_fqdn> HIP ...

CERT RR = ???

7.1.3. Database Entries

7.1.4. Outputs

7.2. Registering an IRM

Specifically handled by the Root Registry (Section 4.1.1).

7.2.1. Inputs

Required:

1. Self-Attestation of IRM
2. IP Address of IRM

7.2.2. DNS Entries

Required on Root:

NS RR = mfr.remoteid.aero NS <irm_det_fqdn>

NS RR = 1.det.remoteid.aero NS <irm_det_fqdn>

AAAA RR = <irm_det_fqdn> AAAA ...

CERT RR = ???

Required on IRM:

HIP RR = <irm_det_fqdn> HIP ...

CERT RR = ???

7.2.3. Database Entries

7.2.4. Outputs

Required:

1. Attestation: Root on IRM

7.3. Registering an HDA

Specifically handled by an RAA (Section 4.1.2).

7.3.1. Inputs

Required:

1. Self-Attestation of HDA

2. IP Address of HDA

7.3.2. DNS Entries

Required on RAA:

NS RR = <hda_value>.<raa_value>.det.remoteid.aero NS <hda_det_fqdn>

AAAA RR = <hda_det_fqdn> AAAA ...

CERT RR = ???

Required on HDA:

HIP RR = <hda_det_fqdn> HIP ...

7.3.3. Database Entries

7.3.4. Outputs

7.4. Registering an MRA

Specifically handled by the IRM Registry (Section 4.1.2.1).

7.4.1. Inputs

Required:

1. ICAO Manufacturer Code
2. Self-Attestation of MRA
3. IP Address of MRA

7.4.2. DNS Entries

Required on IRM:

NS RR = <icao_mfr_code>.mfr.remoteid.aero NS <mra_det_fqdn>

NS RR = <hda_value>.1.det.remoteid.aero NS <mra_det_fqdn>

AAAA RR = <mra_det_fqdn> AAAA ...

CERT RR = ???

Required on MRA:

HIP RR = <mra_det_fqdn> HIP ...

CERT RR = ???

7.4.3. Database Entries

(HDA value, MRA Details)

7.4.4. Outputs

Required:

1. Attestation: IRM on MRA

7.5. Registering a Serial Number

Specifically handled by a MRA (Section 4.1.3.1).

7.5.1. Inputs

Required:

1. Serial Number
2. Aircraft Metadata

Optional:

1. SelfAttestation: Aircraft on Aircraft (if DET encoded)

7.5.2. DNS Entries

Required on MRA:

A/AAAA with Serial Number FQDN (Section 5.1)

Optional on MRA:

HIP RR of Aircraft with DET FQDN (Section 5.3) (<sn_det_fqdn> HIP ...)

CERT RRs of SelfAttestation and BroadcastAttestation

7.5.3. Database Entries

(Serial Number, [DET], Metadata, [SelfAttestation])

7.5.4. Outputs

Optional:

1. BroadcastAttestation: Mfr on Aircraft

7.6. Registering an Operator

Specifically handled by a RIDR (Section 4.1.3.2).

7.6.1. Inputs

Required:

1. SelfAttestation: Operator on Operator
2. Operator PII

Optional: TODO

7.6.2. DNS Entries

Optional on RIDR:

HIP RR of Operator

CERT RRs SelfAttestation of Operator, A-ro

7.6.3. Database Entries

TODO

7.6.4. Outputs

Required:

1. Attestation (A-ro) - using SA-rr and SA-oo

Optional:

1. ConciseAttestation (CA-ro) - using SA-oo
2. BroadcastAttestation (BA-ro) - using SA-oo

7.7. Registering a Session ID

Specifically handled by a RIDR (Section 4.1.3.2).

