

drip  
Internet-Draft  
Intended status: Informational  
Expires: 22 September 2022

S. Card  
A. Wiethuechter  
AX Enterprize  
R. Moskowitz  
HTT Consulting  
S. Zhao (Editor)  
Tencent  
A. Gurtov  
Linköping University  
21 March 2022

Drone Remote Identification Protocol (DRIP) Architecture  
draft-ietf-drip-arch-22

Abstract

This document describes an architecture for protocols and services to support Unmanned Aircraft System (UAS) Remote Identification (RID) and tracking, plus UAS RID-related communications. This architecture adheres to the requirements listed in the DRIP Requirements document (RFC9153).

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 22 September 2022.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Overview of Unmanned Aircraft System (UAS) Remote ID (RID) and Standardization . . . . .	3
1.2. Overview of Types of UAS Remote ID . . . . .	4
1.2.1. Broadcast RID . . . . .	4
1.2.2. Network RID . . . . .	5
1.3. Overview of USS Interoperability . . . . .	7
1.4. Overview of DRIP Architecture . . . . .	8
2. Terms and Definitions . . . . .	10
2.1. Additional Abbreviations . . . . .	10
2.2. Additional Definitions . . . . .	11
3. HHIT as the DRIP Entity Identifier . . . . .	11
3.1. UAS Remote Identifiers Problem Space . . . . .	12
3.2. HHIT as A Trustworthy DRIP Entity Identifier . . . . .	12
3.3. HHIT for DRIP Identifier Registration and Lookup . . . . .	14
3.4. HHIT as a Cryptographic Identifier . . . . .	14
4. DRIP Identifier Registration and Registries . . . . .	14
4.1. Public Information Registry . . . . .	15
4.1.1. Background . . . . .	15
4.1.2. DNS as the Public DRIP Identifier Registry . . . . .	15
4.2. Private Information Registry . . . . .	15
4.2.1. Background . . . . .	15
4.2.2. EPP and RDAP as the Private DRIP Identifier Registry . . . . .	16
4.2.3. Alternative Private DRIP Registry methods . . . . .	16
5. DRIP Identifier Trust . . . . .	16
6. Harvesting Broadcast Remote ID messages for UTM Inclusion . . . . .	17
6.1. The CS-RID Finder . . . . .	18
6.2. The CS-RID SDSP . . . . .	18
7. DRIP Contact . . . . .	18
8. IANA Considerations . . . . .	19
9. Security Considerations . . . . .	19
10. Privacy & Transparency Considerations . . . . .	20
11. References . . . . .	20
11.1. Normative References . . . . .	20
11.2. Informative References . . . . .	20
Appendix A. Overview of Unmanned Aircraft Systems (UAS) Traffic Management (UTM) . . . . .	24
A.1. Operation Concept . . . . .	24
A.2. UAS Service Supplier (USS) . . . . .	24

A.3. UTM Use Cases for UAS Operations . . . . .	25
Appendix B. Automatic Dependent Surveillance Broadcast (ADS-B) . . . . .	25
Acknowledgements . . . . .	26
Authors' Addresses . . . . .	26

## 1. Introduction

This document describes an architecture for protocols and services to support Unmanned Aircraft System (UAS) Remote Identification (RID) and tracking, plus RID-related communications. The architecture takes into account both current (including proposed) regulations and non-IETF technical standards.

The architecture adheres to the requirements listed in the DRIP Requirements document [RFC9153]. The requirements document provides an extended introduction to the problem space and use cases.

### 1.1. Overview of Unmanned Aircraft System (UAS) Remote ID (RID) and Standardization

UAS Remote Identification (RID) is an application that enables a UAS to be identified by Unmanned Aircraft Systems Traffic Management (UTM) and UAS Service Supplier (USS) (Appendix A) or third party entities such as law enforcement. Many considerations (e.g., safety) dictate that UAS be remotely identifiable.

Civil Aviation Authorities (CAAs) worldwide are mandating UAS RID. CAAs currently promulgate performance-based regulations that do not specify techniques, but rather cite industry consensus technical standards as acceptable means of compliance.

#### Federal Aviation Administration (FAA)

The FAA published a Notice of Proposed Rule Making [NPRM] in 2019 and thereafter published a "Final Rule" in 2021 [FAA\_RID], imposing requirements on UAS manufacturers and operators, both commercial and recreational. The rule clearly states that Automatic Dependent Surveillance Broadcast (ADS-B) Out and transponders cannot be used to satisfy the UAS RID requirements on UAS to which the rule applies (see Appendix B).

#### European Union Aviation Safety Agency (EASA)

The EASA published a [Delegated] regulation in 2019 imposing requirements on UAS manufacturers and third-country operators, including but not limited to UAS RID requirements. The EASA also published in 2019 an [Implementing] regulation laying down detailed rules and procedures for UAS operations and operating personnel.

#### American Society for Testing and Materials (ASTM)

ASTM International, Technical Committee F38 (UAS), Subcommittee F38.02 (Aircraft Operations), Work Item WK65041, developed the ASTM [F3411] Standard Specification for Remote ID and Tracking.

ASTM defines one set of UAS RID information and two means, MAC-layer broadcast and IP-layer network, of communicating it. If an UAS uses both communication methods, the same information must be provided via both means. [F3411] is cited by the FAA in its UAS RID final rule [FAA\_RID] as "a potential means of compliance" to a Remote ID rule.

#### The 3rd Generation Partnership Project (3GPP)

With release 16, the 3GPP completed the UAS RID requirement study [TS-22.825] and proposed a set of use cases in the mobile network and services that can be offered based on UAS RID. Release 17 specification focuses on enhanced UAS service requirements and provides the protocol and application architecture support that will be applicable for both 4G and 5G networks. The study of Further Architecture Enhancement for Uncrewed Aerial Vehicles (UAV) and Urban Air Mobility (UAM) [FS\_AEUA] in release 18 further enhances the communication mechanism between UAS and USS/UTM. The UAS RID discussed in Section 3 may be used as the 3GPP CAA-level UAS ID for Remote Identification purposes.

### 1.2. Overview of Types of UAS Remote ID

This specification introduces two types UAS Remote ID defined in ASTM [F3411].

#### 1.2.1. Broadcast RID

[F3411] defines a set of UAS RID messages for direct, one-way, broadcast transmissions from the UA over Bluetooth or Wi-Fi. These are currently defined as MAC-Layer messages. Internet (or other Wide Area Network) connectivity is only needed for UAS registry information lookup by Observers using the directly received UAS ID. Broadcast RID should be functionally usable in situations with no Internet connectivity.

The minimum Broadcast RID data flow is illustrated in Figure 1.

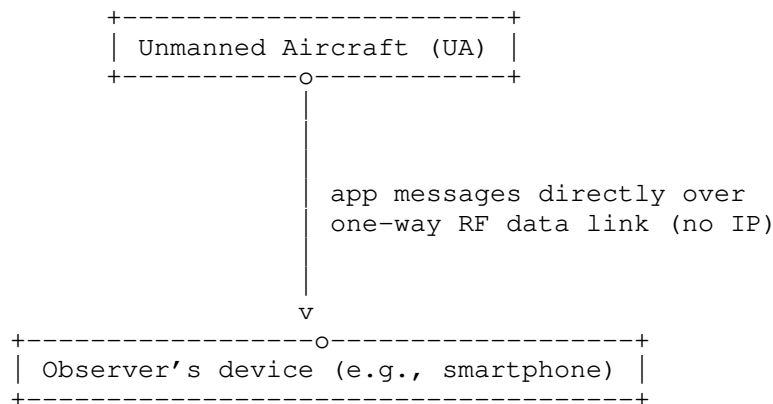


Figure 1

Broadcast RID provides information only about unmanned aircraft (UA) within direct Radio Frequency (RF) Line-Of-Sight (LOS), typically similar to Visual LOS (VLOS), with a range up to approximately 1 km. This information may be 'harvested' from received broadcasts and made available via the Internet, enabling surveillance of areas too large for local direct visual observation or direct RF link-based ID (see Section 6).

#### 1.2.2. Network RID

[F3411], using the same data dictionary that is the basis of Broadcast RID messages, defines a Network Remote Identification (Net-RID) data flow as follows.

- \* The information to be reported via UAS RID is generated by the UAS. Typically some of this data is generated by the UA and some by the GCS (Ground Control Station), e.g., their respective Global Navigation Satellite System (GNSS) derived locations.
- \* The information is sent by the UAS (UA or GCS) via unspecified means to the cognizant Network Remote Identification Service Provider (Net-RID SP), typically the USS under which the UAS is operating if participating in UTM.
- \* The Net-RID SP publishes via the Discovery and Synchronization Service (DSS) over the Internet that it has operations in various 4-D airspace volumes (Section 2.2 of [RFC9153]), describing the volumes but not the operations.

- \* An Observer's device, which is expected, but not specified, to be web-based, queries a Network Remote Identification Display Provider (Net-RID DP), typically also a USS, about any operations in a specific 4-D airspace volume.
- \* Using fully specified web-based methods over the Internet, the Net-RID DP queries all Net-RID SP that have operations in volumes intersecting that of the Observer's query for details on all such operations.
- \* The Net-RID DP aggregates information received from all such Net-RID SP and responds to the Observer's query.

The minimum Net-RID data flow is illustrated in Figure 2:

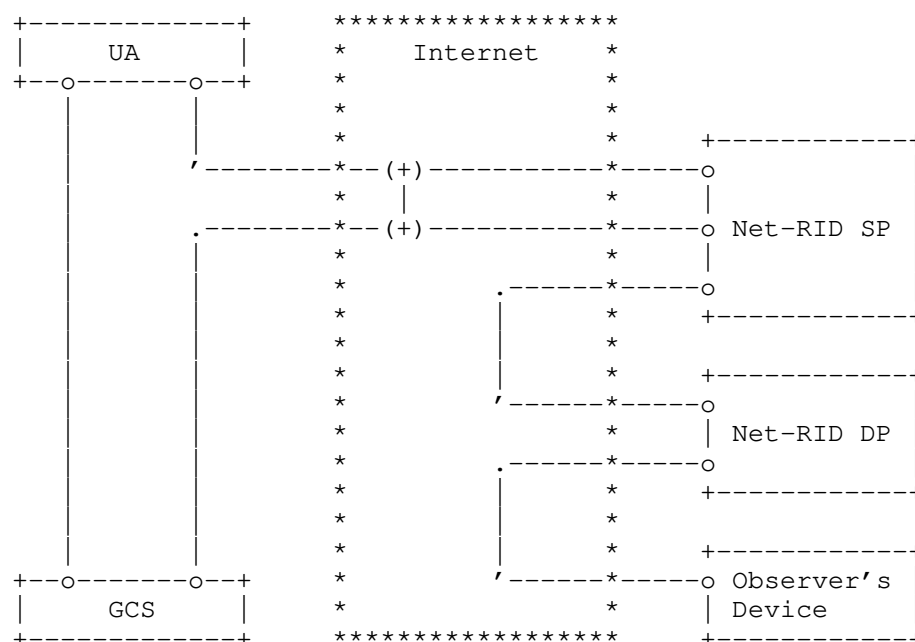


Figure 2

Command and Control (C2) must flow from the GCS to the UA via some path. Currently (in the year 2022) this is typically a direct RF link; however, with increasing Beyond Visual Line of Sight (BVLOS) operations, it is expected often to be a wireless link at either end with the Internet between.

Telemetry (at least UA's position and heading) flows from the UA to the GCS via some path, typically the reverse of the C2 path. Thus, UAS RID information pertaining to both the GCS and the UA can be sent, by whichever has Internet connectivity, to the Net-RID SP, typically the USS managing the UAS operation.

The Net-RID SP forwards UAS RID information via the Internet to subscribed Net-RID DPs, typically USS. Subscribed Net-RID DPs then forward RID information via the Internet to subscribed Observer devices. Regulations require and [F3411] describes UAS RID data elements that must be transported end-to-end from the UAS to the subscribed Observer devices.

[F3411] prescribes the protocols between the Net-RID SP, Net-RID DP, and the DSS. It also prescribes data elements (in JSON) between the Observer and the Net-RID DP. DRIP could address standardization of secure protocols between the UA and GCS (over direct wireless and Internet connection), between the UAS and the Net-RID SP, and/or between the Net-RID DP and Observer devices.

Informative note: Neither link layer protocols nor the use of links (e.g., the link often existing between the GCS and the UA) for any purpose other than carriage of UAS RID information is in the scope of [F3411] Network RID.

### 1.3. Overview of USS Interoperability

With Net-RID, there is direct communication between each UAS and its USS. Multiple USS exchange information with the assistance of a DSS so all USS collectively have knowledge about all activities in a 4D airspace. The interactions among an Observer, multiple UAS, and their USS are shown in Figure 3.

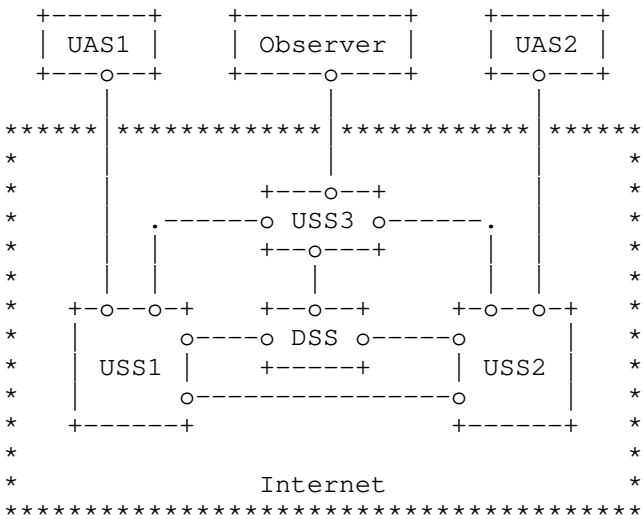
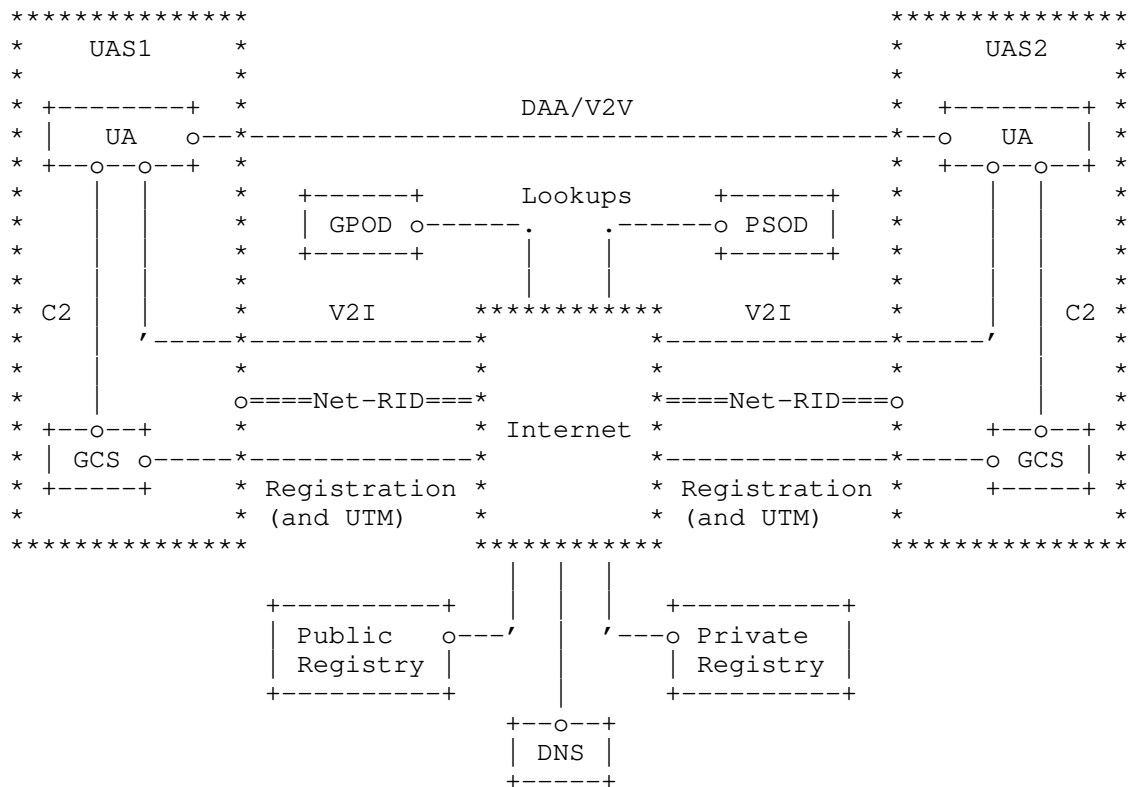


Figure 3

1.4. Overview of DRIP Architecture

Figure 4 illustrates a global UAS RID usage scenario. Broadcast RID links are not shown as they reach from any UA to any listening receiver in range and thus would obscure the intent of the figure. Figure 4 shows, as context, some entities and interfaces beyond the scope of DRIP (as currently (2022) chartered).





DAA: Detect And Avoid  
GPOD: General Public Observer Device  
PSOD: Public Safety Observer Device  
V2I: Vehicle-to-Infrastructure  
V2V: Vehicle-to-Vehicle

Figure 4

DRIP is meant to leverage existing Internet resources (standard protocols, services, infrastructures, and business models) to meet UAS RID and closely related needs. DRIP will specify how to apply IETF standards, complementing [F3411] and other external standards, to satisfy UAS RID requirements.

This document outlines the DRIP architecture in the context of the UAS RID architecture. This includes presenting the gaps between the CAAs' Concepts of Operations and [F3411] as it relates to the use of Internet technologies and UA direct RF communications. Issues include, but are not limited to:

- Design of trustworthy remote identifiers (Section 3).
- Mechanisms to leverage Domain Name System (DNS [RFC1034]), Extensible Provisioning Protocol (EPP [RFC5731]) and Registration Data Access Protocol (RDAP) ([RFC9082]) for publishing public and private information (see Section 4.1 and Section 4.2).
- Specific authentication methods and message payload formats to enable verification that Broadcast RID messages were sent by the claimed sender (Section 5) and that sender is in the claimed registry (Section 4 and Section 5).
- Harvesting Broadcast RID messages for UTM inclusion, with the optional DRIP extension of Crowd Sourced Remote ID (CS-RID, Section 6), using the DRIP support for gateways required by GEN-5 [RFC9153].
- Methods for instantly establishing secure communications between an Observer and the pilot of an observed UAS (Section 7), using the DRIP support for dynamic contact required by GEN-4 [RFC9153].
- Privacy in UAS RID messages (PII protection) (Section 10).

## 2. Terms and Definitions

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.

To encourage comprehension necessary for adoption of DRIP by the intended user community, the UAS community's norms are respected herein.

This document uses terms defined in [RFC9153].

### 2.1. Additional Abbreviations

DET:	DRIP Entity Tag
EdDSA:	Edwards-Curve Digital Signature Algorithm
HHIT:	Hierarchical HIT
HI:	Host Identity

HIP: Host Identity Protocol

HIT: Host Identity Tag

## 2.2. Additional Definitions

This section introduces the terms "Claims", "Assertions", "Attestations", and "Certificates" as used in DRIP. DRIP certificate has a different context compared with security certificates and Public Key Infrastructure used in X.509.

### Claims:

A claim in DRIP is a predicate (e.g., "X is Y", "X has property Y", and most importantly "X owns Y" or "X is owned by Y").

### Assertions:

An assertion in DRIP is a set of claims. This definition is borrowed from JWT [RFC7519] and CWT [RFC8392].

### Attestations:

An attestation in DRIP is a signed assertion. The signer may be the claimant or a related party with stake in the assertion(s). Under DRIP this is normally used when an entity asserts a relationship with another entity, along with other information, and the asserting entity signs the assertion, thereby making it an attestation.

### Certificates:

A certificate in DRIP is an attestation, strictly over identity information, signed by a third party. This third party should be one with no stake in the attestation(s) over which it is signing.

## 3. HHIT as the DRIP Entity Identifier

This section describes the DRIP architectural approach to meeting the basic requirements of a DRIP entity identifier within external technical standard ASTM [F3411] and regulatory constraints. It justifies and explains the use of Hierarchical Host Identity Tags (HHITs) [RFC7401] as self-asserting IPv6 addresses suitable as a UAS ID type and, more generally, as trustworthy multipurpose remote identifiers.