7.7.1. Inputs

Required:

1. Attestation: Registry on Operator
2. Attestation: Operator on Aircraft
3. UAS Serial Number

Optional:

1. ConciseAttestation: Operator on Aircraft
2. MutualAttestation: Operator on Aircraft
3. LinkAttestation: Operator on Aircraft
4. Operational Intent ID (GUFI)

7.7.2. DNS Entries

Required on RIDR:

HIP RR of Aircraft with DET FQDN (Section 5.3) (<session_det_fqdn> HIP ...)

CERT RRs for SelfAttestation of Aircraft, BroadcastAttestation

7.7.3. Database Entries

(Session ID, Serial Number, GUFI, A-oa, BA-ra, AC-roa)

7.7.4. Outputs

Required:

1. BroadcastAttestation (BA-ra) - generated using the embedded SA-aa from A-oa
2. AttestationCertificate (AC-roa) - using A-oa

Optional:

1. MutualCertificate (MC-roa) - using MA-oa
2. ConciseCertificate (CC-roa) - using CA-oa
3. LinkCertificate (LC-roa) - using LA-oa
4. BroadcastAttestation's of parent Registries in chain

8. Provisioning

Under DRIP UAS RID a special provisioning procedure is required to properly generate and distribute the certificates and attestations to all parties in the USS/UTM ecosystem using DRIP RID.

Keypairs are expected to be generated on the device hardware it will be used on. Due to hardware limitations (see Section 10) and connectivity it is acceptable under DRIP RID to generate keypairs

for the Aircraft on Operator devices and later securely inject them into the Aircraft (as defined in Section 8.6.2). The methods to securely inject and store keypair information in a "secure element" of the Aircraft is out of scope of this document.

8.1. Overview of Transactions

In DRIP, each Operator MUST generate a Host Identity of the Operator (HIo) and derived Hierarchical HIT of the Operator (HHITo). These are registered with a Private Information Registry along with whatever Operator data (inc. PII) is required by the cognizant CAA and the registry. In response, the Operator will obtain an attestation from the Registry, Attestation: Registry on Operator (A-ro), signed with the Host Identity of the Registry private key (HIr(priv)) proving such registration.

An Operator may now claim one or more UA.

* An Operator MUST generate a Host Identity of the Aircraft (HIa) and derived Hierarchical HIT of the Aircraft (HHITa)

* Create an attestation from the Operator on the Aircraft (A-oa) signed with the Host Identity of the Operator private key (HIo(priv)) to associate the UA with its Operator

* Register them with a Private Information Registry along with whatever UAS data is required by the cognizant CAA and Registry

* Obtain an attestation from the Registry on the Operator and Aircraft ("AC-roa") signed with the HIr(priv) proving such registration

* And obtain a broadcast attestation from the Registry on the Aircraft (BA-ra) signed with HIr(priv) proving UA registration in that specific registry while preserving Operator privacy.

The operator then MUST provision the UA with HIa, HIa(priv), HHITa and B-Ara.

* UA engaging in Broadcast RID MUST use HIa(priv) to sign Authentication Messages and MUST periodically broadcast BA-ra.

* UAS engaging in Network RID MUST use HIa(priv) to sign Authentication Messages.

* Observers MUST use HIa from received BA-ra to verify received Broadcast RID Authentication messages.

* Observers without Internet connectivity MAY use BA-ra to identify the trust class of the UAS based on known registry vetting.

*Observers with Internet connectivity MAY use HHITa to perform lookups in the Public Information Registry and MAY then query the Private Information Registry which MUST enforce AAA policy on Operator PII and other sensitive information

8.2. HHIT Delegation

Under the FAA [NPRM], it is expecting that IDs for UAS are assigned by the UTM and are generally one-time use. The methods for this however are unspecified leaving two options.

1. The entity generates its own HHIT, discovering and using thr RAA and HDA for the target Registry. The method for discovering a Registry's RAA and HDA is out of scope here. This allows for the device to generate an HHIT to send to the Registry to be accepted (thus generating the required Host Identity Claim) or denied.

2. The entity sends to the Registry its HI for it to be hashed and result in the HHIT. The Registry would then either accept (returning the HHIT to the device) or deny this pairing.