Self-asserting in this usage means that, given the Host Identity (HI), the HHIT ORCHID construction and a signature of the registry on the HHIT, the HHIT can be verified by the receiver. The explicit registration hierarchy within the HHIT provides registry discovery (managed by a Registrar) to either yield the HI for a 3rd-party (seeking UAS ID attestation) validation or prove that the HHIT and HI have been registered uniquely.

### 3.1. UAS Remote Identifiers Problem Space

A DRIP entity identifier needs to be "Trustworthy" (See DRIP Requirement GEN-1, ID-4 and ID-5 in [RFC9153]). This means that given a sufficient collection of UAS RID messages, an Observer can establish that the identifier claimed therein uniquely belongs to the claimant. To satisfy DRIP requirements and maintain important security properties, the DRIP identifier should be self-generated by the entity it names (e.g., a UAS) and registered (e.g., with a USS, see Requirements GEN-3 and ID-2).

Broadcast RID, especially its support for Bluetooth 4, imposes severe constraints. ASTM UAS RID [F3411] allows a UAS ID of types 1, 2 and 3 of 20 bytes; a revision to [F3411], currently in balloting (as of Oct 2021), adds type 4, Specific Session ID, to be standardized by IETF and other standards development organizations (SDOs) as extensions to ASTM UAS RID, consumes one of those bytes to index the sub-type, leaving only 19 for the identifier (see DRIP Requirement ID-1).

Likewise, the maximum ASTM UAS RID [F3411] Authentication Message payload is 201 bytes for most authentication types. A type 5 is also added in this revision for IETF and other SDOs to develop Specific Authentication Methods as extensions to ASTM UAS RID. One byte out of 201 bytes is consumed to index the sub-type which leaves only 200 for DRIP authentication payloads, including one or more DRIP entity identifiers and associated authentication data.

### 3.2. HHIT as A Trustworthy DRIP Entity Identifier

A Remote UAS ID that can be trustworthy for use in Broadcast RID can be built from an asymmetric keypair. In this method, the UAS ID is cryptographically derived directly from the public key. The proof of UAS ID ownership (verifiable attestation, versus mere claim) is guaranteed by signing this cryptographic UAS ID with the associated private key. The association between the UAS ID and the private key is ensured by cryptographically binding the public key with the UAS ID; more specifically, the UAS ID results from the hash of the public key. The public key is designated as the HI while the UAS ID is designated as the HIT.

By construction, the HIT is statistically unique through the cryptographic hash feature of second-preimage resistance. The cryptographically-bound addition of the Hierarchy and an HHIT registration process provide complete, global HHIT uniqueness. This registration forces the attacker to generate the same public key rather than a public key that generates the same HHIT. This is in contrast to general IDs (e.g., a UUID or device serial number) as the subject in an X.509 certificate.

A UA equipped for Broadcast RID SHOULD be provisioned not only with its HHIT but also with the HI public key from which the HHIT was derived and the corresponding private key, to enable message signature. A UAS equipped for Network RID SHOULD be provisioned likewise; the private key resides only in the ultimate source of Network RID messages (i.e., on the UA itself if the GCS is merely relaying rather than sourcing Network RID messages). Each Observer device SHOULD be provisioned either with public keys of the DRIP identifier root registries or certificates for subordinate registries.

HHITs can also be used throughout the USS/UTM system. Operators and Private Information Registries, as well as other UTM entities, can use HHITs for their IDs. Such HHITs can facilitate DRIP security functions such as used with HIP to strongly mutually authenticate and encrypt communications.

A self-attestation of a HHIT used as a UAS ID can be done in as little as 84 bytes when Ed25519 [RFC8032] is used, by avoiding an explicit encoding technology like ASN.1 or Concise Binary Object Representation (CBOR [RFC8949]). This attestation consists of only the HHIT, a timestamp, and the EdDSA signature on them.

A DRIP identifier can be assigned to a UAS as a static HHIT by its manufacturer, such as a single HI and derived HHIT encoded as a hardware serial number per [CTA2063A]. Such a static HHIT SHOULD only be used to bind one-time use DRIP identifiers to the unique UA. Depending upon implementation, this may leave a HI private key in the possession of the manufacturer (more details in Section 9).

In general, Internet access may be needed to validate Attestations or Certificates. This may be obviated in the most common cases (e.g., attestation of the UAS ID), even in disconnected environments, by prepopulating small caches on Observer devices with Registry public keys and a chain of Attestations or Certificates (tracing a path through the Registry tree). This is assuming all parties on the trust path also use HHITs for their identities.

### 3.3. HHIT for DRIP Identifier Registration and Lookup

UAS RID needs a deterministic lookup mechanism that rapidly provides actionable information about the identified UA. Given the size constraints imposed by the Bluetooth 4 broadcast media, the UAS ID itself needs to be a non-spoofable inquiry input into the lookup.

A DRIP registration process based on the explicit hierarchy within a HHIT provides manageable uniqueness of the HI for the HHIT. This is the defense against a cryptographic hash second pre-image attack on the HHIT (e.g., multiple HIs yielding the same HHIT, see Requirement ID-3). A lookup of the HHIT into this registration data provides the registered HI for HHIT proof of ownership. A first-come-first-served registration for a HHIT provides deterministic access to any other needed actionable information based on inquiry access authority (more details in Section 4.2).

### 3.4. HHIT as a Cryptographic Identifier

The only (known to the authors at the time of this writing) existing types of IP address compatible identifiers cryptographically derived from the public keys of the identified entities are Cryptographically Generated Addresses (CGAs) [RFC3972] and Host Identity Tags (HITs) [RFC7401]. CGAs and HITs lack registration/retrieval capability. To provide this, each HHIT embeds plaintext information designating the hierarchy within which it is registered and a cryptographic hash of that information concatenated with the entity's public key, etc. Although hash collisions may occur, the registrar can detect them and reject registration requests rather than issue credentials, e.g., by enforcing a first-claimed, first-attested policy. Pre-image hash attacks are also mitigated through this registration process, locking the HHIT to a specific HI

## 4. DRIP Identifier Registration and Registries

DRIP registries hold both public and private UAS information (See PRIV-1 in [RFC9153]) resulting from the DRIP identifier registration process. Given these different uses, and to improve scalability, security, and simplicity of administration, the public and private information can be stored in different registries. This section introduces the public and private information registries for DRIP identifiers. This DRIP Identifier registration process satisfies the following DRIP requirements defined in [RFC9153]: GEN-3, GEN-4, ID-2, ID-4, ID-6, PRIV-3, PRIV-4, REG-1, REG-2, REG-3 and REG-4.

#### 4.1. Public Information Registry

##### 4.1.1. Background

The public information registry provides trustable information such as attestations of UAS RID ownership and registration with the HDA (Hierarchical HIT Domain Authority). Optionally, pointers to the registries for the HDA and RAA (Registered Assigning Authority) implicit in the UAS RID can be included (e.g., for HDA and RAA HHIT|HI used in attestation signing operations). This public information will be principally used by Observers of Broadcast RID messages. Data on UAS that only use Network RID, is available via an Observer's Net-RID DP that would directly provide all public information registry information. The Net-RID DP is the only source of information for a query on an airspace volume.

##### 4.1.2. DNS as the Public DRIP Identifier Registry

A DRIP identifier SHOULD be registered as an Internet domain name (at an arbitrary level in the hierarchy, e.g., in .ip6.arpa). Thus DNS can provide all the needed public DRIP information. A standardized HHIT FQDN (Fully Qualified Domain Name) can deliver the HI via a HIP RR (Resource Record) [RFC8005] and other public information (e.g., RRA and HDA PTRs, and HIP RVS (Rendezvous Servers) [RFC8004]). These public information registries can use secure DNS transport (e.g., DNS over TLS) to deliver public information that is not inherently trustable (e.g., everything other than attestations).

#### 4.2. Private Information Registry

##### 4.2.1. Background

The private information required for DRIP identifiers is similar to that required for Internet domain name registration. A DRIP identifier solution can leverage existing Internet resources: registration protocols, infrastructure, and business models, by fitting into an UAS ID structure compatible with DNS names. The HHIT hierarchy can provide the needed scalability and management structure. It is expected that the private information registry function will be provided by the same organizations that run a USS, and likely integrated with a USS. The lookup function may be implemented by the Net-RID DPs.

#### 4.2.2. EPP and RDAP as the Private DRIP Identifier Registry

A DRIP private information registry supports essential registry operations (e.g., add, delete, update, query) using interoperable open standard protocols. It can accomplish this by using the Extensible Provisioning Protocol (EPP [RFC5730]) and the Registry Data Access Protocol (RDAP [RFC7480] [RFC9082] [RFC9083]). The DRIP private information registry in which a given UAS is registered needs to be findable, starting from the UAS ID, using the methods specified in [RFC7484].

#### 4.2.3. Alternative Private DRIP Registry methods

A DRIP private information registry might be an access-controlled DNS (e.g., via DNS over TLS). Additionally, WebFinger [RFC7033] can be deployed. These alternative methods may be used by Net-RID DP with specific customers.

### 5. DRIP Identifier Trust

While the DRIP entity identifier is self-asserting, it alone does not provide the trustworthiness (non-repudiability, protection vs. spoofing, message integrity protection, scalability, etc.) essential to UAS RID, as justified in [RFC9153]. For that it MUST be registered (under DRIP Registries) and be actively used by the party (in most cases the UA). A sender's identity can not be approved by only possessing a DRIP Entity Tag (DET), which is an HHIT-based UA ID and broadcasting a claim that it belongs to that sender. Even the sender using that HI's private key to sign static data proves nothing as well, as it is subject to trivial replay attacks. Only sending the DET and a signature on frequently changing data that can be sanity-checked by the Observer (such as a Location/Vector message) proves that the observed UA possesses the claimed UAS ID.

For Broadcast RID, it is a challenge to balance the original requirements of Broadcast RID and the efforts needed to satisfy the DRIP requirements all under severe constraints. From received Broadcast RID messages and information that can be looked up using the received UAS ID in online registries or local caches, it is possible to establish levels of trust in the asserted information and the Operator.

Optimization of different DRIP Authentication Messages allows an Observer, without Internet connection (offline) or with (online), to be able to validate a UAS DRIP ID in real-time. First is the sending of Broadcast Attestations (over DRIP Link Authentication Messages) [I-D.ietf-drip-auth] containing the relevant registration of the UA's DRIP ID in the claimed Registry. Next is sending DRIP Wrapper



Authentication Messages that sign over both static (e.g., above registration) and dynamically changing data (such as UA location data). Combining these two sets of information, an Observer can piece together a chain of trust and real-time evidence to make their determination of the UA's claims.

This process (combining the DRIP entity identifier, Registries and Authentication Formats for Broadcast RID) can satisfy the following DRIP requirement defined in [RFC9153]: GEN-1, GEN-2, GEN-3, ID-2, ID-3, ID-4 and ID-5.

#### 6. Harvesting Broadcast Remote ID messages for UTM Inclusion

ASTM anticipated that regulators would require both Broadcast RID and Network RID for large UAS, but allow UAS RID requirements for small UAS to be satisfied with the operator's choice of either Broadcast RID or Network RID. The EASA initially specified Broadcast RID for essentially all UAS, and is now also considering Network RID. The FAA UAS RID Final Rules [FAA\_RID] permit only Broadcast RID for rule compliance, but still encourage Network RID for complementary functionality, especially in support of UTM.

One opportunity is to enhance the architecture with gateways from Broadcast RID to Network RID. This provides the best of both and gives regulators and operators flexibility. It offers advantages over either form of UAS RID alone: greater fidelity than Network RID reporting of planned area operations; surveillance of areas too large for local direct visual observation and direct RF-LOS link based Broadcast RID (e.g., a city or a national forest).

These gateways could be pre-positioned (e.g., around airports, public gatherings, and other sensitive areas) and/or crowd-sourced (as nothing more than a smartphone with a suitable app is needed). As Broadcast RID media have limited range, gateways receiving messages claiming locations far from the gateway can alert authorities or a SDSP to the failed sanity check possibly indicating intent to deceive. Surveillance SDSPs can use messages with precise date/time/position stamps from the gateways to multilaterate UA location, independent of the locations claimed in the messages, which are entirely operator self-reported in UAS RID and UTM, and thus are subject not only to natural time lag and error but also operator misconfiguration or intentional deception.

Multilateration technologies use physical layer information, such as precise Time Of Arrival (TOA) of transmissions from mobile transmitters at receivers with a priori precisely known locations, to estimate the locations of the mobile transmitters.

Further, gateways with additional sensors (e.g., smartphones with cameras) can provide independent information on the UA type and size, confirming or refuting those claims made in the UAS RID messages.

Section 6.1 and Section 6.2 define two additional entities that are required to provide this Crowd Sourced Remote ID (CS-RID).

This approach satisfies the following DRIP requirements defined in [RFC9153]: GEN-5, GEN-11, and REG-1.

#### 6.1. The CS-RID Finder

A CS-RID Finder is the gateway for Broadcast Remote ID Messages into UTM. It performs this gateway function via a CS-RID SDSP. A CS-RID Finder could implement, integrate, or accept outputs from a Broadcast RID receiver. However, it should not depend upon a direct interface with a GCS, Net-RID SP, Net-RID DP or Network RID client. It would present a new interface to a CS-RID SDSP, similar to but readily distinguishable from that between a GCS and a Net-RID SP.

#### 6.2. The CS-RID SDSP

A CS-RID SDSP aggregates and processes (e.g., estimates UA location using multilateration when possible) information collected by CS-RID Finders. A CS-RID SDSP should appear (i.e., present the same interface) to a Net-RID SP as a Net-RID DP.

### 7. DRIP Contact

One of the ways in which DRIP can enhance [F3411] with immediately actionable information is by enabling an Observer to instantly initiate secure communications with the UAS remote pilot, Pilot In Command, operator, USS under which the operation is being flown, or other entity potentially able to furnish further information regarding the operation and its intent and/or to immediately influence further conduct or termination of the operation (e.g., land or otherwise exit an airspace volume). Such potentially distracting communications demand strong "AAA" (Authentication, Attestation, Authorization, Access Control, Accounting, Attribution, Audit) per applicable policies (e.g., of the cognizant CAA).

A DRIP entity identifier based on a HHIT as outlined in Section 3 embeds an identifier of the registry in which it can be found (expected typically to be the USS under which the UAS is flying) and the procedures outlined in Section 5 enable Observer verification of that relationship. A DRIP entity identifier with suitable records in public and private registries as outlined in Section 5 can enable lookup not only of information regarding the UAS, but also identities

of and pointers to information regarding the various associated entities (e.g., the USS under which the UAS is flying an operation), including means of contacting those associated entities (i.e., locators, typically IP addresses).

A suitably equipped Observer could initiate a cryptographic handshake to a similarly equipped and identified entity: the UA itself, if operating autonomously; the GCS, if the UA is remotely piloted and the necessary records have been populated in DNS; the USS, etc. Assuming mutual authentication is successful, keys can then be negotiated for an IPsec Encapsulating Security Payload (ESP) tunnel, over which arbitrary standard higher layer protocols can then be used for Observer to Pilot (O2P) communications (e.g., SIP [RFC3261] et seq), V2X communications (e.g., [MAVLink]), etc. Certain preconditions are necessary: each party needs a currently usable means (typically DNS) of resolving the other party's DRIP entity identifier to a currently usable locator (IP address); and there must be currently usable bidirectional IP (not necessarily Internet) connectivity between the parties. One method directly supported by the use of HHITs as DRIP entity identifiers is initiation of a HIP Base Exchange (BEX) and Bound End-to-End Tunnel (BEET).

This approach satisfies DRIP requirement GEN-6 Contact, supports satisfaction of requirements [RFC9153] GEN-8, GEN-9, PRIV-2, PRIV-5 and REG-3, and is compatible with all other DRIP requirements.

## 8. IANA Considerations

This document does not make any IANA request.

## 9. Security Considerations

The security provided by asymmetric cryptographic techniques depends upon protection of the private keys. It may be necessary for the GCS to have the key pair to register the HHIT to the USS. Thus it may be the GCS that generates the key pair and delivers it to the UA, making the GCS a part of the key security boundary. Leakage of the private key either from the UA or GCS to the component manufacturer is a valid concern and steps need to be in place to ensure safe keeping of the private key.

The size of the public key hash in the HHIT is also of concern. It is well within current server array technology to compute another key pair that hashes to the same HHIT. Thus an adversary could impersonate a validly registered UA. This attack would only be exposed when the HI in DRIP authentication message is checked back to the USS and found not to match.

Finally, the UAS RID sender of a small harmless UA (or the entire UA) could be carried by a larger dangerous UA as a "false flag." Compromise of a registry private key could do widespread harm. Key revocation procedures are as yet to be determined. These risks are in addition to those involving Operator key management practices.

## 10. Privacy & Transparency Considerations

Broadcast RID messages can contain Personally Identifiable Information (PII). A viable architecture for PII protection would be symmetric encryption of the PII using a session key known to the UAS and its USS. Authorized Observers could obtain plaintext in either of two ways. An Observer can send the UAS ID and the cyphertext to a server that offers decryption as a service. An Observer can send the UAS ID only to a server that returns the session key, so that Observer can directly locally decrypt all cyphertext sent by that UA during that session (UAS operation). In either case, the server can be: a Public Safety USS, the Observer's own USS, or the UA's USS if the latter can be determined (which under DRIP it can be, from the UAS ID itself). PII can be protected unless the UAS is informed otherwise. This could come as part of UTM operation authorization. It can be special instructions at the start or during an operation. PII protection MUST NOT be used if the UAS loses connectivity to the USS. The UAS always has the option to abort the operation if PII protection is disallowed.

## 11. References

### 11.1. Normative References

- [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>>.
- [RFC9153] Card, S., Ed., Wiethuechter, A., Moskowitz, R., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Requirements and Terminology", RFC 9153, DOI 10.17487/RFC9153, February 2022, <<https://www.rfc-editor.org/info/rfc9153>>.

### 11.2. Informative References

- [CTA2063A] ANSI, "Small Unmanned Aerial Systems Serial Numbers", 2019.
- [Delegated] European Union Aviation Safety Agency (EASA), "EU Commission Delegated Regulation 2019/945 of 12 March 2019 on unmanned aircraft systems and on third-country operators of unmanned aircraft systems", 2019.
- [F3411] ASTM International, "Standard Specification for Remote ID and Tracking", February 2020, <<http://www.astm.org/cgi-bin/resolver.cgi?F3411>>.
- [FAA\_RID] United States Federal Aviation Administration (FAA), "Remote Identification of Unmanned Aircraft", 2021, <<https://www.govinfo.gov/content/pkg/FR-2021-01-15/pdf/2020-28948.pdf>>.
- [FAA\_UAS\_Concept\_Of\_Ops] United States Federal Aviation Administration (FAA), "Unmanned Aircraft System (UAS) Traffic Management (UTM) Concept of Operations (V2.0)", 2020, <[https://www.faa.gov/uas/research\\_development/traffic\\_management/media/UTM\\_ConOps\\_v2.pdf](https://www.faa.gov/uas/research_development/traffic_management/media/UTM_ConOps_v2.pdf)>.
- [FS\_AEUA] "Study of Further Architecture Enhancement for UAV and UAM", 2021, <[https://www.3gpp.org/ftp/tsg\\_sa/WG2\\_Arch/TSGS2\\_147E\\_Electronic\\_2021-10/Docs/S2-2107092.zip](https://www.3gpp.org/ftp/tsg_sa/WG2_Arch/TSGS2_147E_Electronic_2021-10/Docs/S2-2107092.zip)>.
- [I-D.ietf-drip-auth] Wiethuechter, A., Card, S., and R. Moskowitz, "DRIP Authentication Formats & Protocols for Broadcast Remote ID", Work in Progress, Internet-Draft, draft-ietf-drip-auth-05, 7 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-drip-auth-05.txt>>.
- [Implementing] European Union Aviation Safety Agency (EASA), "EU Commission Implementing Regulation 2019/947 of 24 May 2019 on the rules and procedures for the operation of unmanned aircraft", 2019.
- [LAANC] United States Federal Aviation Administration (FAA), "Low Altitude Authorization and Notification Capability", n.d., <[https://www.faa.gov/uas/programs\\_partnerships/data\\_exchange/](https://www.faa.gov/uas/programs_partnerships/data_exchange/)>.

- [MAVLink] "Micro Air Vehicle Communication Protocol", 2021, <<http://mavlink.io/>>.
- [NPRM] United States Federal Aviation Administration (FAA), "Notice of Proposed Rule Making on Remote Identification of Unmanned Aircraft Systems", 2019.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<https://www.rfc-editor.org/info/rfc3972>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.
- [RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <<https://www.rfc-editor.org/info/rfc5731>>.
- [RFC7033] Jones, P., Salgueiro, G., Jones, M., and J. Smarr, "WebFinger", RFC 7033, DOI 10.17487/RFC7033, September 2013, <<https://www.rfc-editor.org/info/rfc7033>>.
- [RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <<https://www.rfc-editor.org/info/rfc7401>>.
- [RFC7480] Newton, A., Ellacott, B., and N. Kong, "HTTP Usage in the Registration Data Access Protocol (RDAP)", STD 95, RFC 7480, DOI 10.17487/RFC7480, March 2015, <<https://www.rfc-editor.org/info/rfc7480>>.
- [RFC7484] Blanchet, M., "Finding the Authoritative Registration Data (RDAP) Service", RFC 7484, DOI 10.17487/RFC7484, March 2015, <<https://www.rfc-editor.org/info/rfc7484>>.

- [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>>.
- [RFC8004] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", RFC 8004, DOI 10.17487/RFC8004, October 2016, <<https://www.rfc-editor.org/info/rfc8004>>.
- [RFC8005] Laganier, J., "Host Identity Protocol (HIP) Domain Name System (DNS) Extension", RFC 8005, DOI 10.17487/RFC8005, October 2016, <<https://www.rfc-editor.org/info/rfc8005>>.
- [RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <<https://www.rfc-editor.org/info/rfc8032>>.
- [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>>.
- [RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <<https://www.rfc-editor.org/info/rfc8949>>.
- [RFC9082] Hollenbeck, S. and A. Newton, "Registration Data Access Protocol (RDAP) Query Format", STD 95, RFC 9082, DOI 10.17487/RFC9082, June 2021, <<https://www.rfc-editor.org/info/rfc9082>>.
- [RFC9083] Hollenbeck, S. and A. Newton, "JSON Responses for the Registration Data Access Protocol (RDAP)", STD 95, RFC 9083, DOI 10.17487/RFC9083, June 2021, <<https://www.rfc-editor.org/info/rfc9083>>.
- [TS-22.825] 3GPP, "Study on Remote Identification of Unmanned Aerial Systems (UAS)", n.d., <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3527>>.
- [U-Space] European Organization for the Safety of Air Navigation (EUROCONTROL), "U-space Concept of Operations", 2019, <<https://www.sesarju.eu/sites/default/files/documents/u-space/CORUS%20ConOps%20vol2.pdf>>.

## Appendix A. Overview of Unmanned Aircraft Systems (UAS) Traffic Management (UTM)

### A.1. Operation Concept

The National Aeronautics and Space Administration (NASA) and FAA's effort to integrate UAS operations into the national airspace system (NAS) led to the development of the concept of UTM and the ecosystem around it. The UTM concept was initially presented in 2013 and version 2.0 was published in 2020 [FAA\_UAS\_Concept\_Of\_Ops].

The eventual concept refinement, initial prototype implementation, and testing were conducted by the joint FAA and NASA UTM research transition team. World efforts took place afterward. The Single European Sky ATM Research (SESAR) started the CORUS project to research its UTM counterpart concept, namely [U-Space]. This effort is led by the European Organization for the Safety of Air Navigation (Eurocontrol).

Both NASA and SESAR have published their UTM concepts of operations to guide the development of their future air traffic management (ATM) system and ensure safe and efficient integration of manned and unmanned aircraft into the national airspace.

UTM comprises UAS operations infrastructure, procedures and local regulation compliance policies to guarantee safe UAS integration and operation. The main functionality of UTM includes, but is not limited to, providing means of communication between UAS operators and service providers and a platform to facilitate communication among UAS service providers.

### A.2. UAS Service Supplier (USS)

A USS plays an important role to fulfill the key performance indicators (KPIs) that UTM has to offer. Such an Entity acts as a proxy between UAS operators and UTM service providers. It provides services like real-time UAS traffic monitoring and planning, aeronautical data archiving, airspace and violation control, interacting with other third-party control entities, etc. A USS can coexist with other USS to build a large service coverage map that can load-balance, relay, and share UAS traffic information.

The FAA works with UAS industry shareholders and promotes the Low Altitude Authorization and Notification Capability [LAANC] program, which is the first system to realize some of the envisioned functionality of UTM. The LAANC program can automate UAS operational intent (flight plan) submission and application for airspace authorization in real-time by checking against multiple aeronautical



databases such as airspace classification and operating rules associated with it, FAA UAS facility map, special use airspace, Notice to Airmen (NOTAM), and Temporary Flight Restriction (TFR).

#### A.3. UTM Use Cases for UAS Operations

This section illustrates a couple of use case scenarios where UAS participation in UTM has significant safety improvement.

1. For a UAS participating in UTM and taking off or landing in controlled airspace (e.g., Class Bravo, Charlie, Delta, and Echo in the United States), the USS under which the UAS is operating is responsible for verifying UA registration, authenticating the UAS operational intent (flight plan) by checking against designated UAS facility map database, obtaining the air traffic control (ATC) authorization, and monitoring the UAS flight path in order to maintain safe margins and follow the pre-authorized sequence of authorized 4-D volumes (route).
2. For a UAS participating in UTM and taking off or landing in uncontrolled airspace (e.g., Class Golf in the United States), pre-flight authorization must be obtained from a USS when operating beyond-visual-of-sight (BVLOS). The USS either accepts or rejects the received operational intent (flight plan) from the UAS. Accepted UAS operation may share its current flight data such as GPS position and altitude to USS. The USS may keep the UAS operation status near real-time and may keep it as a record for overall airspace air traffic monitoring.

#### Appendix B. Automatic Dependent Surveillance Broadcast (ADS-B)

The ADS-B is the de jure technology used in manned aviation for sharing location information, from the aircraft to ground and satellite-based systems, designed in the early 2000s. Broadcast RID is conceptually similar to ADS-B, but with the receiver target being the general public on generally available devices (e.g., smartphones).

For numerous technical reasons, ADS-B itself is not suitable for low-flying small UAS. Technical reasons include but not limited to the following:

1. Lack of support for the 1090 MHz ADS-B channel on any consumer handheld devices
2. Weight and cost of ADS-B transponders on CSWaP constrained UA

3. Limited bandwidth of both uplink and downlink, which would likely be saturated by large numbers of UAS, endangering manned aviation

Understanding these technical shortcomings, regulators worldwide have ruled out the use of ADS-B for the small UAS for which UAS RID and DRIP are intended.

#### Acknowledgements

The work of the FAA's UAS Identification and Tracking (UAS ID) Aviation Rulemaking Committee (ARC) is the foundation of later ASTM and proposed IETF DRIP WG efforts. The work of ASTM F38.02 in balancing the interests of diverse stakeholders is essential to the necessary rapid and widespread deployment of UAS RID. Thanks to Alexandre Petrescu and Stephan Wenger for the helpful and positive comments. Thanks to chairs Daniel Migault and Mohamed Boucadair for direction of our team of authors and editor, some of whom are newcomers to writing IETF documents. Laura Welch is also thanked for her valuable review comments that led to great improvements of this memo. Thanks especially to Internet Area Director Eric Vyncke for guidance and support.

#### Authors' Addresses

Stuart W. Card  
AX Enterprize  
4947 Commercial Drive  
Yorkville, NY, 13495  
United States of America  
Email: [stu.card@axenterprize.com](mailto:stu.card@axenterprize.com)

Adam Wiethuechter  
AX Enterprize  
4947 Commercial Drive  
Yorkville, NY, 13495  
United States of America  
Email: [adam.wiethuechter@axenterprize.com](mailto:adam.wiethuechter@axenterprize.com)

Robert Moskowitz  
HTT Consulting  
Oak Park, MI, 48237  
United States of America  
Email: [rgm@labs.htt-consult.com](mailto:rgm@labs.htt-consult.com)

Shuai Zhao  
Tencent  
2747 Park Blvd  
Palo Alto, 94588  
United States of America  
Email: shuai.zhao@ieee.org

Andrei Gurtov  
Linköping University  
IDA  
SE-58183 Linköping Linköping  
Sweden  
Email: gurtov@acm.org

DRIP Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 12 November 2022

A. Wiethuechter (Editor)  
S. Card  
AX Enterprize, LLC  
R. Moskowitz  
HTT Consulting  
11 May 2022

DRIP Entity Tag Authentication Formats & Protocols for Broadcast Remote  
ID  
draft-ietf-drip-auth-10

## Abstract

This document describes how to include trust into the ASTM Remote ID specification defined in ASTM F3411 under Broadcast Remote ID (RID). It defines a few message schemes (sent within the Authentication Message) that can be used to authenticate past messages sent by a unmanned aircraft (UA) and provide proof of UA trustworthiness even in the absence of Internet connectivity at the receiving node.

## 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 12 November 2022.

## Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. DRIP Requirements Addressed . . . . .	3
2. Terminology . . . . .	4
2.1. Required Terminology . . . . .	4
2.2. Definitions . . . . .	4
3. Background . . . . .	4
3.1. Problem Space and Focus . . . . .	4
3.1.1. Broadcast RID RF Options . . . . .	4
3.2. Reasoning for IETF DRIP Authentication . . . . .	5
3.3. ASTM Authentication Message . . . . .	5
3.3.1. Authentication Page . . . . .	5
3.3.2. ASTM Constraints . . . . .	8
4. Forward Error Correction . . . . .	8
4.1. Encoding . . . . .	8
4.1.1. Single Page FEC . . . . .	8
4.1.2. Multiple Page FEC . . . . .	9
4.2. Decoding . . . . .	11
4.2.1. Single Page FEC . . . . .	12
4.2.2. Multiple Page FEC . . . . .	12
4.3. FEC Limitations . . . . .	12
5. Broadcast Attestation Structure . . . . .	13
6. DRIP Authentication Formats . . . . .	14
6.1. DRIP Authentication Field Definitions . . . . .	15
6.2. Message Set Signature . . . . .	16
6.3. Specific Authentication Method . . . . .	17
6.3.1. SAM Data Format . . . . .	17
6.3.2. DRIP Link . . . . .	19
6.3.3. DRIP Wrapper . . . . .	19
6.3.4. DRIP Manifest . . . . .	21
6.3.5. DRIP Frame . . . . .	24
7. Requirements & Recommendations . . . . .	26
7.1. Legacy Transports . . . . .	26
7.2. Extended Transports . . . . .	26
7.3. Authentication . . . . .	26
7.4. Operational . . . . .	27
7.4.1. DRIP Wrapper . . . . .	28
8. ICAO Considerations . . . . .	28
9. IANA Considerations . . . . .	28
10. Security Considerations . . . . .	29
10.1. Manifest Hash Length . . . . .	29
10.2. Replay Attacks . . . . .	29
10.3. Trust Timestamp Offsets . . . . .	30

11. Acknowledgments . . . . .	30
12. References . . . . .	30
12.1. Normative References . . . . .	30
12.2. Informative References . . . . .	31
Appendix A. Authentication State Diagrams & Color Scheme . . . . .	31
A.1. State Table . . . . .	32
A.2. State Diagrams . . . . .	32
A.2.1. Notations . . . . .	33
A.2.2. General . . . . .	33
A.2.3. DRIP SAM . . . . .	34
A.2.4. DRIP Link . . . . .	35
A.2.5. DRIP Wrapper/Manifest/Frame . . . . .	36
Appendix B. HDA-UA Broadcast Attestation . . . . .	38
Appendix C. Example TX/RX Flow . . . . .	40
Authors' Addresses . . . . .	40

## 1. Introduction

Unmanned Aircraft Systems (UAS) are usually in a volatile environment when it comes to communication. UA are generally small with little computational (or flying) horsepower to carry standard communication equipment. This limits the mediums of communication to few viable options.

Observer systems (e.g. smartphones and tablets) place further constraints on the communication options. The Remote ID Broadcast messages MUST be available to applications on these platforms without modifying the devices.

The ASTM [F3411] standard focuses on two ways of communicating to a UAS for Remote ID (RID): Broadcast and Network.

This document will focus on adding trust to Broadcast RID via the Authentication Message by combining dynamically signed data with an Attestation of the UA's identity from a Registry.

This authentication methodology also provides the missing, but US FAA mandated, Error Correction for the Bluetooth 4 transmissions (see Section 4). This is error correction not only for the authentication message itself, but indirectly, to other messages authenticated via the Manifest method (see Section 6.3.4).

### 1.1. DRIP Requirements Addressed

The following [drip-requirements] will be addressed:

GEN 1: Provable Ownership This will be addressed using the DRIP Link and DRIP Wrapper or DRIP Manifest.

GEN 2: Provable Binding This requirement is addressed using the DRIP Wrapper or DRIP Manifest.

GEN 3: Provable Registration This requirement is addressed using the DRIP Link.

See Section 7.3 for further clarification.

## 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.

Legacy Transports: uses broadcast frames (Bluetooth 4).

Extended Transports: uses the extended advertisements (Bluetooth 5), service info (Wi-Fi NAN) or vendor specific element information (Wi-Fi BEACON). Must use ASTM [F3411] Message Pack (Message Type 0xF).

## 3. Background

### 3.1. Problem Space and Focus

The current standard for Remote ID does not, in any meaningful capacity, address the concerns of trust in the UA space with communication in the Broadcast RID environment. This is a requirement that will need to be addressed eventually for various different parties that have a stake in the UA industry.

#### 3.1.1. Broadcast RID RF Options

A UA has the option of broadcasting using Bluetooth (4 and 5) or Wi-Fi (BEACON or NAN), see Section 7. With Bluetooth, FAA and other CAA mandate transmitting simultaneously over both 4 and 5. With Wi-Fi, use of BEACON is recommended. Wi-Fi NAN is another option, depending on CAA.

Bluetooth 4 presents a payload size challenge in that it can only transmit 25 bytes of payload where the others all can support 252 byte payloads.

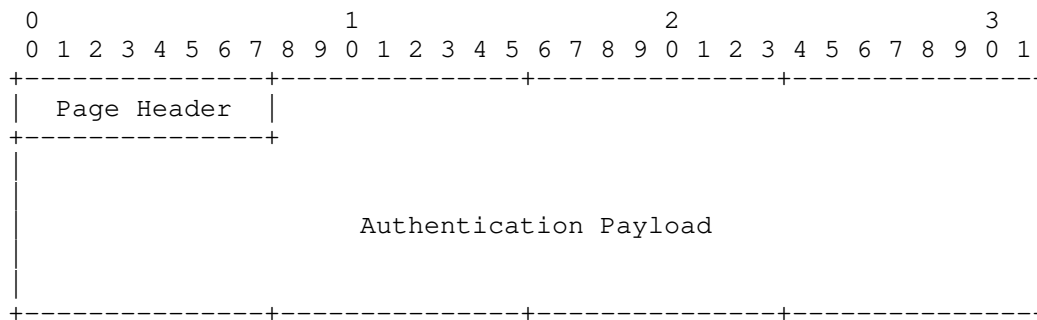
### 3.2. Reasoning for IETF DRIP Authentication

The ASTM Authentication Message has provisions in [F3411] to allow for other organizations to standardize additional Authentication formats beyond those explicitly in [F3411]. The standardization of specific formats to support the DRIP requirements in UAS RID for trustworthy communications over Broadcast RID is an important part of the chain of trust for a UAS ID. No existing formats (defined in [F3411] or other organizations leveraging this feature) provide the functionality to satisfy this goal resulting in the work reflected in this document.

### 3.3. ASTM Authentication Message

The ASTM Authentication Message (Message Type 0x2) is a unique message in the Broadcast [F3411] standard as it is the only one that is larger than the Bluetooth 4 frame size. To address this, it is defined as a set of "pages" that each fits into a single Bluetooth 4 broadcast frame. For other media these pages are still used but all in a single frame.

### 3.3.1. Authentication Page



Page Header: (1 byte)

Authentication Type (4 bits)

Page Number (4 bits)

Authentication Payload: (23 bytes per page)

Authentication Payload, including headers. Null padded.

Figure 1: Standard ASTM Authentication Message Page



A single Authentication Message is akin to a UDP packet. The Authentication Message is structured as a set of up to 16 pages. Over Bluetooth 4, these pages are "fragmented" into separate Bluetooth 4 broadcast frames.

Either as a single Authentication Message or a set of fragmented Authentication Message Pages the structure(s) is further wrapped by outer ASTM framing and the specific link framing (Bluetooth or Wi-Fi).

#### 3.3.1.1. Authentication Type

[F3411] has the following subset of Authentication Type's defined and that can be used in the Page Header:

Authentication Type	Description
0x2	Operator ID Signature
0x3	Message Set Signature
0x5	Specific Authentication Method

Table 1

##### 3.3.1.1.1. Specific Authentication Method (SAM)

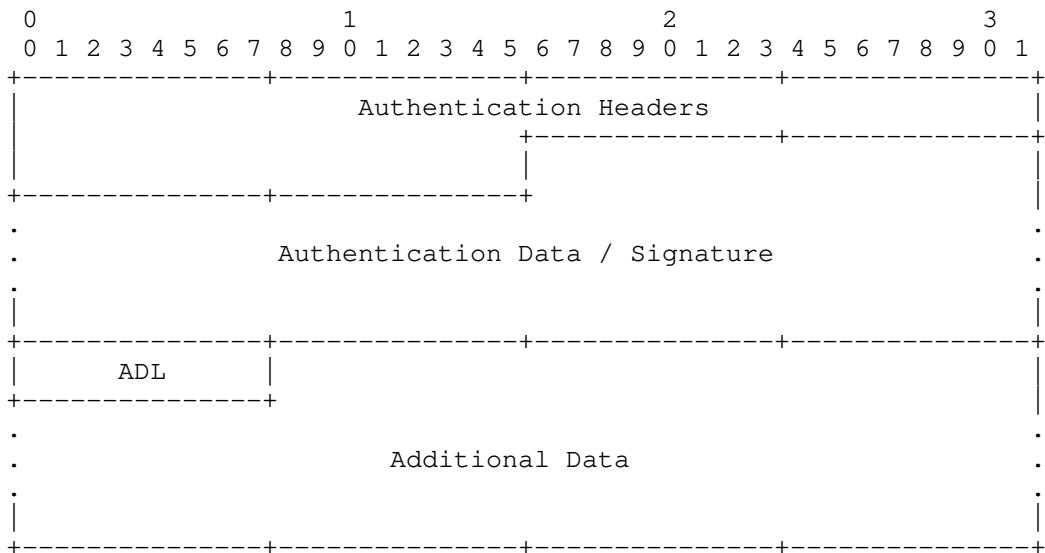
This document leverages Authentication Type 0x5, Specific Authentication Method (SAM), defining a set of SAM Types in Section 6.3. Other Authentication Types are also used in DRIP and their use is defined in Section 6.

##### 3.3.1.2. Page Number

There is a technical maximum of 16 pages (indexed 0 to 15 in the Page Header) that can be sent for a single Authentication Message, with each page carrying a max 23-byte Authentication Payload. See Section 3.3.2 for more details.

##### 3.3.1.3. Authentication Payload Field

The following is shown in its complete format.



Authentication Headers: (6-bytes)  
As defined in F3411.

Authentication Data / Signature: (255-bytes max)  
Opaque authentication data.

Additional Data Length (ADL): (1-byte - unsigned)  
Length in bytes of Additional Data.

Additional Data: (255-bytes max):  
Data that follows the Authentication Data / Signature but  
is not considered part of the Authentication Data.

Figure 2: ASTM Authentication Message Fields

Figure 2 is the source data view of the data fields found in the Authentication Message as defined by [F3411]. This data is placed into Figure 1's Authentication Payload, spanning multiple pages.

When Additional Data is being sent, a single unsigned byte (Additional Data Length) directly follows the Authentication Data / Signature and has the length, in bytes, of the following Additional Data. For DRIP, this field is used to carry Forward Error Correction as defined in Section 4.

### 3.3.2. ASTM Constraints

To keep consistent formatting across the different transports (Legacy and Extended) and their independent restrictions the authentication data being sent is REQUIRED to fit within the page limit of the most constrained existing transport can support. Under Broadcast RID the transport that can hold the least amount of authentication data is Bluetooth 5 and Wi-Fi BEACON at 9-pages.