In either case the Registry must decide on if the HI/HHIT pairing is valid. This in its simplest form is checking the current Registry for a collision on the HHIT.

Upon accepting a HI/HHIT pair the Registry MUST populate the required the DNS serving the HDA with the HIP RR and other relevant RR types (such as TXT and CERT). The Registry MUST also generate the appropriate Attestation for the given operation.

If the Registry denied the HI/HHIT pair, because there was a HHIT collision or any other reason, the Registry MUST signal back to the device being provisioned that a new HI needs to be generated.

8.3. Registry

(Editor Note: this should break down the individual registrations between Root/RAA, RAA/HDA and their special variants).

TODO

DRIP UAS RID defines two levels of hierarchy maintained by the Registration Assigning Authority (RAA) and HHIT Domain Authority (HDA). The authors anticipate that an RAA is owned and operated by a regional CAA (or a delegated party by an CAA in a specific airspace region) with HDAs being contracted out. As such a chain of trust for registries is required to ensure trustworthiness is not compromised. More information on the registries can be found in [hhit-registries].

Both the RAA and HDA generate their own keypairs and self-signed attestations (SelfAttestation: RAA on RAA and SelfAttestation: HDA on HDA respectively). The HDA sends to the RAA its self-signed attestation to be added into the RAA DNS.

The RAA confirms the attestation received is valid and that no HHIT collisions occur before added a HIP RR to its DNS for the new HDA. An Attestation: RAA on HDA (A-rh) is sent as a confirmation that provisioning was successful.

The HDA is now a valid "Registry" and uses its keypair and SelfAttestation: HDA on HDA (SA-hh) with all provisioning requests from downstream.

8.4. Manufacturer

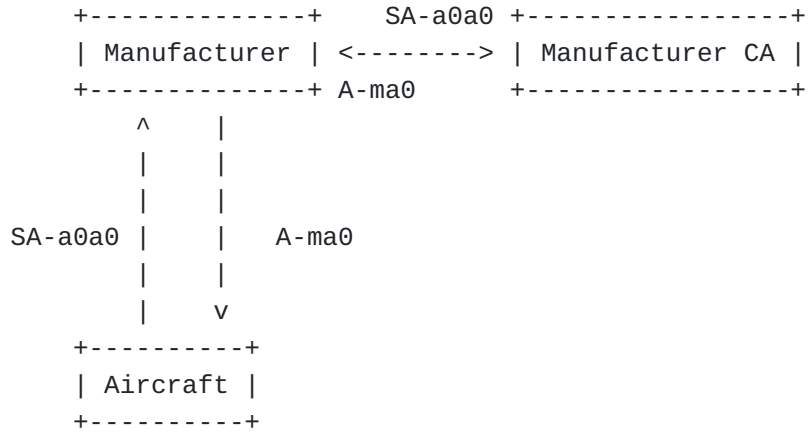


Figure 12: Manufacturer Provision

During the initial configuration and production at the factory the Aircraft MUST be configured to have a serial number. ASTM defines this to be an ANSI/CTA-2063A. Under DRIP a HHIT can be encoded as such to be able to convert back and forth between them. This is out of scope for this document. TODO: link from UAS RID document.

Under DRIP the Manufacturer SHOULD be using HHITs and have their own keypair and SA-mm (SelfAttestation: Manufacturer on Manufacturer). (Ed. Note: some words on aircraft keypair and certs here?).

SelfAttestation: Aircraft 0 on Aircraft 0 (SA-a0a0) is extracted by the manufacturer and sent to their Certificate Authority (CA) to be verified and added. A resulting attestation (Attestation: Manufacturer on Aircraft 0 [A-ma0]) SHOULD be a DRIP Attestation - however this could be a X.509 certificate binding the serial number to the manufacturer.