As such DRIP transmitters are REQUIRED to adhere to the following when using the Authentication Message:

1. Authentication Data / Signature data MUST fit in a 9 pages (Page Numbers 0 through 8).
2. The Length field in the Authentication Headers (which denotes the length in bytes of Authentication Data / Signature only) MUST NOT exceed the value of 201.

## 4. Forward Error Correction

For Broadcast RID, Forward Error Correction (FEC) is provided by the lower layers in Extended Transports (Bluetooth 5, Wi-Fi NaN, and Wi-Fi BEACON). The Bluetooth 4 Legacy Transport does not have supporting FEC so with DRIP Authentication the following application level FEC scheme is used to add FEC. This section is only used for Bluetooth 4 transmission/reception.

### 4.1. Encoding

For any encoding the FEC data MUST start on a new ASTM Authentication Page. To do this, null padding is added before the actual FEC data starts and the length of the whole blob (null padding and FEC) is used as the Additional Data Length. To properly fit FEC data into an Authentication Page the number of parity-bytes is limited to 23 or a multiple thereof (size of Authentication data per page). That is, the Page Header (and anything before it) is omitted in the FEC process.

#### 4.1.1. Single Page FEC

To generate the parity a simple XOR operation using the previous and current page is used. Only the 23-byte Authentication Page data is used in the XOR operation. For Page 0, a 23-byte null pad is used for the previous page. The resulting parity fills the last 23 bytes of the Additional Data field of [F3411] with the Additional Data Length field being set to 23 or greater (depending on number of null pad bytes are needed to get onto the next page).

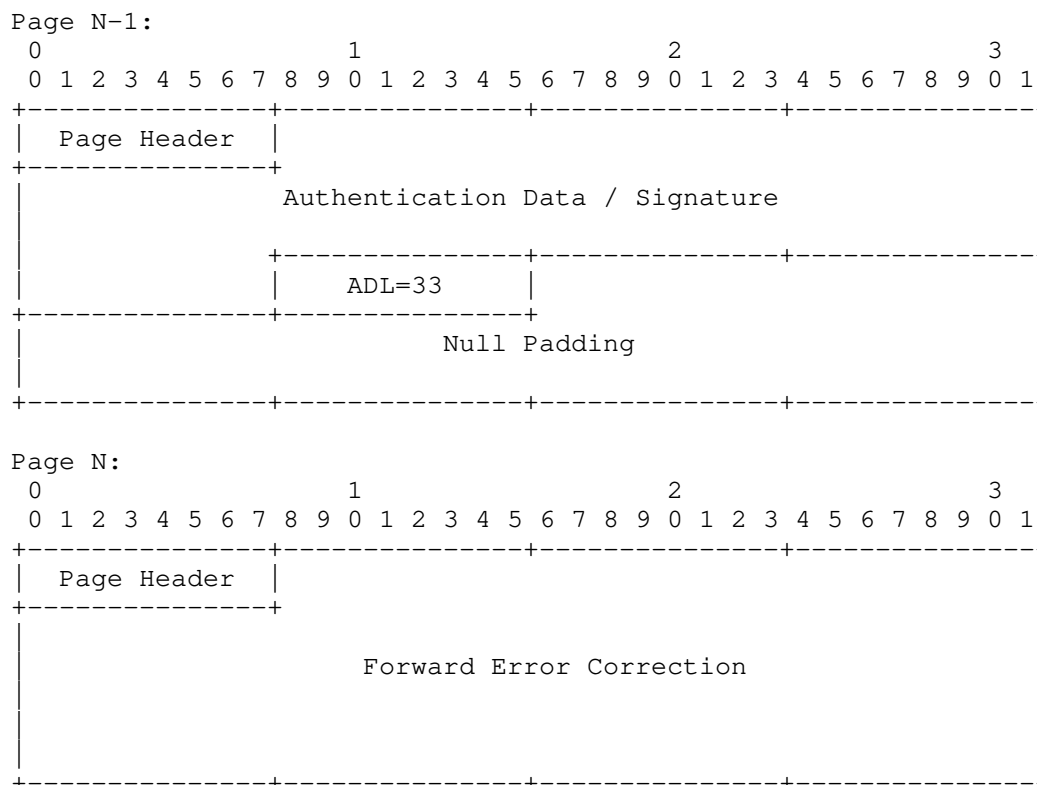


Figure 3: Example Single Page FEC Encoding

#### 4.1.2. Multiple Page FEC

For Multiple Page FEC there are two flavors: Frame Recovery and Page Recovery. Both follow a similar process, but are offset at what data is actually protected.

(Editor Note: to improve interop we MUST explicitly select a polynomial for Reed Solomon for DRIP - need suggestions)

##### 4.1.2.1. Page Recovery

Take the following example of an Authentication Message with 7 pages that 3 pages of parity are to be generated for. The first column is just the Page Header with a visual space here to show the boundary.

```
50 098960bf8c05042001001000a00145aac6b00abba268b7
51 2001001000a0014579d8a404d48f2ef9bb9a4470ada5b4
52 ff1352c7402af9d9ebd20034e8d7a12920f4d7e91c1a73
53 dca7d04e776150825863c512c6eb075a206a95c59b297e
54 f2935fd416f27b1b42fd5d9dfaa0dec79f32287f41b454
55 7101415def153a770d3e6c0b17ae560809bc634a822c1f
56 3b1064b80a0000000000000000000000000000000000
```

For Page Recovery the first column is ignored and the last 23-bytes of each page are extracted to have Reed Solomon performed on them in a column wise fashion to produce parity bytes. For the example the following 3-bytes of parity are generated with the first byte of each page:

```
dc6c2b = ReedSolomon.encoder(0920ffdcf2713b)
```

Each set of parity is the placed into a pseudo-frame as follows (each byte in its own message in the same column). Below is an example of the full parity generated and each 23-bytes of parity added into the additional pages as Additional Data:

57 dc6657acd30b2ec4aa582049f52adf9f922e62c469563a  
58 6c636a59145a55417a3895fd543f19e94200be4abc5e94  
59 02bba5e28f5896d754caf50016a983993b149b5c9e6eeb

#### 4.1.2.2. Frame Recovery

Frame Recovery uses the full ASTM Message and performs Reed Solomon over each byte. Up to 240 (255 minus 15 pages max of FEC data) messages can be protected using Frame Recovery.

Below is an example of a number of messages. Here the first column is an additional ASTM Header that contain the Message Type; with a visual space for clarity. The last 24-bytes are the actual message contents; be it location information or an Authentication Page.

```

10 42012001001000a0014579d8a404d48f2ef9000000000000
11 249600006efeb019ee111ed37a097a0948081c10ffff0000
12 50098960bf8c05042001001000a00145aac6b00abba268b7
12 512001001000a0014579d8a404d48f2ef9bb9a4470ada5b4
12 52ff1352c7402af9d9ebd20034e8d7a12920f4d7e91c1a73
12 53dca7d04e776150825863c512c6eb075a206a95c59b297e
12 54f2935fd416f27b1b42fd5d9dfaa0dec79f32287f41b454
12 557101415def153a770d3e6c0b17ae560809bc634a822c1f
12 563b1064b80a000000000000000000000000000000000000
13 0052656372656174696f6e616c2054657374000000000000
14 02c2fffb019322d1ed301000c008e40700fc080000000000
15 004e2e4f5031323334353600000000000000000000000000

```

A similar process is followed as in Section 4.1.2.1. Here every column of bytes has parity generated for it (even the ASTM Header). In the below example 5-bytes of parity are generated using the ASTM Header column:

```
6c3f42b8a8 = ReedSolomon.encoder(101112121212121212131415)
```

After doing this to all columns the following pseudo-frames would have been generated:

```
6c86337bf7ab746f5d62bb7f8de954104b121585d3975f6e92
3f06c1bce165b0e25930d57a63c24f751145e1dd8dc115029b
42e9979580327a6a14d421c12a33aa2e1a2e517daaee581016
b8012a7b3964f7b2720d387bfa77e945556f1831cd477ef3a3
a85bb403aada89926fb8fc2a14a9caacb4ec2f3a6ed2d8e9f9
```

These 25-byte chunks are now concatenated together and are placed in Authentication Pages, using the Additional Data, 23-bytes at a time. In the below figure the first column is the ASTM Header as before, the second column is the Page Header for each Authentication Page and then last column is the 23-bytes of data for each page.

```
12 57 6c86337bf7ab746f5d62bb7f8de954104b121585d3975f
12 58 6e923f06c1bce165b0e25930d57a63c24f751145e1dd8d
12 59 c115029b42e9979580327a6a14d421c12a33aa2e1a2e51
12 5a 7daaee581016b8012a7b3964f7b2720d387bfa77e94555
12 5b 6f1831cd477ef3a3a85bb403aada89926fb8fc2a14a9ca
12 5c acb4ec2f3a6ed2d8e9f90000000000000000000000000000
```

#### 4.2. Decoding

Due to the nature of Bluetooth 4 and the existing ASTM paging structure an optimization can be used. If a Bluetooth frame fails its CRC check, then the frame is dropped without notification to the upper protocol layers. From the Remote ID perspective this means the loss of a complete frame/message/page. In Authentication Messages, each page is already numbered so the loss of a page allows the receiving application to build a "dummy" page filling the entire page with nulls.

If Page 0 is being reconstructed an additional check of the Last Page Index to check against how many pages are actually present, MUST be performed for sanity. An additional check on the Length field SHOULD also be performed.

To determine if Single Page FEC or Multiple Page FEC has been used a simple check of the Last Page Index can be used. If the number of pages left after the Length of Authentication Data is exhausted than

it is clear that the remaining pages are all FEC. The Additional Data Length byte can further confirm this; taking into account any null padding needed for page alignment.

#### 4.2.1. Single Page FEC

Using the same methods as encoding, an XOR operation is used between the previous and current page (a 23-byte null pad is used as the start). The resulting 23-bytes should be data of the missing page.

#### 4.2.2. Multiple Page FEC

To determine if Page Recovery or Frame Recovery is used two modulo checks with the ADL after the length of the null-pad is removed are needed. One against the value of 23, and the other against the value of 25. If 23 comes back with a value of 0 then Page Recovery is being used. If 25 comes back with 0 then Frame Recovery is used. Any other combination indicates an error.

##### 4.2.2.1. Page Recovery

To decode Page Recovery, dummy pages (pages with nulls as the data) are needed in the places no page was received. Then Reed Solomon can decode across the columns of the 23-bytes of each page. Erasures can be used as it is known which pages are missing and can improve the Reed Solomon results by specifying them.

##### 4.2.2.2. Frame Recovery

To decode Frame Recovery, the receiver must first extract all FEC data from the pages; concatenate them and then break into 25-byte chunks. This will produce the pseudo-frames. Now Reed Solomon can be used to decode columns, with dummy frames inserted where needed.

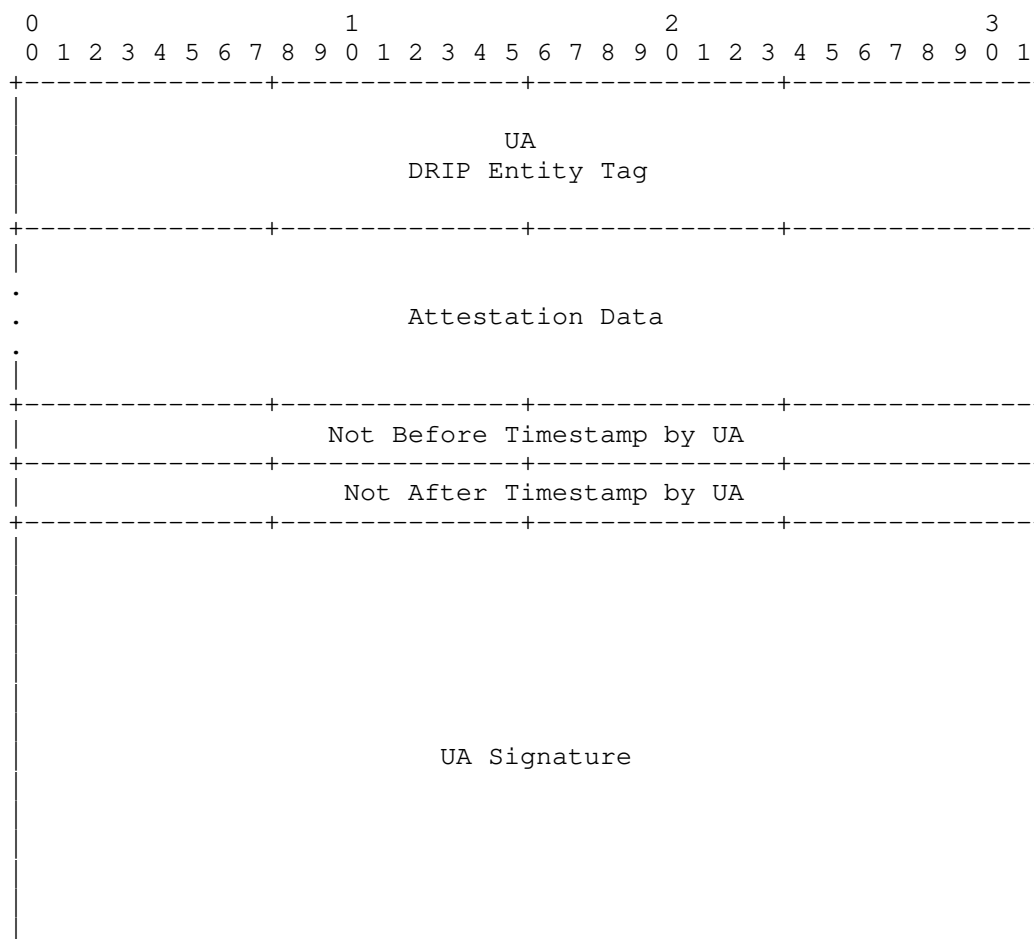
#### 4.3. FEC Limitations

The worst case scenario is when the Authentication Data / Signature ends perfectly on a page (Page N-1). This means the Additional Data Length would start the next page (Page N) and have 22-bytes worth of null padding to align the FEC in to the next page (Page N+1). In this scenario an entire page (Page N) is being wasted just to carry the Additional Data Length. This should be avoided at all costs - in an effort to maintain efficiency.

## 5. Broadcast Attestation Structure

To directly support Broadcast RID a variation of the Attestation Structure format of [drip-registries] SHOULD be used when running DRIP under the various Authentication Types (filling the Authentication Data / Signature field of Figure 2) and SAM Types (filling the SAM Authentication Data field (Section 6.3.1.2)). The notable changes of the structure is that the timestamps are set by the UA and the Attestor Identity Information is set to the DET of the UA.

When using this structure the UA is always self-attesting its DRIP Entity Tag (DET). The Host Identity of the UA DET can be looked up by mechanisms described in [drip-registries] or by extracting it from Broadcast Attestation (see Section 6.3.2 and Section 7.3).





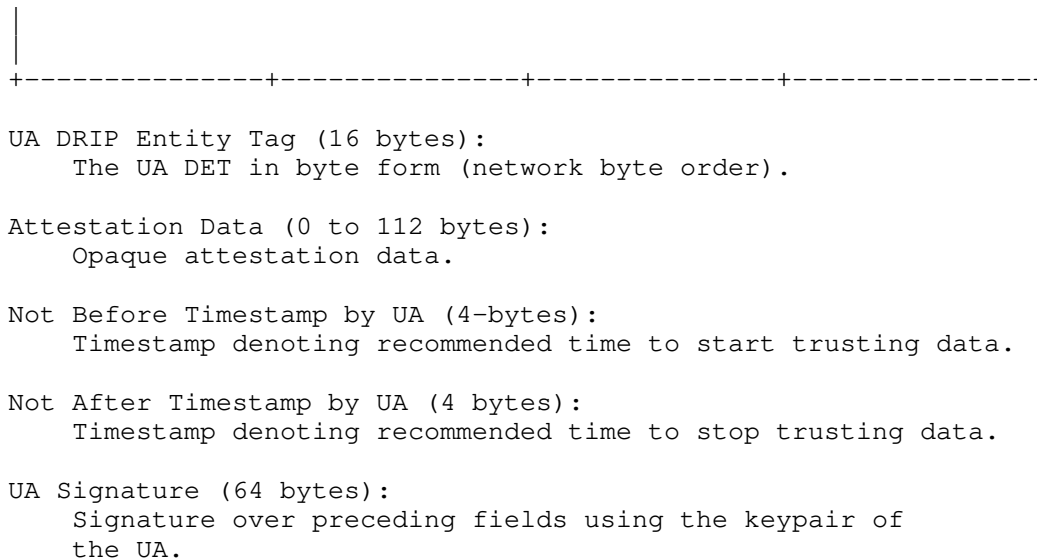


Figure 4: Broadcast Attestation Structure

Attestation Data is a field with a maximum of 112-bytes, containing data that the UA is attesting during its flight.

The Not After Timestamp and Not Before Timestamp MUST follow the format defined in [F3411]. That is a Unix-style timestamp but with an epoch of 01/01/2019 00:00:00. Not Before Timestamp MUST be set to the time the structure is signed over. An additional offset is then added to push the Not After Timestamp a short time into the future to avoid replay attacks.

The offset used against the Unix-style timestamp is not defined in this document. Best practice identifying an acceptable offset should be used taking into consideration the UA environment, and propagation characteristics of the messages being sent and clock differences between the UA and Observers. A reasonable time would be to set Not After Timestamp 2 minutes ahead of Not Before Timestamp.

## 6. DRIP Authentication Formats

All formats defined in this section are the content for the Authentication Data / Signature field in Figure 2.

When sending data over a medium that does not have underlying Forward Error Correction (FEC), for example Bluetooth 4, then Section 4 MUST be used.

## 6.1. DRIP Authentication Field Definitions

UA DRIP Entity Tag (16-bytes): The UA DET in byte form (network byte order)

Not Before Timestamp by UA (4-bytes): Timestamp denoting recommended time to start trusting data. MUST follow the format defined in [F3411]. That is a Unix-style timestamp but with an epoch of 01/01/2019 00:00:00. MUST be set to the time the signature is generated.

Not After Timestamp by UA (4-bytes): Timestamp denoting recommended time to stop trusting data. MUST follow the format defined in [F3411]. That is a Unix-style timestamp but with an epoch of 01/01/2019 00:00:00 with an additional offset is then added to push a short time into the future (relative to Not Before Timestamp) to avoid replay attacks. The offset used against the Unix-style timestamp is not defined in this document. Best practice identifying an acceptable offset should be used taking into consideration the UA environment, and propagation characteristics of the messages being sent and clock differences between the UA and Observers. A reasonable time would be to set Not After Timestamp 2 minutes ahead of Not Before Timestamp.

UA Signature (64-bytes): Signature over preceding fields using the keypair of the UA.

ASTM Message (25-bytes): Full ASTM Message as defined in [F3411] specifically Message Types 0x0, 0x1, 0x3, 0x4, and 0x5

Previous Manifest Hash (12-bytes): See Section 6.3.4.2.

Current Manifest Hash (12-bytes): See Section 6.3.4.2.

ASTM Message Hash (12-bytes): Hash of a single full ASTM Message using hash operations described in (Section 6.3.4.1). Multiple hashes MUST be in Message Type order.

Broadcast Attestation (136-bytes): HDA over UA. Generated by a DRIP Registry during Session ID registration. Used in Section 6.3.2.

Frame Type (1-byte): Sub-type for future different DRIP Frame formats. See Section 6.3.5.1.

Attestation Data (0 to 111 bytes): Opaque attestation data.

## 6.2. Message Set Signature

When running under Extended Transports, the Authentication Message can be used to sign other messages within the Message Pack (Message Type 0xF).

To generate the below structure the sender must concatenate all of the messages in the Message Pack (excluding Authentication) in Message Type order. This blob of Message is then set between the UA DRIP Entity Tag and Not Before Timestamp to generate the signature. The blob of Messages is then removed as it is redundant and the below structure is placed into an Authentication Message (of Authentication Type 0x3) to be sent in the same Message Pack.

To verify the signature the receiver must concatenate all of the messages in the Message Pack (excluding Authentication Message found in the same Message Pack) in Message Type order and place the blob between the UA DRIP Entity Tag and Not Before Timestamp before performing signature verification.

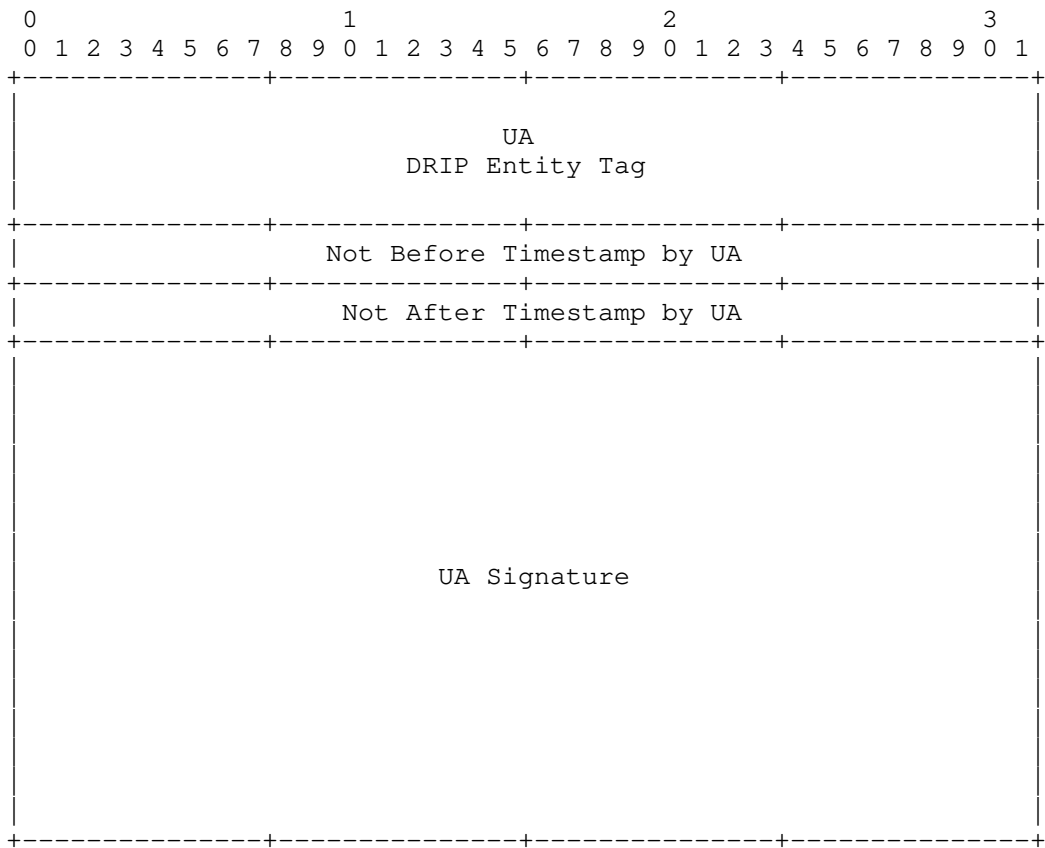


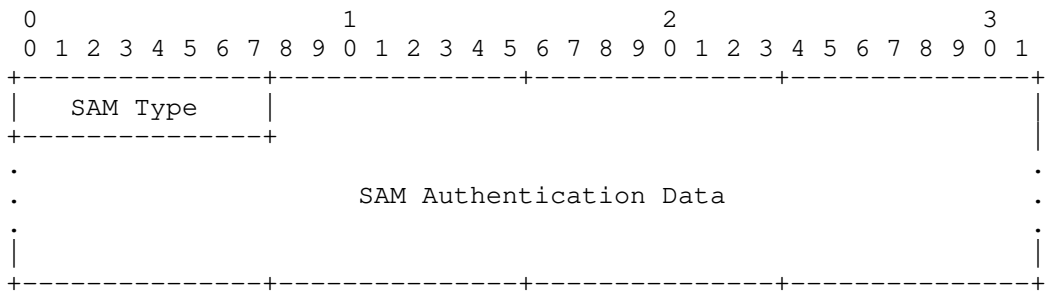
Figure 5: DRIP Message Set Signature

6.3. Specific Authentication Method

DRIP Authentication uses the Specific Authentication Method (SAM, Authentication Type 0x5). The first byte of the Authentication Data / Signature of Figure 2, is used to multiplex between various formats.

6.3.1. SAM Data Format

Figure 6 is the general format to hold authentication data when using SAM and is placed inside the Authentication Data / Signature field in Figure 2.



SAM Type (1 byte):  
Byte defined by F3411 to multiplex SAMs

SAM Authentication Data (0 to 200 bytes):  
Opaque SAM authentication data.

Figure 6: SAM Data Format

6.3.1.1. SAM Type

The SAM Type field is maintained by the International Civil Aviation Organization (ICAO) and for DRIP four are planned to be allocated:

SAM Type	Description
0x01	DRIP Link (Section 6.3.2)
0x02	DRIP Wrapper (Section 6.3.3)
0x03	DRIP Manifest (Section 6.3.4)
0x04	DRIP Frame (Section 6.3.5)

Table 2

6.3.1.2. SAM Authentication Data

This field has a maximum size of 200-bytes, as defined by Section 3.3.2. When possible the Broadcast Attestation Structure (Section 5) should be used in this space.

### 6.3.2. DRIP Link

This SAM Type is used to transmit Broadcast Attestation's. The Broadcast Attestation of the Registry (HDA) over the UA MUST be sent (see Section 7.3). Its structure is defined in [drip-registries] and an example of it can be found in Appendix B.

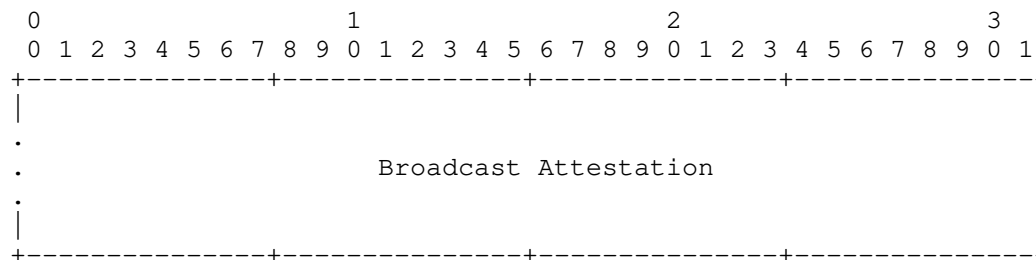


Figure 7: DRIP Link

This DRIP format MUST be used in conjunction with another DRIP SAM Type (such as Manifest or Wrapper) that contains data that is guaranteed to be unique and easily cross checked by the receiving device. A good candidate for this is using the DRIP Wrapper to encapsulate a Location Message (Message Type 0x2).

#### 6.3.2.1. Link Limitations

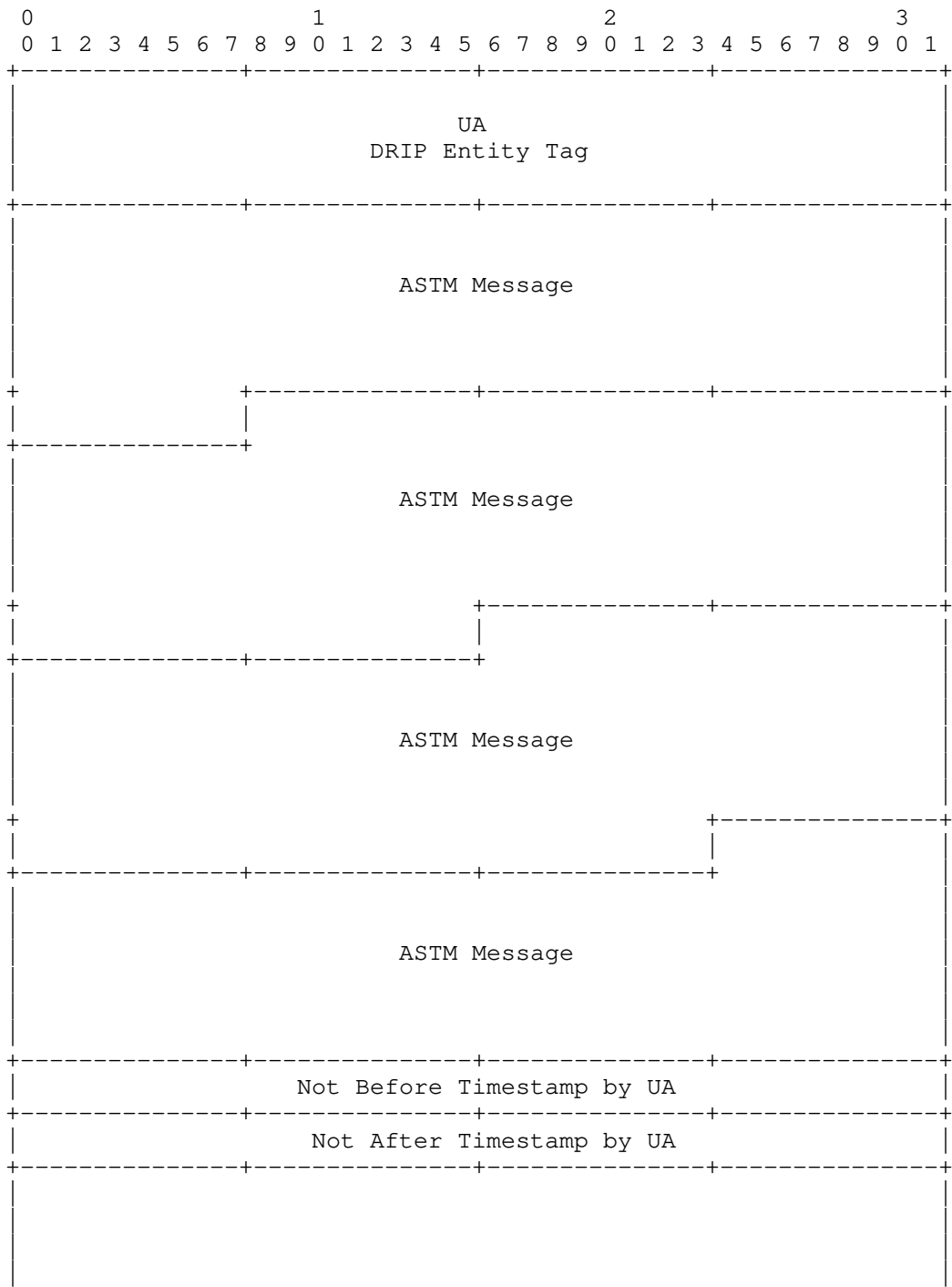
See Section 10.2 for details on why this structure is not dynamically signed.

### 6.3.3. DRIP Wrapper

This SAM Type is used to wrap and sign over a list of other [F3411] Broadcast RID messages. It MUST use the Broadcast Attestation Structure (Section 5).

The Attestation Data field is filled with full (25-byte) [F3411] Broadcast RID messages. The minimum number being 1 and the maximum being 4. The encapsulated messages MUST be in Message Type order as defined by [F3411]. All message types except Authentication (Message Type 0x2) and Message Pack (Message Type 0xF) are allowed.

To determine the number of messages wrapped the receiver can check that the length of the Attestation Data field of the DRIP Broadcast Attestation (Section 5) is a multiple of 25-bytes.



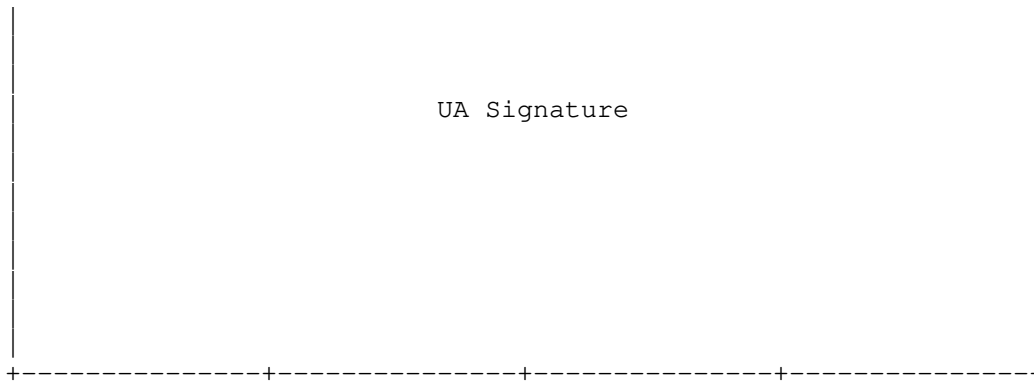


Figure 8: Example 4-Message DRIP Wrapper

#### 6.3.3.1. Wrapper Limitations

The primary limitation of the Wrapper format is the bounding of up to 4 ASTM Messages that can be sent within it. Another limitation is that the format can not be used as a surrogate for messages it is wrapping. This is due to high potential a receiver on the ground does not support DRIP. Thus when Wrapper is being used the wrapper data must effectively be sent twice; once as a single framed message (as specified in [F3411]) and then again wrapped within the Wrapper format.

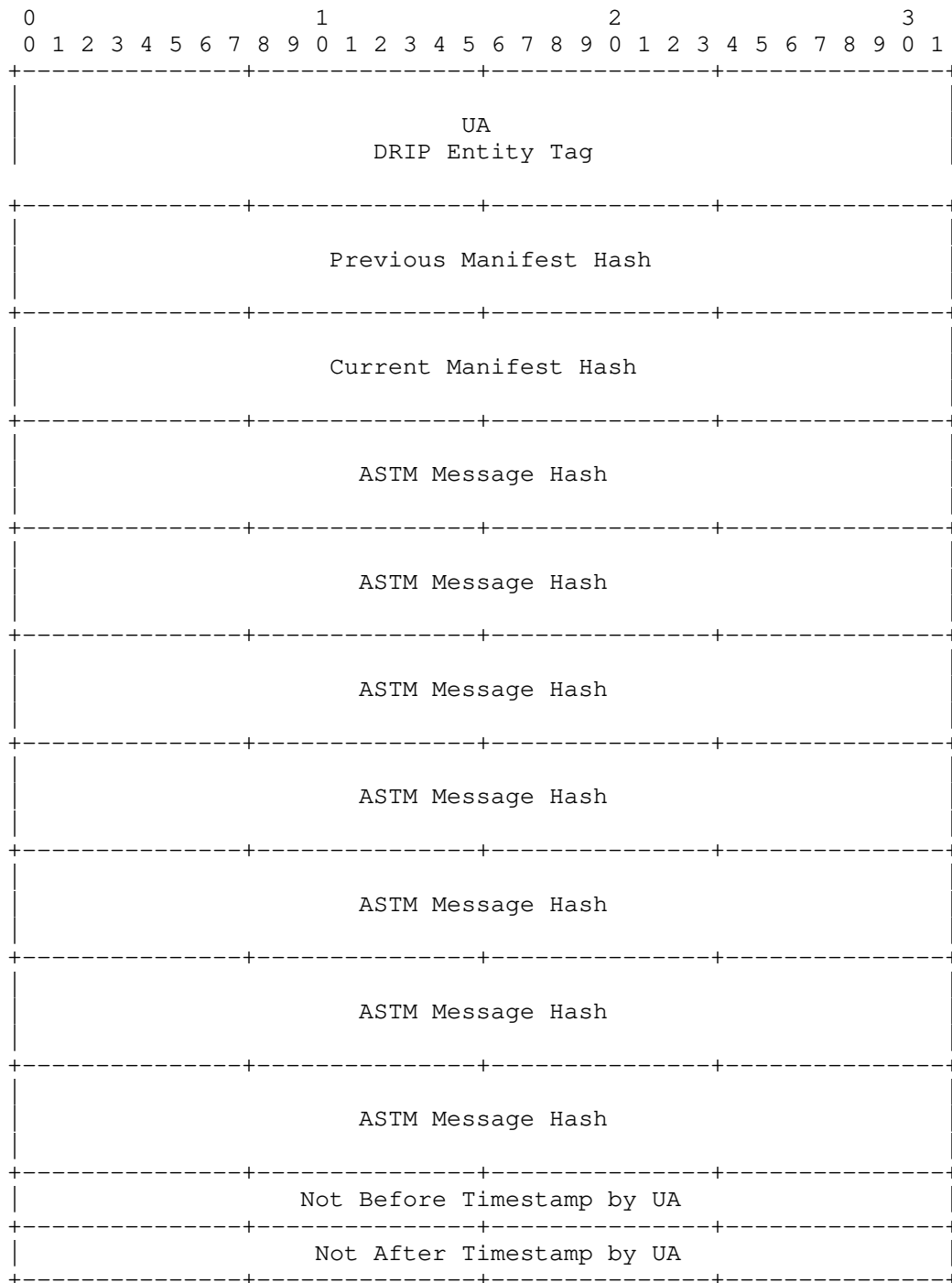
#### 6.3.4. DRIP Manifest

This SAM Type is used to create message manifests. It MUST use the Broadcast Attestation Structure (Section 5).

By hashing previously sent messages and signing them we gain trust in UAs previous reports. An observer who has been listening for any considerable length of time can hash received messages and cross-check against listed hashes. This is a way to evade the limitation of a maximum of 4 messages in the Wrapper Format and reduce overhead.

The Attestation Data field is filled with 12-byte hashes of previous [F3411] Broadcast messages. A receiver does not need to have received every message in the manifest to verify it. A manifest SHOULD typically encompass a single transmission cycle of messages being sent, see Section 7.4.





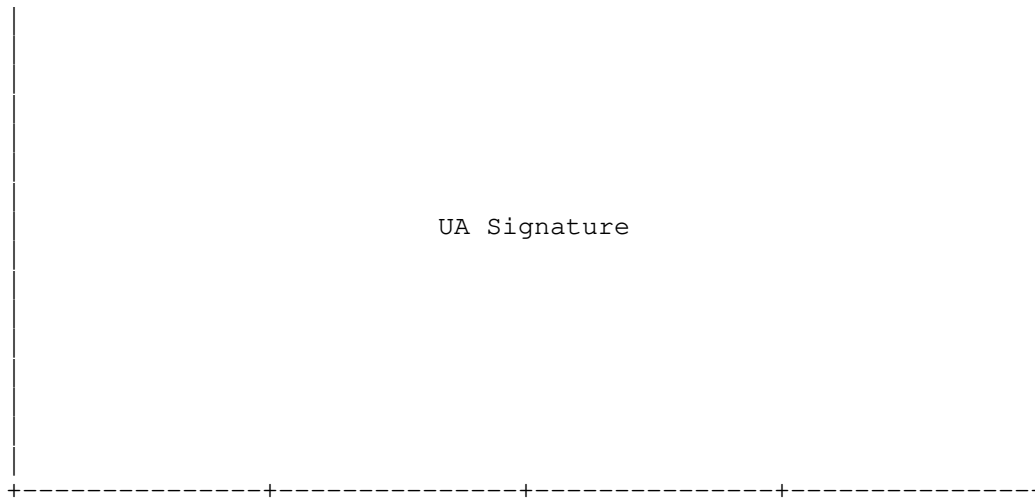


Figure 9: Example DRIP Manifest

#### 6.3.4.1. Message Hash Algorithms and Operation

The hash algorithm used for the Manifest Message is the same hash algorithm used in creation of the DET [drip-rid] that is signing the Manifest.

An DET using cSHAKE128 [NIST.SP.800-185] computes the hash as follows:

```
cSHAKE128(ASM Message, 96, "", "Remote ID Auth Hash")
```

Note: [drip-rid] specifies cSHAKE128 but is open for the expansion of other OGAs.

##### 6.3.4.1.1. Legacy Transport Hashing

Under this transport DRIP hashes the full ASTM Message being sent over the Bluetooth Advertising frame. For Authentication Messages all the Authentication Message Pages are concatenated together and hashed as one object. For all other Message Types the 25-byte message is hashed.

##### 6.3.4.1.2. Extended Transport Hashing

Under this transport DRIP hashes the full ASTM Message Pack (Message Type 0xF) - regardless of its content.

#### 6.3.4.2. Pseudo-Blockchain Hashes

Two special hashes are included in all Manifest messages; a previous manifest hash, which links to the previous manifest message, as well as a current manifest hash. This gives a pseudo-blockchain provenance to the manifest message that could be traced back if the observer was present for extended periods of time.

Creation: During creation and signing of this message format this field MUST be set to 0. So the signature will be based on this field being 0, as well as its own hash. It is an open question of if we compute the hash, then sign or sign then compute.

Cycling: There are a few different ways to cycle this message. We can "roll up" the hash of 'current' to 'previous' when needed or to completely recompute the hash. This mostly depends on the previous note.

#### 6.3.4.3. Manifest Limitations

A potential limitation to this format is dwell time of the UA. If the UA is not sticking to a general area then most likely the Observer will not obtain many (if not all) of the messages in the manifest. Examples of such scenarios include delivery or survey UA.