8.5. Operator

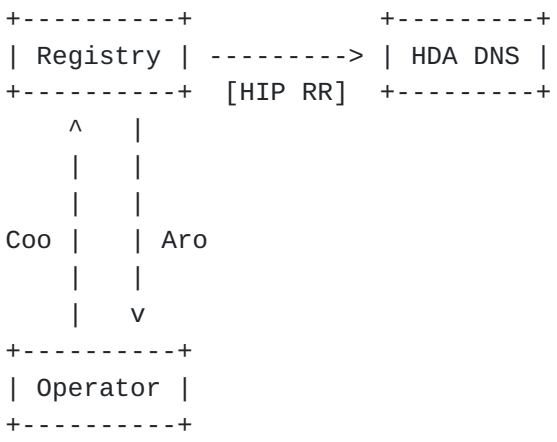


Figure 13: Operator Provision

The Operator generates a keypair and HHIT as specified in DRIP UAS RID. A self-signed attestation (Attestation: Operator on Operator [SA-oo]) is generated and sent to the desired Registry (HDA). Other relevant information and possibly personally identifiable information needed may also be required to be sent to the Registry (all over a secure channel - the method of which is out of scope for this document).

The Registry cross checks any personally identifiable information as required. Certificate: Operator on Operator is verified (both using the expiration timestamp and signature). The HHIT is searched in the Registries database to confirm that no collision occurs. A new attestation is generated (Attestation: Registry on Operator) and sent securely back to the Operator. Optionally the HHIT/HI pairing can be added to the Registries DNS in to form of a HIP Resource Record (RR). Other RRs, such as CERT and TXT, may also be used to hold public information.

With the receipt of Attestation: Registry on Operator (A-ro) the provisioning of an Operator is complete.

8.6. Aircraft

8.6.1. Standard Provisioning

Under standard provisioning the Aircraft has its own connectivity to the Registry, the method which is out of scope for this document.

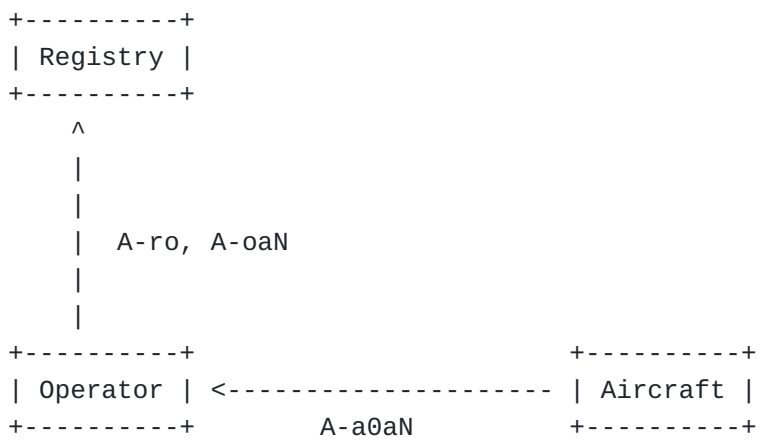


Figure 14: Standard Provision: Step 1

Through mechanisms not specified in this document the Aircraft should have methods to instruct the Aircraft onboard systems to generate a keypair and certificate. This certificate is chained to the factory provisioned certificate (SelfAttestation: Aircraft 0 on Aircraft 0 [SA-a0a0]). This new attestation (Attestation: Aircraft 0 on Aircraft N [A-a0aN]) is securely extracted by the Operator.

With A-a0aN the sub-attestation (SelfAttestation: Aircraft N on Aircraft N [SA-aNaN]) is used by the Operator to generate Attestation: Operator on Aircraft N (A-oaN). This along with Attestation: Registry on Operator (A-ro) is sent to the Registry.