Another limitation is the length of hash, which is discussed in Section 10.1.

#### 6.3.5. DRIP Frame

This SAM Type is for when the authentication data does not fit in other defined formats under DRIP and is reserved for future expansion under DRIP if required. This SAM Type SHOULD use the Broadcast Attestation Structure (Section 5).

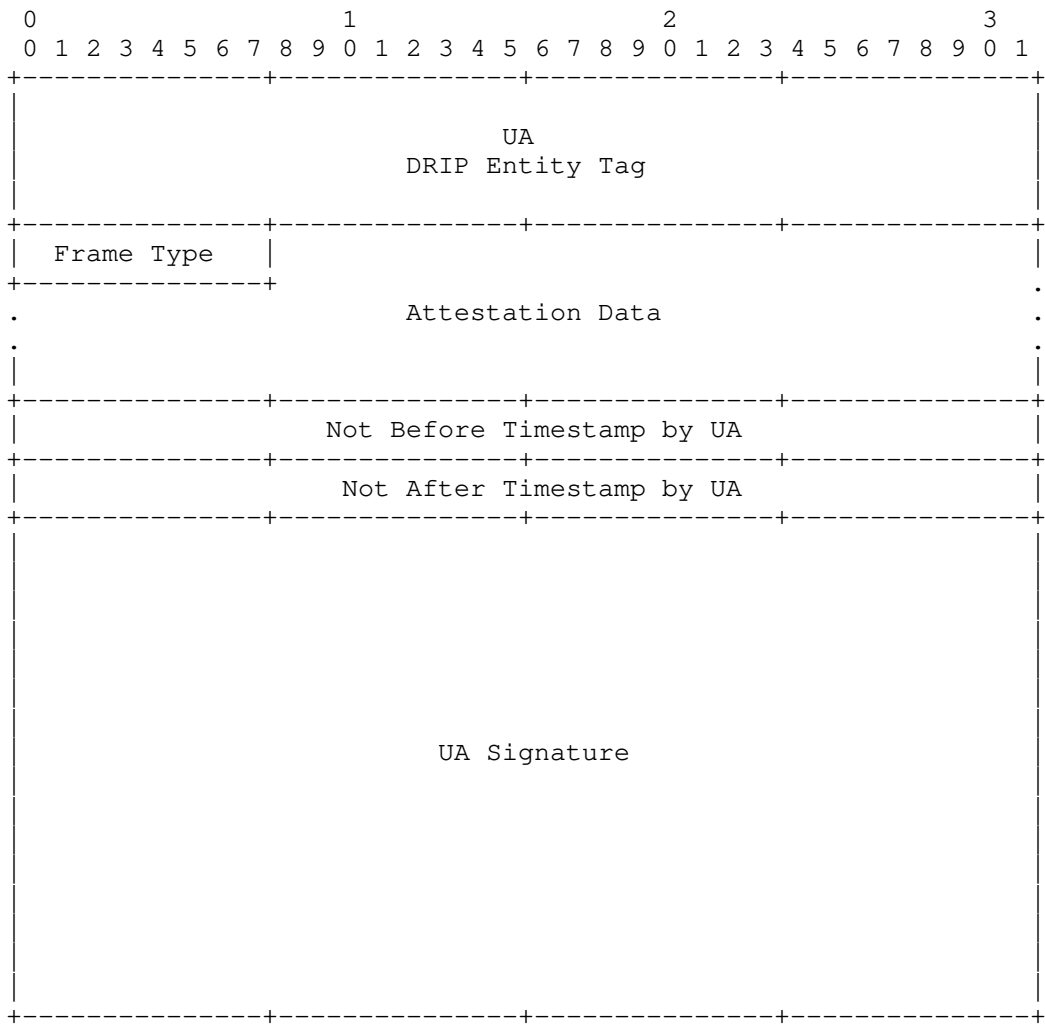


Figure 10: Example DRIP Frame

6.3.5.1. Frame Type

Byte to sub-type for future different DRIP Frame formats.

Frame Type	Name	Description
0x00	Reserved	Reserved
0xC0-0xFF	Experimental	Experimental Use

Table 3

#### 6.3.5.2. Frame Limitations

With the Broadcast Attestation Structure only 115-bytes of Attestation Data are free for use.

### 7. Requirements & Recommendations

#### 7.1. Legacy Transports

With Legacy Advertisements the goal is to attempt to bring reliable receipt of the paged Authentication Message. Forward Error Correction (Section 4) MUST be used when using Legacy Advertising methods (such as Bluetooth 4).

Under ASTM Bluetooth 4 rules, transmission of dynamic messages are at least every 1 second. DRIP Authentication Messages typically contain dynamic data (such as the DRIP Manifest or DRIP Wrapper) and must be sent at the dynamic rate of 1 per second.

#### 7.2. Extended Transports

Under the ASTM specification, Bluetooth 5, Wi-Fi NaN, and Wi-Fi BEACON transport of Remote ID is to use the Message Pack (Message Type 0xF) format for all transmissions. Under Message Pack messages are sent together (in Message Type order) in a single Bluetooth 5 extended frame (up to 9 single frame equivalent messages under Bluetooth 4). Message Packs are required by ASTM to be sent at a rate of 1 per second (like dynamic messages).

Without any fragmentation or loss of pages with transmission Forward Error Correction (Section 4) MUST NOT be used as it is impractical.

#### 7.3. Authentication

It is REQUIRED that a UA send the following Authentication Formats to fulfill the [drip-requirements]:

1. DRIP Link using the Broadcast Attestation of HDA and the UA (satisfying GEN-1 and GEN-3)
2. Any other DRIP Authentication Format (RECOMMENDED: DRIP Manifest or DRIP Wrapper) where the UA is dynamically signing data (satisfying GEN-1 and GEN-2)

It is RECOMMENDED the following set of Authentication Formats are sent for support of offline Observers:

1. DRIP Link using the Broadcast Attestation of HID Root and the RAA (CAA) (satisfies GEN-3)
2. DRIP Link using the Broadcast Attestation of RAA (CAA) and the HDA (USS) (satisfies GEN-3)
3. DRIP Link using the Broadcast Attestation of HDA (USS) and the UA (satisfies GEN-1 and GEN-3)
4. Any other DRIP Authentication Format (RECOMMENDED: DRIP Manifest or DRIP Wrapper) where the UA is dynamically signing data (satisfies GEN-1 and GEN-2)

#### 7.4. Operational

UAS operation may impact the frequency of sending DRIP Authentication messages. Where a UA is dwelling in one location, and the channel is heavily used by other devices, "occasional" message authentication may be sufficient for an observer. Contrast this with a UA traversing an area, and then every message should be authenticated as soon as possible for greatest success as viewed by the receiver.

Thus how/when these DRIP authentication messages are sent is up to each implementation. Further complication comes in contrasting Legacy and Extended Transports. In Legacy, each message is a separate hash within the Manifest. So, again in dwelling, may lean toward occasional message authentication. In Extended Transports, the hash is over the Message Pack so only few hashes need to be in a Manifest. A single Manifest can handle a potential two Message Packs (for a full set of messages) and a DRIP Link Authentication Message for the HDA UA assertion.

A separate issue is the frequency of transmitting the DRIP Link Authentication Message for the HDA UA assertion when using a Manifest Message. This message content is static; its hash never changes radically. The only change is the 4-byte timestamp in the Authentication Message headers. Thus, potentially, in a dwelling operation it can be sent once per minute, where its hash is in every Manifest. A receiver can cache all DRIP Link Authentication Message for the HDA UA assertion to mitigate potential packet loss.

The preferred mode of operation is to send the HDA UA assertion every 3 seconds and Manifest messages immediately after a set of UA operation messages (e.g. Basic, Location, and System messages).

#### 7.4.1. DRIP Wrapper

The DRIP Wrapper MUST NOT be used in place of sending the ASTM messages as is. All receivers MUST be able to process all the messages specified in [F3411]. Only sending them within the DRIP Wrapper will make them opaque to receivers lacking support for DRIP authentication messages. Thus messages within a Wrapper are sent twice: in the clear, and authenticated within the Wrapper. The DRIP Manifest format would seem to be a more efficient use of the transport channel.

The DRIP Wrapper has a specific use case for DRIP aware receivers. For receiver plotting received Location Messages (Message Type 0x2) on a map display an embedded Location Message in a DRIP Wrapper can be colored differently to signify trust in the Location data - be it current or previous Location reports that are wrapped.

### 8. ICAO Considerations

DRIP requests the following SAM Type's to be allocated:

1. DRIP Link
2. DRIP Wrapper
3. DRIP Manifest
4. DRIP Frame

### 9. IANA Considerations

This document requests a new number field for Frame Type with initial values as defined in Section 6.3.5.1.

## 10. Security Considerations

### 10.1. Manifest Hash Length

For DRIP Manifest an 12-byte hash length has been selected by the authors for a number of reasons.

1. Hash lengths smaller than 8-bytes (for example 4-bytes) were originally contemplated but ruled out by comments by various cryptographers. The main concern raised in this forum was that the length of hash would not provide strong resistance against collision rate. The authors also after further review agreed with this and also realized operationally it was not necessarily viable. While 4-byte hashes would allow more messages to be filled into a single DRIP Manifest payload (up to 22 individual hashes) the length of time for the UA to stay in a single place where the Observer would receive all the originally messages to rehash to verify such a message was impractical.
2. Hash lengths larger than 8-bytes (for example 12 or 16-bytes) were also considered by the authors. These got the approval of the cryptographers but the number of hashes to send became much lower (only 5 individual hashes). While this lower number is a more reasonable number of original messages the Observer would have to capture it would also mean that potentially more DRIP Manifests would need to be sent. Overall the increase length of the hash did not operationally justify the cost.
3. Simplifying the current design and locking it into using the same hash as the HHIT instead of allowing for agility in either hash algorithm or length seemed more realistic to the authors today.

### 10.2. Replay Attacks

The astute reader may note that the DRIP Link messages, which are recommended to be sent, are static in nature and contain various timestamps. These Attestation Link messages can easily be replayed by an attacker who has copied them from previous broadcasts. There are two things to mitigate this in DRIP:

1. If an attacker (who is smart and spoofs more than just the UAS ID/data payloads) willing replays an Attestation Link message they have in principle actually helped by ensuring the message is sent more frequently and be received by potential Observers.
2. It is RECOMMENDED to send more than just DRIP Link messages, specifically those that sign over changing data using the current session keypair, and those messages are sent more frequently. An



UA beaconing these messages then actually signing other messages using the keypair validates the data receiver by an Observer. An UA who does not either run DRIP themselves or does not have possession of the same private key, would be clearly exposed upon signature verification.

### 10.3. Trust Timestamp Offsets

Note the discussion of Trust Timestamp Offsets here is in context of the DRIP Wrapper (Section 6.3.3) and DRIP Manifest (Section 6.3.4) messages. For DRIP Link (Section 6.3.2) messages these offsets are set by the Attestor (typically a registry) and have their own set of considerations as seen in [drip-registries].

The offset of the Trust Timestamp (defined as a very short Expiration Timestamp) is one that needs careful consideration for any implementation. The offset should be shorter than any given flight duration (typically less than an hour) but be long enough to be received and processed by Observers (larger than a few seconds). It is recommended that 3-5 minutes should be sufficient to serve this purpose in any scenario, but is not limited by design.

## 11. Acknowledgments

Ryan Quigley and James Mussi of AX Enterprize, LLC for early prototyping to find holes in the draft specifications.

Soren Friis for pointing out that Wi-Fi implementations would not always give access to the MAC Address, originally used in calculation of the hashes for DRIP Manifest. Also, for confirming that Message Packs (0xF) can only carry up to 9 ASTM frames worth of data (9 Authentication pages) - this drove the requirement for max page length of Authentication Data itself.

## 12. References

### 12.1. Normative References

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

[NIST.SP.800-185]

Kelsey, J., Change, S., and R. Perlner, "SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash and ParallelHash", NIST Special Publication SP 800-185, DOI 10.6028/nist.sp.800-185, December 2016, <<http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-185.pdf>>.

- [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-registries] Wiethuechter, A., Card, S., and R. Moskowitz, "DRIP Registries", Work in Progress, Internet-Draft, draft-wiethuechter-drip-registries-01, 22 October 2021, <<https://www.ietf.org/archive/id/draft-wiethuechter-drip-registries-01.txt>>.
- [drip-requirements] Card, S., Ed., Wiethuechter, A., Moskowitz, R., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Requirements and Terminology", RFC 9153, DOI 10.17487/RFC9153, February 2022, <<https://www.rfc-editor.org/info/rfc9153>>.
- [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>>.

## Appendix A. Authentication State Diagrams & Color Scheme

ASTM Authentication has only 3 states: None, Invalid or Valid. This is because under ASTM the idea is that Authentication is done by an external service hosted somewhere on the Internet so it is assumed you will always get some sort of answer back. With DRIP this classification becomes more complex with the support of "offline" scenarios where the receiver does not have Internet connectivity. With the use of asymmetric keys this means the public key (PK) must somehow be obtained - [drip-registries] gets more into detail how these keys are stored on DNS and one reason for DRIP Authentication is to send PK's over Broadcast RID.

There are two keys of interest: the PK of the UA and the PK of the HDA (or Registry). This document gives a clear way to send the PK of the UA over the Broadcast RID messages - however the PK of the Registry is not. It can be using the same mechanism but is not

required to do so due to potential operational constraints and implementation of a given UA transmitter. As such there are scenarios where you may have part of the key-chain but not all of it.

The intent of this appendix is to give some kind of recommended way to classify these various states and convey it to the user through colors and state names/text.

#### A.1. State Table

The table below lays out the RECOMMENDED colors to associate with state.

State	Color	Details
None	Black	No Authentication being received
Partial	Gray	Authentication being received but missing pages
Unsupported	Brown	Authentication Type/SAM Type of received message not supported
Unverifiable	Yellow	Data needed for verification missing
Verified	Green	Valid verification results
Trusted	Blue	Valid verification results and HDA is marked as trusted
Questionable	Orange	Inconsistent verification results
Unverified	Red	Invalid verification results
Conflicting	Purple	Inconsistent verification results and HDA is marked as trusted

Table 4

#### A.2. State Diagrams

This section gives some RECOMMENDED state flows that DRIP should follow.

## A.2.1. Notations

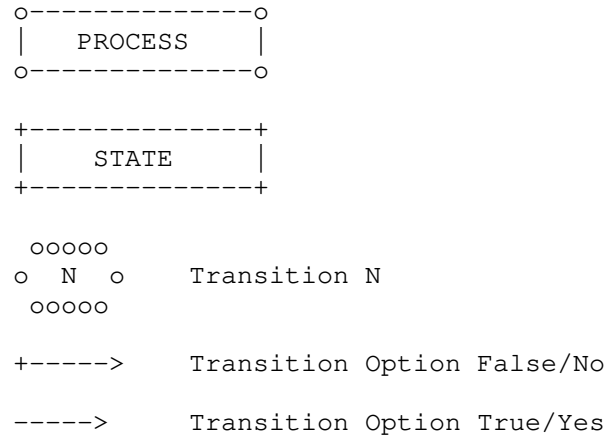


Figure 11: Diagram Notations

## A.2.2. General

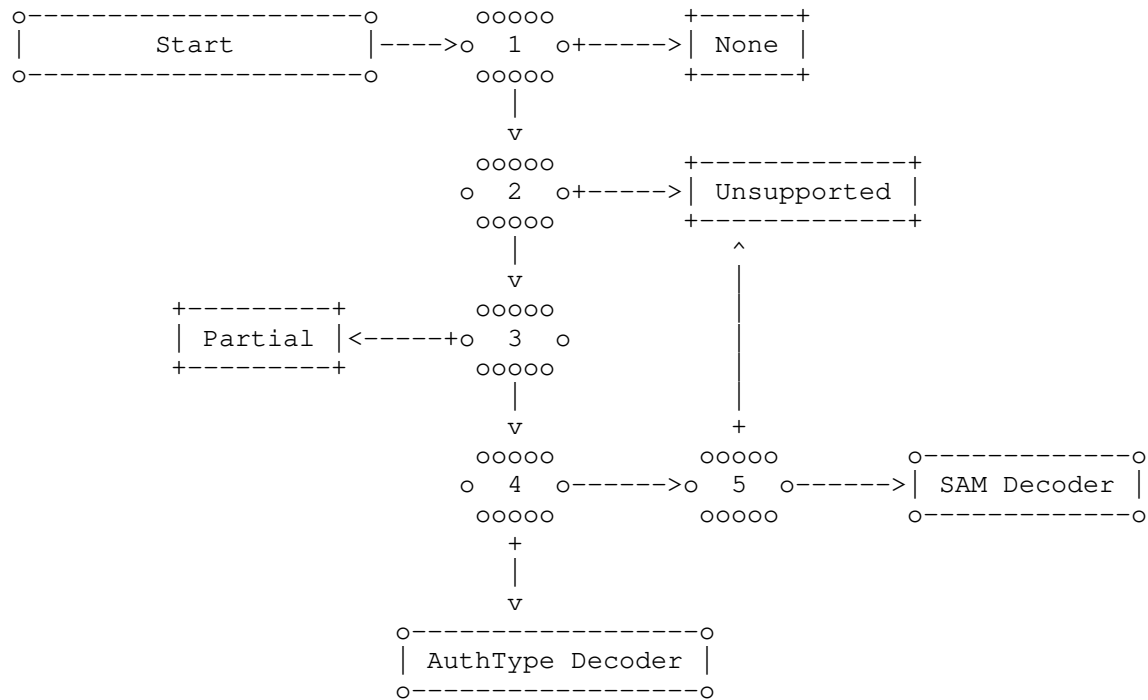


Figure 12: Standard Authentication Colors/State

Transition	Transition Query	Next State/Process/ Transition (Yes, No)
1	Receiving Authentication Pages?	2, None
2	Authentication Type Supported?	3, Unsupported
3	All Pages of Authentication Message Received?	4, Partial
4	Is Authentication Type received 5?	5, AuthType Decoder
5	Is SAM Type Supported?	SAM Decoder, Unsupported

Table 5

A.2.3. DRIP SAM

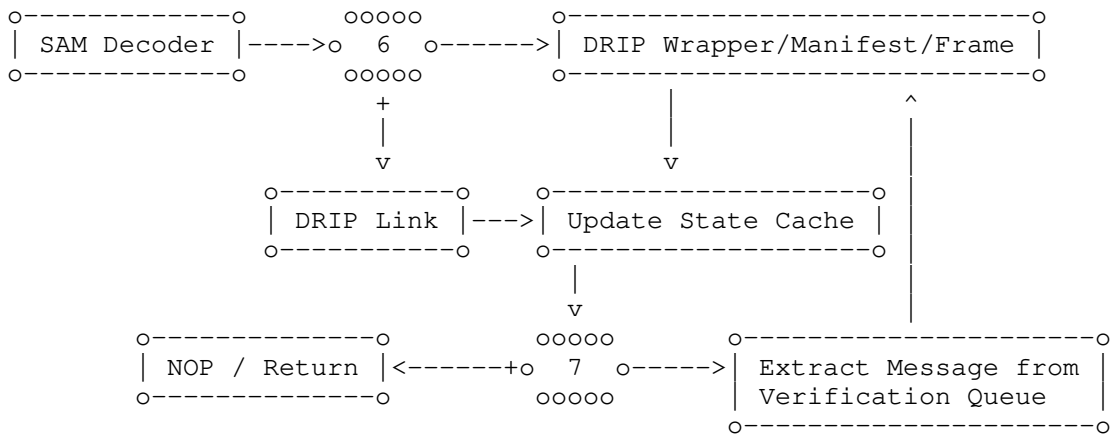


Figure 13: DRIP SAM Decoder

Transition	Transition Query	Next State/Process/ Transition (Yes, No)
6	Is SAM Type DRIP Link?	DRIP Link, DRIP Wrapper/Manifest/Frame
7	Messages in Verification Queue?	Extract Message from Verification Queue, NOP / Return