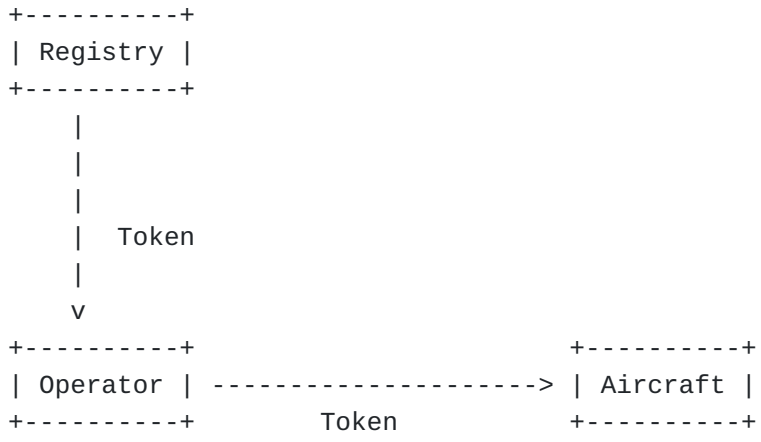


Figure 15: Standard Provision: Step 2

On the Registry, A-ro is verified and used as confirmation that the Operator is already registered. A-oaN also undergoes a validation check and used to generate a token to return to the Operator to continue provisioning.

Upon receipt of this token, the Operator injects it into the Aircraft and its used to form a secure connection to the Registry.

The Aircraft then sends Attestation: Manufacturer on Aircraft 0 (A-ma0) and Attestation: Aircraft 0 to Aircraft N (A-a0aN).

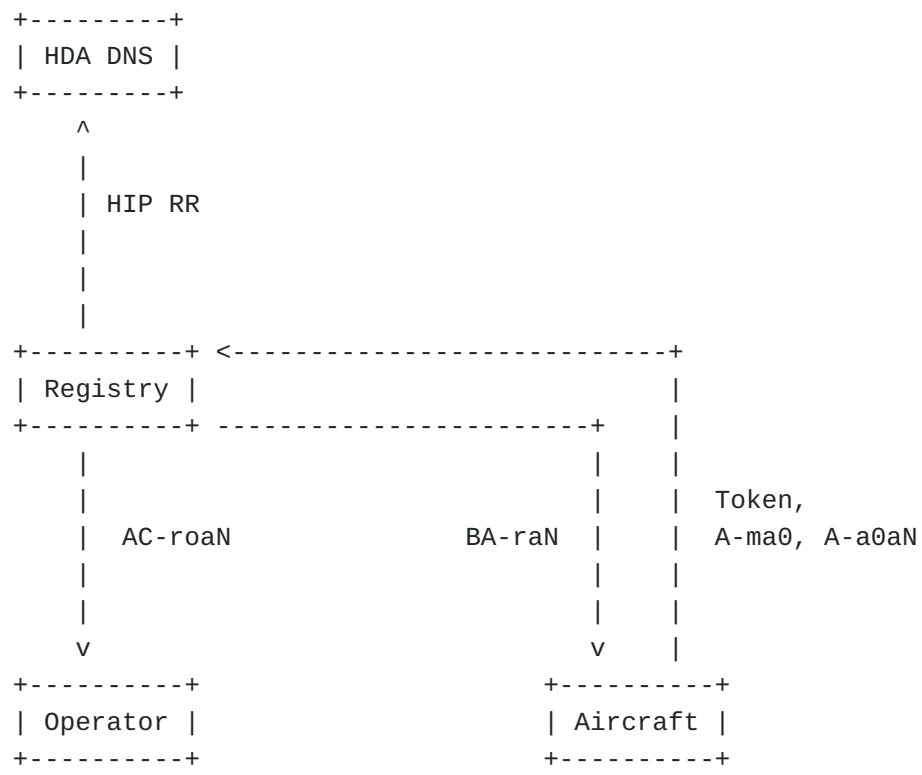


Figure 16: Standard Provision: Step 3

The Registry uses Attestation: Manufacturer on Aircraft 0 (with an external database if supported) to confirm the validity of the Aircraft. Attestation: Aircraft 0 on Aircraft N is correlated with Attestation: Operator on Aircraft N and Attestation: Manufacturer on Aircraft 0 to see the chain of ownership. The new HHIT tied to Aircraft N is then checked for collisions in the HDA. With the information the Registry generates two items: AttestationCertificate: Registry on Operator on Aircraft N (AC-roaN) and BroadcastAttestation: Registry on Aircraft N (BA-raN). A HIP RR (and other RR types as needed) are generated and inserted into the HDA.

AC-roaN is sent via a secure channel back to the Operator to be stored. ABA-raN is sent to the Aircraft to be used in Broadcast RID as specified in (Editors Note: add link to -auth-formats).

8.6.2. Operator Assisted Provisioning

This provisioning scheme is for when the Aircraft is unable to connect to the Registry itself or does not have the hardware required to generate keypairs and certificates.

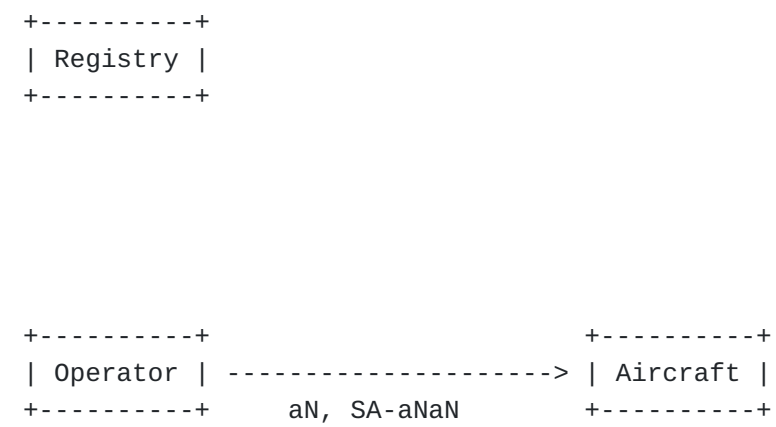


Figure 17: Operator Assisted Provision: Step 1

To start the Operator generates on behalf of the Aircraft a new keypair and Attestation: Aircraft N on Aircraft N (SA-aNaN). This keypair and certificate are injected into the Aircraft for it to generate Attestation: Aircraft 0 on Aircraft N (A-a0aN). After injecting the keypair and certificate, the Operator MUST destroy all copies of the keypair.

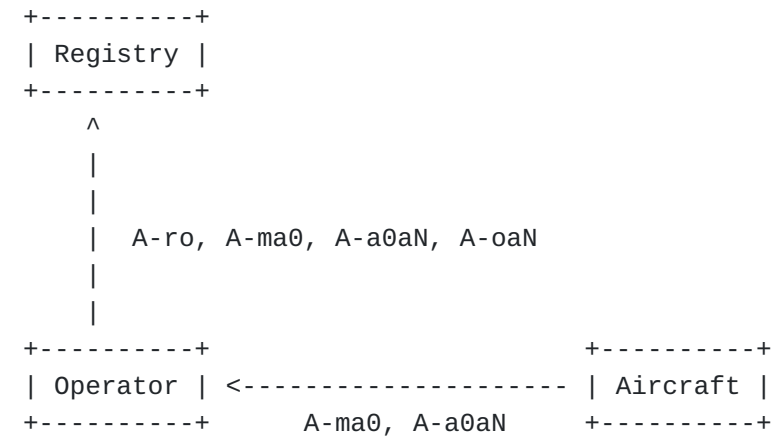


Figure 18: Operator Assisted Provision: Step 2

Attestation: Manufacturer on Aircraft 0 (A-ma0) and Attestation: Aircraft 0 on Aircraft N (A-a0aN) is extracted by the Operator and the following data items are sent to the Registry; Attestation: Registry on Operator (A-ro), Attestation: Manufacturer on Aircraft 0 (A-ma0), Attestation: Aircraft 0 on Aircraft N (A-a0aN), Attestation: Operator on Aircraft N (A-oaN).