Table 6

A.2.4. DRIP Link

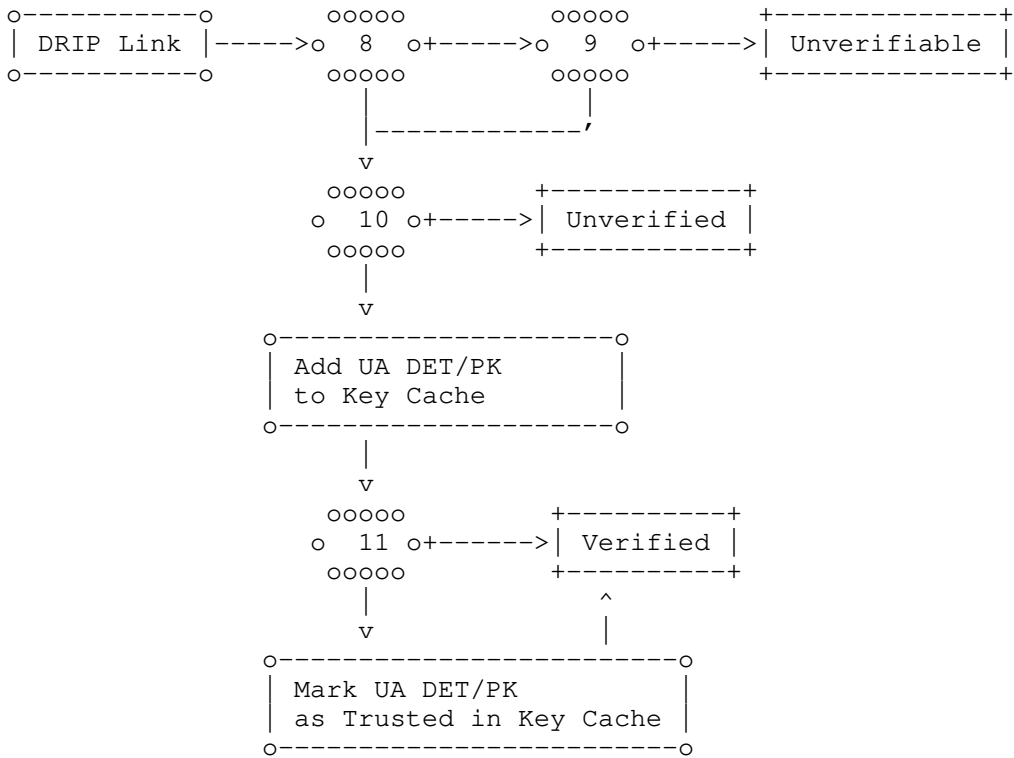


Figure 14: DRIP Link State Decoder

Transition	Transition Query	Next State/Process/ Transition (Yes, No)
8	Registry DET/PK in Key Cache?	10, 9
9	Registry PK found Online?	10, Unverifiable
10	Registry Signature Verified?	Add UA DET/PK to Key Cache, Unverified
11	Registry DET/PK marked as Trusted in Key Cache?	Mark UA DET/PK as Trusted in Key Cache, Verified

Table 7

## A.2.5. DRIP Wrapper/Manifest/Frame

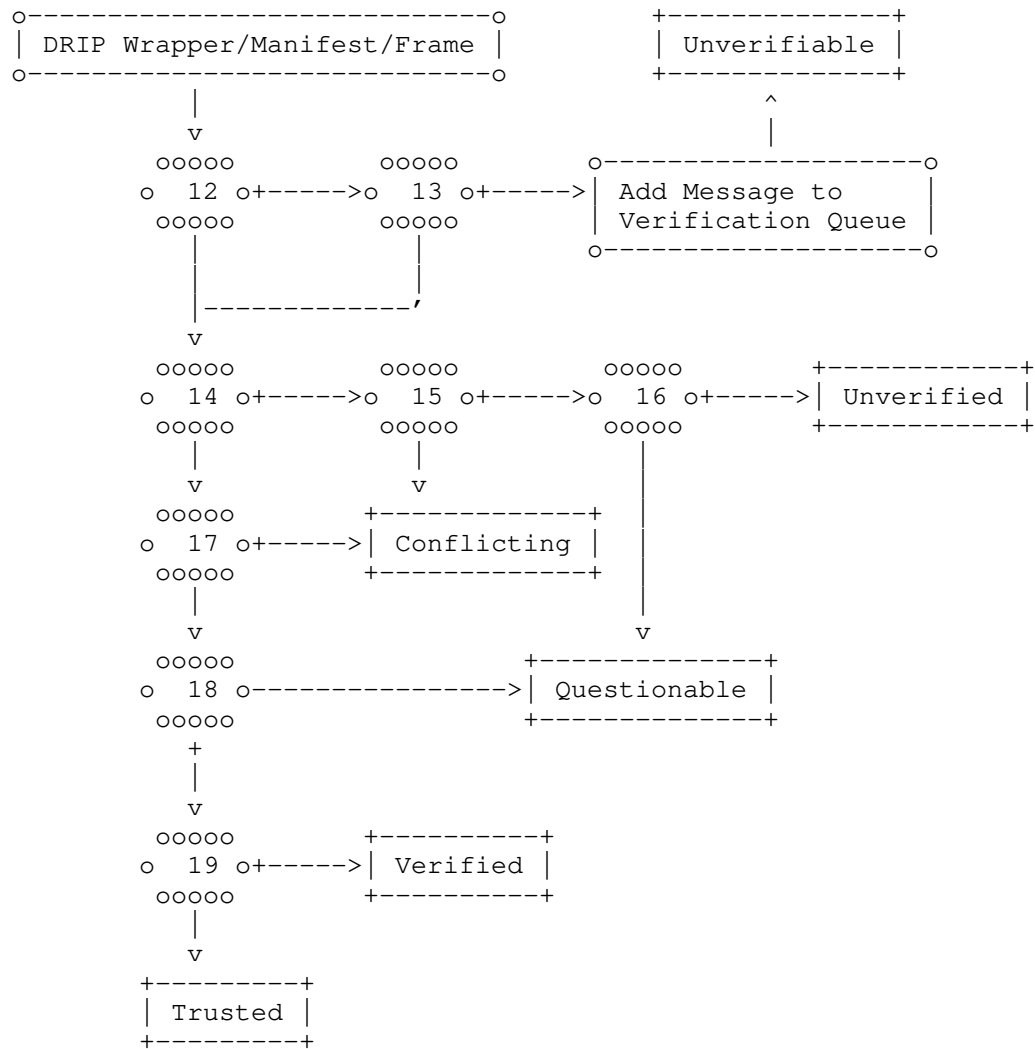


Figure 15: DRIP Wrapper/Manifest/Frame State Decoder

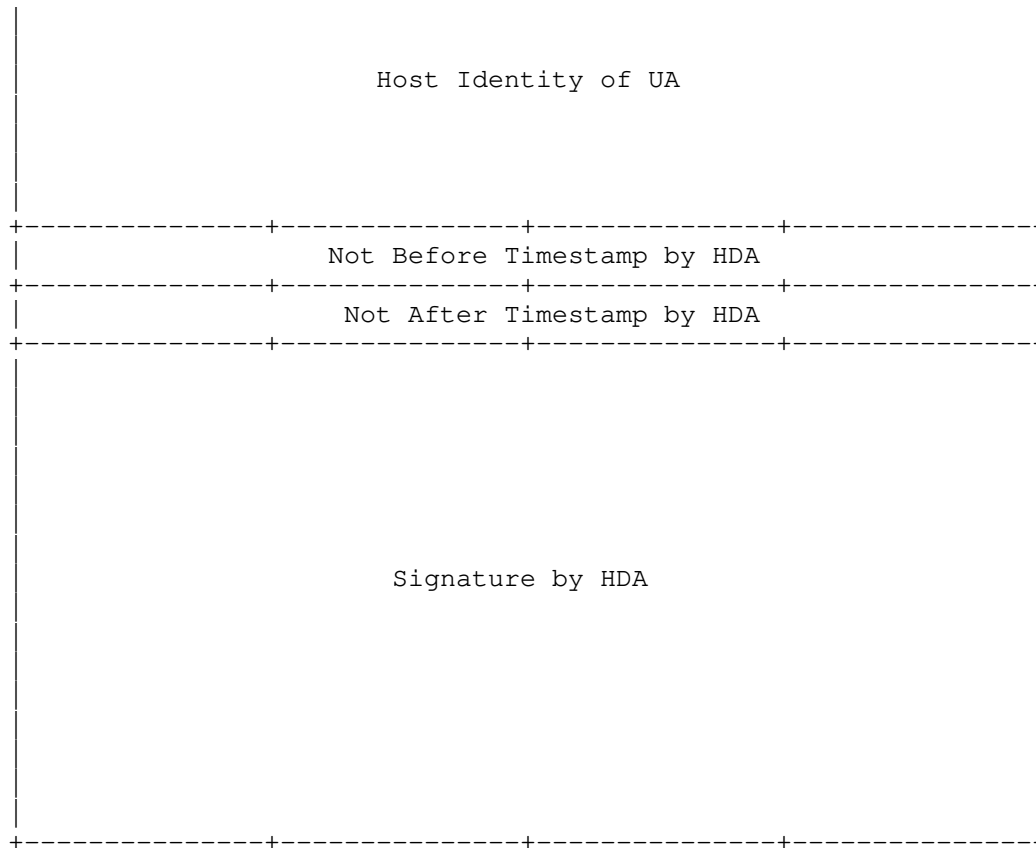


Transition	Transition Query	Next State/Process/ Transition (Yes, No)
12	UA DET/PK in Key Cache?	14, 13
13	UA PK found Online?	14, Add Message to Verification Queue
14	UA Signature Verified?	17, 15
15	Has past Messages of this type been marked as Trusted?	Conflicting, 16
16	Has past Messages of this type been marked as Questionable or Verified?	Questionable, Unverified
17	Has past Messages of this type been marked as Conflicting?	Conflicting, 18
18	Has past Messages of this type been marked as Questionable or Unverified?	Questionable, 19
19	Is UA DET/PK marked as Trusted in Key Cache?	Trusted, Verified

Table 8

## Appendix B. HDA-UA Broadcast Attestation

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1			
DRIP Entity Tag of HDA			
DRIP Entity Tag of UA			



DRIP Entity Tag of HDA: (16-bytes)  
DET of HDA.

DRIP Entity Tag of UA: (16-bytes)  
DET of UA.

Host Identity of UA: (32-bytes)  
HI of UA

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

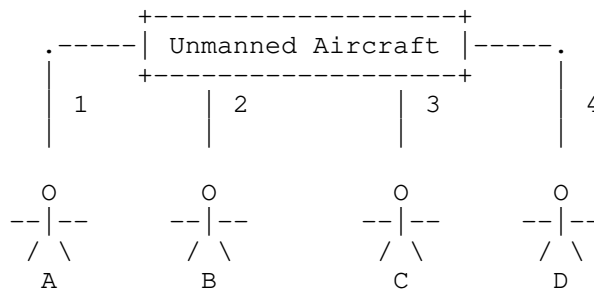
Signing Timestamp by HDA (4 bytes):  
Current time at signing.

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

Figure 16: Example DRIP HDA-UA Broadcast Attestation

## Appendix C. Example TX/RX Flow

In this example the UA is sending all DRIP Authentication Message formats (DRIP Link, DRIP Wrapper and DRIP Manifest) during flight, along with standard ASTM Messages. The objective is to show the combinations of messages that must be received to properly validate a DRIP equipped UA and examples of their various states (Appendix A).



### Broadcast Paths: Messages Received

- 1: DRIP Link
- 2: DRIP Link and DRIP Wrapper or DRIP Manifest
- 3: DRIP Wrapper or DRIP Manifest
- 4: None

## Observers: Authentication State

- A: Unverifiable  
B: Verified, Trusted, Unverified, Questionable, or Conflicting  
C: Unverifiable  
D: None

As the above example shows to properly authenticate both a DRIP Link and a DRIP Wrapper or DRIP Manifest are required.

## 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](mailto:stu.card@axenterprize.com)

Robert Moskowitz  
HTT Consulting  
Oak Park, MI 48237  
United States of America  
Email: [rgm@labs.htt-consult.com](mailto:rgm@labs.htt-consult.com)

DRIP  
Internet-Draft  
Updates: 7401, 7343 (if approved)  
Intended status: Standards Track  
Expires: 14 November 2022

R. Moskowitz  
HTT Consulting  
S. Card  
A. Wiethuechter  
AX Enterprize, LLC  
A. Gurtov  
Linköping University  
13 May 2022

DRIP Entity Tag (DET) for Unmanned Aircraft System Remote ID (UAS RID)  
draft-ietf-drip-rid-26

## Abstract

This document describes the use of Hierarchical Host Identity Tags (HHITs) as self-asserting IPv6 addresses and thereby a trustable identifier for use as the Unmanned Aircraft System Remote Identification and tracking (UAS RID).

This document updates RFC7401 and RFC7343.

Within the context of RID, HHITs will be called DRIP Entity Tags (DETs). HHITs self-attest to the included explicit hierarchy that provides registry (via, e.g., DNS, EPP) discovery for 3rd-party identifier attestation.

## 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 14 November 2022.

## Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. HHIT Statistical Uniqueness different from UUID or X.509 Subject . . . . .	4
2. Terms and Definitions . . . . .	4
2.1. Requirements Terminology . . . . .	4
2.2. Notations . . . . .	4
2.3. Definitions . . . . .	4
3. The Hierarchical Host Identity Tag (HHIT) . . . . .	6
3.1. HHIT Prefix for RID Purposes . . . . .	7
3.2. HHIT Suite IDs . . . . .	7
3.2.1. HDA custom HIT Suite IDs . . . . .	8
3.3. The Hierarchy ID (HID) . . . . .	8
3.3.1. The Registered Assigning Authority (RAA) . . . . .	8
3.3.2. The Hierarchical HIT Domain Authority (HDA) . . . . .	9
3.4. Edward-Curve Digital Signature Algorithm for HHITs . . . . .	9
3.4.1. HOST_ID . . . . .	10
3.4.2. HIT_SUITE_LIST . . . . .	11
3.5. ORCHIDs for Hierarchical HITs . . . . .	11
3.5.1. Adding Additional Information to the ORCHID . . . . .	12
3.5.2. ORCHID Encoding . . . . .	13
3.5.3. ORCHID Decoding . . . . .	15
3.5.4. Decoding ORCHIDs for HIPv2 . . . . .	15
4. Hierarchical HITs as DRIP Entity Tags . . . . .	15
4.1. Nontransferability of DETs . . . . .	16
4.2. Encoding HHITs in CTA 2063-A Serial Numbers . . . . .	16
4.3. Remote ID DET as one Class of Hierarchical HITs . . . . .	17
4.4. Hierarchy in ORCHID Generation . . . . .	17
4.5. DRIP Entity Tag (DET) Registry . . . . .	18
4.6. Remote ID Authentication using DETs . . . . .	18
5. DRIP Entity Tags (DETs) in DNS . . . . .	18
6. Other UTM Uses of HHITs Beyond DET . . . . .	20
7. Summary of Addressed DRIP Requirements . . . . .	20
8. IANA Considerations . . . . .	20
8.1. New Well-Known IPv6 prefix for DETs . . . . .	20
8.2. New IANA DRIP Registry . . . . .	21
8.3. IANA CGA Registry Update . . . . .	22
8.4. IANA HIP Registry Updates . . . . .	22

8.5.	IANA IPSECKEY Registry Update . . . . .	23
9.	Security Considerations . . . . .	23
9.1.	DET Trust in ASTM messaging . . . . .	25
9.2.	DET Revocation . . . . .	25
9.3.	Privacy Considerations . . . . .	26
9.4.	Collision Risks with DETs . . . . .	27
10.	References . . . . .	27
10.1.	Normative References . . . . .	27
10.2.	Informative References . . . . .	28
Appendix A.	EU U-Space RID Privacy Considerations . . . . .	31
Appendix B.	The 14/14 HID split . . . . .	31
Appendix C.	Calculating Collision Probabilities . . . . .	33
Acknowledgments	. . . . .	33
Authors' Addresses	. . . . .	34

## 1. Introduction

DRIP Requirements [RFC9153] describe an Unmanned Aircraft System Remote ID (UAS ID) as unique (ID-4), non-spoofable (ID-5), and identify a registry where the ID is listed (ID-2); all within a 19-character identifier (ID-1).

This document describes (per Section 3 of [drip-architecture]) the use of Hierarchical Host Identity Tags (HHITs) (Section 3) as self-asserting IPv6 addresses and thereby a trustable identifier for use as the UAS Remote ID. HHITs add explicit hierarchy to the 128-bit HITs, enabling DNS HHIT queries (Host ID for authentication, e.g., [drip-authentication]) and for Extensible Provisioning Protocol (EPP) Registrar discovery [RFC9224] for 3rd-party identification attestation (e.g., [drip-authentication]).

This addition of hierarchy to HITs is an extension to [RFC7401] and requires an update to [RFC7343]. As this document also adds EdDSA (Section 3.4) for Host Identities (HIs), a number of Host Identity Protocol (HIP) parameters in [RFC7401] are updated, but these should not be needed in a DRIP implementation that does not use HIP.

HHITs as used within the context of Unmanned Aircraft System (UAS) are labeled as DRIP Entity Tags (DETs). Throughout this document HHIT and DET will be used appropriately. HHIT will be used when covering the technology, and DET for their context within UAS RID.

Hierarchical HITs provide self-attestation of the HHIT registry. A HHIT can only be in a single registry within a registry system (e.g., EPP and DNS).

Hierarchical HITs are valid, though non-routable, IPv6 addresses [RFC8200]. As such, they fit in many ways within various IETF technologies.

### 1.1. HHIT Statistical Uniqueness different from UUID or X.509 Subject

HHITs are statistically unique through the cryptographic hash feature of second-preimage resistance. The cryptographically-bound addition of the hierarchy and a HHIT registration process [drip-registries] provide complete, global HHIT uniqueness. This contrasts with using general identifiers (e.g., a Universally Unique IDentifiers (UUID) [RFC4122] or device serial numbers) as the subject in an X.509 [RFC5280] certificate.

In a multi-Certificate Authority (multi-CA) PKI alternative to HHITs, a Remote ID as the Subject (Section 4.1.2.6 of [RFC5280]) can occur in multiple CAs, possibly fraudulently. CAs within the PKI would need to implement an approach to enforce assurance of the uniqueness achieved with HHITs.

## 2. Terms and Definitions

### 2.1. Requirements 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. Notations

| Signifies concatenation of information - e.g., X | Y is the concatenation of X and Y.

### 2.3. Definitions

This document uses the terms defined in Section 2.2 of [RFC9153]. The following new terms are used in the document:

cSHAKE (The customizable SHAKE function [NIST.SP.800-185]):  
Extends the SHAKE [NIST.FIPS.202] scheme to allow users to customize their use of the SHAKE function.

HDA (HHIT Domain Authority):  
The 14-bit field that identifies the HHIT Domain Authority under a Registered Assigning Authority (RAA).



**HHIT**

Hierarchical Host Identity Tag. A HIT with extra hierarchical information not found in a standard HIT [RFC7401].

**HI**

Host Identity. The public key portion of an asymmetric key pair as defined in [RFC9063].

**HID (Hierarchy ID):**

The 28-bit field providing the HIT Hierarchy ID.

**HIP (Host Identity Protocol)**

The origin [RFC7401] of HI, HIT, and HHIT.

**HIT**

Host Identity Tag. A 128-bit handle on the HI. HITs are valid IPv6 addresses.

**Keccak (KECCAK Message Authentication Code):**

The family of all sponge functions with a KECCAK-f permutation as the underlying function and multi-rate padding as the padding rule. It refers in particular to all the functions referenced from [NIST.FIPS.202] and [NIST.SP.800-185].

**KMAC (KECCAK Message Authentication Code [NIST.SP.800-185]):**

A Pseudo Random Function (PRF) and keyed hash function based on KECCAK.

**RAA (Registered Assigning Authority):**

The 14-bit field identifying the business or organization that manages a registry of HDAs.

**RVS (Rendezvous Server):**

A Rendezvous Server such as the HIP Rendezvous Server for enabling mobility, as defined in [RFC8004].

**SHAKE (Secure Hash Algorithm KECCAK [NIST.FIPS.202]):**

A secure hash that allows for an arbitrary output length.

**XOF (eXtendable-Output Function [NIST.FIPS.202]):**

A function on bit strings (also called messages) in which the output can be extended to any desired length.

### 3. The Hierarchical Host Identity Tag (HHIT)

The Hierarchical HIT (HHIT) is a small but important enhancement over the flat Host Identity Tag (HIT) space, constructed as an Overlay Routable Cryptographic Hash Identifier (ORCHID) [RFC7343]. By adding two levels of hierarchical administration control, the HHIT provides for device registration/ownership, thereby enhancing the trust framework for HITs.