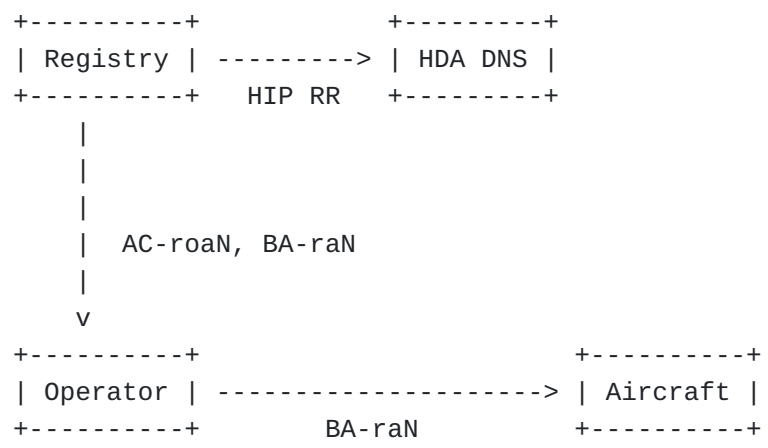


Figure 19: Operator Assisted Provision: Step 3

On the Registry validation checks are done on all attestations as per the previous sections. Once complete then the Registry checks for a HHIT collision, adding to the HDA if clear and generates AttestationCertificate: Registry on Operator on Aircraft N (AC-roaN) and BroadcastAttestation: Registry on Aircraft N (BA-raN). Both are sent back to the Operator.

The Operator securely inject BA-raN and securely stores AC-roaN of Aircraft N.

8.6.3. Initial Provisioning

A special form of provisioning is used when the Aircraft is first sold to an Operator. Instead of generating a new keypair, the built in keypair and certificate done by the Manufacturer is used to provision and register the aircraft to the owner.

For this either Standard or Operator Assisted methods can be used.

9. IANA Considerations

(Editor Note: EPP/RDAP adds to existing registries, CERT RR update, HIP RR update)

10. Security Considerations

TODO

11. Contributors

*Scott Hollenbeck for his guidance with EPP/RDAP

*Andrei Gurtov for his insights as a pilot

12. References

12.1. Normative References

[F3411-19] "Standard Specification for Remote ID and Tracking", February 2020.

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

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

12.2. Informative References

[drip-requirements] Card, S. W., Wiethuechter, A., Moskowitz, R., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Requirements", Work in Progress, Internet-Draft, draft-ietf-drip-reqs-18, 8 September 2021, <https://www.ietf.org/archive/id/draft-ietf-drip-reqs-18.txt>.

[drip-rid] Moskowitz, R., Card, S. W., Wiethuechter, A., and A. Gurtov, "UAS Remote ID", Work in Progress, Internet-Draft, draft-ietf-drip-uas-rid-01, 9 September 2020, <https://www.ietf.org/archive/id/draft-ietf-drip-uas-rid-01.txt>.

[hhit-registries] Moskowitz, R., Card, S. W., and A. Wiethuechter, "Hierarchical HIT Registries", Work in Progress, Internet-Draft, draft-moskowitz-hip-hhit-registries-02, 9 March 2020, <https://www.ietf.org/archive/id/draft-moskowitz-hip-hhit-registries-02.txt>.

[NPRM] "Notice of Proposed Rule Making on Remote Identification of Unmanned Aircraft Systems", December 2019.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

Authors' Addresses

Adam Wiethuechter
AX Enterprize, LLC
4947 Commercial Drive
Yorkville, NY 13495
United States of America

Email: adam.wiethuechter@axenterprize.com

Stuart Card
AX Enterprize, LLC
4947 Commercial Drive
Yorkville, NY 13495
United States of America

Email: stu.card@axenterprize.com

Robert Moskowitz
HTT Consulting
Oak Park, MI 48237
United States of America

Email: rgm@labs.htt-consult.com

Jim Reid
RTFM llp
St Andrews House
382 Hillington Road

Email: jim@rfc1035.com