The 128-bit HHITs represent the HI in only a 64-bit hash, rather than the 96 bits in HITs. 4 of these 32 freed up bits expand the Suite ID to 8 bits, and the other 28 bits are used to create a hierarchical administration organization for HIT domains. Hierarchical HIT construction is defined in Section 3.5. The input values for the Encoding rules are described in Section 3.5.1.

A HHIT is built from the following fields (Figure 1):

- \* p = an IPV6 prefix (max 28 bit)
- \* 28-bit Hierarchy ID (HID) which provides the structure to organize HITs into administrative domains. HIDs are further divided into two fields:
  - 14-bit Registered Assigning Authority (RAA) (Section 3.3.1)
  - 14-bit Hierarchical HIT Domain Authority (HDA) (Section 3.3.2)
- \* 8-bit HHIT Suite ID (HHSI)
- \* ORCHID hash (96 - prefix length - 8 for HHIT Suite ID, e.g., 64)  
See Section 3.5 for more details.

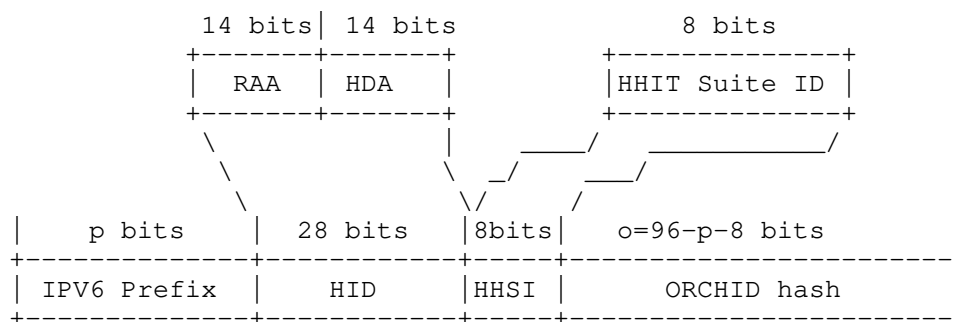


Figure 1: HHIT Format

The Context ID (generated with openssl rand) for the ORCHID hash is:

Context ID := 0x00B5 A69C 795D F5D5 F008 7F56 843F 2C40

Context IDs are allocated out of the namespace introduced for Cryptographically Generated Addresses (CGA) Type Tags [RFC3972].

### 3.1. HHIT Prefix for RID Purposes

The IPv6 HHIT prefix MUST be distinct from that used in the flat-space HIT as allocated in [RFC7343]. Without this distinct prefix, the first 4 bits of the RAA would be interpreted as the HIT Suite ID per HIPv2 [RFC7401].

Initially, for DET use, one 28-bit prefix should be assigned out of the IANA IPv6 Special Purpose Address Block ([RFC6890]).

HHIT Use	Bits	Value
DET	28	TBD6 (suggested value 2001:30::/28)

Other prefixes may be added in the future either for DET use or other applications of HHITs. For a prefix to be added to the registry in Section 8.2, its usage and HID allocation process have to be publicly available.

### 3.2. HHIT Suite IDs

The HHIT Suite IDs specify the HI and hash algorithms. These are a superset of the 4/8-bit HIT Suite ID as defined in Section 5.2.10 of [RFC7401].

The HHIT values of 1 - 15 map to the basic 4-bit HIT Suite IDs. HHIT values of 17 - 31 map to the extended 8-bit HIT Suite IDs. HHIT values unique to HHIT will start with value 32.

As HHIT introduces a new Suite ID, EdDSA/cSHAKE128, and since this is of value to HIPv2, it will be allocated out of the 4-bit HIT space and result in an update to HIT Suite IDs. Future HHIT Suite IDs may be allocated similarly, or may come out of the additional space made available by going to 8 bits.

The following HHIT Suite IDs are defined:

HHIT Suite	Value
RESERVED	0
RSA,DSA/SHA-256	1 [RFC7401]
ECDSA/SHA-384	2 [RFC7401]
ECDSA_LOW/SHA-1	3 [RFC7401]
EdDSA/cSHAKE128	TBD3 (suggested value 5) (RECOMMENDED)

### 3.2.1. HDA custom HIT Suite IDs

Support for 8-bit HHIT Suite IDs allows for HDA custom HIT Suite IDs. These will be assigned values greater than 15 as follows:

HHIT Suite	Value
HDA Private Use 1	TBD4 (suggested value 254)
HDA Private Use 2	TBD5 (suggested value 255)

These custom HIT Suite IDs, for example, may be used for large-scale experimenting with post quantum computing hashes or similar domain specific needs. Note that currently there is no support for domain-specific HI algorithms.

They should not be used to create a "de facto standardization". Section 8.2 states that additional Suite IDs can be made through IETF Review.

### 3.3. The Hierarchy ID (HID)

The Hierarchy ID (HID) provides the structure to organize HITs into administrative domains. HIDs are further divided into two fields:

- \* 14-bit Registered Assigning Authority (RAA)
- \* 14-bit Hierarchical HIT Domain Authority (HDA)

The rationale for the 14/14 HID split is described in Appendix B.

The two levels of hierarchy allows for CAAs to have at least one RAA for their National Air Space (NAS). Within its RAA(s), the CAAs can delegate HDAs as needed. There may be other RAAs allowed to operate within a given NAS; this is a policy decision of each CAA.

#### 3.3.1. The Registered Assigning Authority (RAA)

An RAA is a business or organization that manages a registry of HDAs. For example, the Federal Aviation Authority (FAA) or Japan Civil Aviation Bureau (JCAB) could be an RAA.

The RAA is a 14-bit field (16,384 RAAs). The management of this space is further elaborated in [drip-registries]. An RAA MUST provide a set of services to allocate HDAs to organizations. It SHOULD have a public policy on what is necessary to obtain an HDA. The RAA need not maintain any HIP related services. It MUST maintain a DNS zone minimally for discovering HIP RVS servers for the HID. The zone delegation is also covered in [drip-registries].

As DETs under an administrative control may be used in many different domains (e.g., commercial, recreation, military), RAAs should be allocated in blocks (e.g. 16-19) with consideration on the likely size of a particular usage. Alternatively, different prefixes can be used to separate different domains of use of HHITs.

The RAA DNS zone within the UAS DNS tree may be a PTR for its RAA. It may be a zone in an HHIT specific DNS zone. Assume that the RAA is decimal 100. The PTR record could be constructed as follows:

```
100.hhit.arpa    IN PTR      raa.example.com.
```

Note that if the zone `hhit.arpa` is ultimately used, some registrar will need to manage this for all HHIT applications. Thus further thought will be needed in the actual zone tree and registration process [drip-registries].

### 3.3.2. The Hierarchical HIT Domain Authority (HDA)

An HDA may be an Internet Service Provider (ISP), UAS Service Supplier (USS), or any third party that takes on the business to provide UAS services management, HIP RVSSs or other needed services such as those required for HHIT and/or HIP-enabled devices.

The HDA is a 14-bit field (16,384 HDAs per RAA) assigned by an RAA is further elaborated in [drip-registries]. An HDA must maintain public and private UAS registration information and should maintain a set of RVS servers for UAS clients that may use HIP. How this is done and scales to the potentially millions of customers are outside the scope of this document, though covered in [drip-registries]. This service should be discoverable through the DNS zone maintained by the HDA's RAA.

An RAA may assign a block of values to an individual organization. This is completely up to the individual RAA's published policy for delegation. Such policy is out of scope.

### 3.4. Edwards-Curve Digital Signature Algorithm for HHITs

The Edwards-Curve Digital Signature Algorithm (EdDSA) [RFC8032] is specified here for use as HIs per HIPv2 [RFC7401].

The intent in this document is to add EdDSA as a HI algorithm for DETs, but doing so impacts the HIP parameters used in a HIP exchange. The subsections of this section document the required updates of HIP parameters. Other than the HIP DNS RR (Resource Record), these should not be needed in a DRIP implementation that does not use HIP.

See Section 3.2 for use of the HIT Suite in the context of DRIP.

### 3.4.1. HOST\_ID

The HOST\_ID parameter specifies the public key algorithm, and for elliptic curves, a name. The HOST\_ID parameter is defined in Section 5.2.9 of [RFC7401].

Algorithm profiles	Values
EdDSA	TBD1 (suggested value 13) [RFC8032] (RECOMMENDED)

#### 3.4.1.1. HIP Parameter support for EdDSA

The addition of EdDSA as a HI algorithm requires a subfield in the HIP HOST\_ID parameter (Section 5.2.9 of [RFC7401]) as was done for ECDSA when used in a HIP exchange.

For HIP hosts that implement EdDSA as the algorithm, the following EdDSA curves are represented by the following fields:

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1			
EdDSA Curve		NULL	/
/	Public Key		

EdDSA Curve	Curve label
Public Key	Represented in Octet-string format [RFC8032]

Figure 2

For hosts that implement EdDSA as a HIP algorithm the following EdDSA curves are required:

Algorithm	Curve	Values
EdDSA	RESERVED	0
EdDSA	EdDSA25519	1 [RFC8032] (RECOMMENDED)
EdDSA	EdDSA25519ph	2 [RFC8032]
EdDSA	EdDSA448	3 [RFC8032] (RECOMMENDED)
EdDSA	EdDSA448ph	4 [RFC8032]

### 3.4.1.2. HIP DNS RR support for EdDSA

The HIP DNS RR is defined in [RFC8005]. It uses the values defined for the 'Algorithm Type' of the IPSECKEY RR [RFC4025] for its PK Algorithm field.

The new EdDSA HI uses [RFC8080] for the IPSECKEY RR encoding:

Value	Description
-------	-------------

TBD2 (suggested value 4)	
--------------------------	--

An EdDSA key is present, in the format defined in [RFC8080]	
---	--

### 3.4.2. HIT\_SUITE\_LIST

The HIT\_SUITE\_LIST parameter contains a list of the supported HIT suite IDs of the HIP Responder. Based on the HIT\_SUITE\_LIST, the HIP Initiator can determine which source HIT Suite IDs are supported by the Responder. The HIT\_SUITE\_LIST parameter is defined in Section 5.2.10 of [RFC7401].

The following HIT Suite ID is defined:

HIT Suite	Value	
EdDSA/cSHAKE128	TBD3 (suggested value 5)	(RECOMMENDED)

Table 1 provides more detail on the above HIT Suite combination.

The output of cSHAKE128 is variable per the needs of a specific ORCHID construction. It is at most 96 bits long and is directly used in the ORCHID (without truncation).

Index	Hash function	HMAC	Signature algorithm family	Description
5	cSHAKE128	KMAC128	EdDSA	EdDSA HI hashed with cSHAKE128, output is variable

Table 1: HIT Suites

### 3.5. ORCHIDs for Hierarchical HITs

This section improves on ORCHIDv2 [RFC7343] with three enhancements:

- \* Optional "Info" field between the Prefix and OGA ID.
- \* Increased flexibility on the length of each component in the ORCHID construction, provided the resulting ORCHID is 128 bits.
- \* Use of cSHAKE, NIST SP 800-185 [NIST.SP.800-185], for the hashing function.

The Keccak [Keccak] based cSHAKE XOF hash function is a variable output length hash function. As such it does not use the truncation operation that other hashes need. The invocation of cSHAKE specifies the desired number of bits in the hash output. Further, cSHAKE has a parameter 'S' as a customization bit string. This parameter will be used for including the ORCHID Context Identifier in a standard fashion.

This ORCHID construction includes the fields in the ORCHID in the hash to protect them against substitution attacks. It also provides for inclusion of additional information, in particular the hierarchical bits of the Hierarchical HIT, in the ORCHID generation. This should be viewed as an update to ORCHIDv2 [RFC7343], as it can produce ORCHIDv2 output.

#### 3.5.1. Adding Additional Information to the ORCHID

ORCHIDv2 [RFC7343] is defined as consisting of three components:

ORCHID := Prefix | OGA ID | Encode\_96( Hash )

where:

Prefix : A constant 28-bit-long bitstring value (IPv6 prefix)

OGA ID : A 4-bit long identifier for the Hash\_function in use within the specific usage context. When used for HIT generation this is the HIT Suite ID.

Encode\_96( ) : An extraction function in which output is obtained by extracting the middle 96-bit-long bitstring from the argument bitstring.

The new ORCHID function is as follows:



ORCHID := Prefix (p) | Info (n) | OGA ID (o) | Hash (m)

where:

Prefix (p) : An IPv6 prefix of length p (max 28-bit-long).

Info (n) : n bits of information that define a use of the ORCHID. 'n' can be zero, that is no additional information.

OGA ID (o) : A 4- or 8-bit long identifier for the Hash\_function in use within the specific usage context. When used for HIT generation this is the HIT Suite ID. When used for HHIT generation this is the HHIT Suite ID.

Hash (m) : An extraction function in which output is 'm' bits.

$p + n + o + m = 128$  bits

The ORCHID length MUST be 128 bits. With a 28-bit IPv6 prefix, the remaining 100 bits can be divided in any manner between the additional information ("Info"), OGA ID, and the hash output. Care must be considering the size of the hash portion, taking into account risks like pre-image attacks. 64 bits, as used in Hierarchical HITs may be as small as is acceptable. The size of 'n' is determined as what is left; in the case of the 8-bit OGA used for HHIT, this is 28 bits.

### 3.5.2. ORCHID Encoding

This update adds a different encoding process to that currently used in ORCHIDv2. The input to the hash function explicitly includes all the header content plus the Context ID. The header content consists of the Prefix, the Additional Information ("Info"), and OGA ID (HIT Suite ID). Secondly, the length of the resulting hash is set by sum of the length of the ORCHID header fields. For example, a 28-bit prefix with 28 bits for the HID and 8 bits for the OGA ID leaves 64 bits for the hash length.

To achieve the variable length output in a consistent manner, the cSHAKE hash is used. For this purpose, cSHAKE128 is appropriate. The cSHAKE function call for this update is:

cSHAKE128(Input, L, "", Context ID)

Input := Prefix | Additional Information | OGA ID | HOST\_ID  
L := Length in bits of hash portion of ORCHID

For full Suite ID support (those that use fixed length hashes like SHA256), the following hashing can be used (Note: this does not produce output Identical to ORCHIDv2 for a /28 prefix and Additional Information of zero-length):

```

Hash[L] (Context ID | Input)

Input      := Prefix | Additional Information | OGA ID | HOST_ID
L          := Length in bits of hash portion of ORCHID

Hash[L]    := An extraction function in which output is obtained
               by extracting the middle L-bit-long bitstring
               from the argument bitstring.

```

Hierarchical HITs use the Context ID defined in Section 3.

#### 3.5.2.1. Encoding ORCHIDs for HIPv2

This section discusses how to provide backwards compatibility for ORCHIDv2 [RFC7343] as used in HIPv2 [RFC7401].

For HIPv2, the Prefix is 2001:20::/28 (Section 6 of [RFC7343]). 'Info' is zero-length (i.e., not included), and OGA ID is 4-bit. Thus, the HI Hash is 96-bit length. Further, the Prefix and OGA ID are not included in the hash calculation. Thus, the following ORCHID calculations for fixed output length hashes are used:

```

Hash[L] (Context ID | Input)

Input      := HOST_ID
L          := 96
Context ID := 0xF0EF F02F BFF4 3D0F E793 0C3C 6E61 74EA

Hash[L]    := An extraction function in which output is obtained
               by extracting the middle L-bit-long bitstring
               from the argument bitstring.

```

For variable output length hashes use:

```

Hash[L] (Context ID | Input)

Input      := HOST_ID
L          := 96
Context ID := 0xF0EF F02F BFF4 3D0F E793 0C3C 6E61 74EA

Hash[L]    := The L-bit output from the hash function

```

Then, the ORCHID is constructed as follows:

Prefix | OGA ID | Hash Output

### 3.5.3. ORCHID Decoding

With this update, the decoding of an ORCHID is determined by the Prefix and OGA ID. ORCHIDv2 [RFC7343] decoding is selected when the Prefix is: 2001:20::/28.

For Hierarchical HITs, the decoding is determined by the presence of the HHIT Prefix as specified in Section 8.2.

### 3.5.4. Decoding ORCHIDs for HIPv2

This section is included to provide backwards compatibility for ORCHIDv2 [RFC7343] as used for HIPv2 [RFC7401].

HITs are identified by a Prefix of 2001:20::/28. The next 4 bits are the OGA ID. The remaining 96 bits are the HI Hash.

## 4. Hierarchical HITs as DRIP Entity Tags

HHITs for UAS ID (called, DETs) use the new EdDSA/SHAKE128 HIT suite defined in Section 3.4 (GEN-2 in [RFC9153]). This hierarchy, cryptographically bound within the HHIT, provides the information for finding the UA's HHIT registry (ID-3 in [RFC9153]).

The 2022 forthcoming updated release of ASTM Standard Specification for Remote ID and Tracking [F3411] adds support for DETs. This is within the UAS ID type 4, "Specific Session ID (SSI)".

Note to RFC Editor: This, and all references to F3411 need to be updated to this new version which is in final ASTM editing. A new link and replacement text will be provided when it is published.

The original UAS ID Types 1 - 3 allow for an UAS ID with a maximum length of 20 bytes, this new SSI (Type 4) uses the first byte of the ID for the SSI Type, thus restricting the UAS ID of this type to a maximum of 19 bytes. The SSI Types initially assigned are:

ID 1 IETF - DRIP Drone Remote ID Protocol (DRIP) entity ID.

ID 2 3GPP - IEEE 1609.2-2016 HashedID8

#### 4.1. Nontransferability of DETs

A HI and its HHIT SHOULD NOT be transferable between UA or even between replacement electronics (e.g., replacement of damaged controller CPU) for a UA. The private key for the HI SHOULD be held in a cryptographically secure component.

#### 4.2. Encoding HHITs in CTA 2063-A Serial Numbers

In some cases, it is advantageous to encode HHITs as a CTA 2063-A Serial Number [CTA2063A]. For example, the FAA Remote ID Rules [FAA\_RID] state that a Remote ID Module (i.e., not integrated with UA controller) must only use "the serial number of the unmanned aircraft"; CTA 2063-A meets this requirement.

Encoding an HHIT within the CTA 2063-A format is not simple. The CTA 2063-A format is defined as follows:

Serial Number := MFR Code | Length Code | MFR SN

where:

MFR Code : 4 character code assigned by ICAO  
(International Civil Aviation Organization,  
a UN Agency).

Length Code : 1 character Hex encoding of MFR SN length (1-F).

MFR SN : Alphanumeric code (0-9, A-Z except O and I).  
Maximum length of 15 characters.

There is no place for the HID; there will need to be a mapping service from Manufacturer Code to HID. The HHIT Suite ID and ORCHID hash will take the full 15 characters (as described below) of the MFR SN field.

A character in a CTA 2063-A Serial Number "shall include any combination of digits and uppercase letters, except the letters O and I, but may include all digits". This would allow for a Base34 encoding of the binary HHIT Suite ID and ORCHID hash in 15 characters. Although, programmatically, such a conversion is not hard, other technologies (e.g., credit card payment systems) that have used such odd base encoding have had performance challenges. Thus, here a Base32 encoding will be used by also excluding the letters Z and S (too similar to the digits 2 and 5).

The low-order 72 bits (HHIT Suite ID | ORCHID hash) of the HHIT SHALL be left-padded with 3 bits of zeros. This 75-bit number will be encoded into the 15-character MFR SN field using the digit/letters above. The manufacturer MUST use a Length Code of F (15).

Using the sample DET from Section 5 that is for HDA=20 under RAA=10 and having the ICAO CTA MFR Code of 8653, the 20-character CTA 2063-A Serial Number would be:

8653F02T7B8RA85D19LX

A mapping service (e.g., DNS) MUST provide a trusted (e.g., via DNSSEC [RFC4034]) conversion of the 4-character Manufacturer Code to high-order 58 bits (Prefix | HID) of the HHIT. Definition of this mapping service is currently out of scope of this document.

It should be noted that this encoding would only be used in the Basic ID Message (Section 2.2 of [RFC9153]). The DET is used in the Authentication Messages (i.e., the messages that provide framing for authentication data only).

#### 4.3. Remote ID DET as one Class of Hierarchical HITs

UAS Remote ID DET may be one of a number of uses of HHITs. However, it is out of the scope of the document to elaborate on other uses of HHITs. As such these follow-on uses need to be considered in allocating the RAAs (Section 3.3.1) or HHIT prefix assignments (Section 8).

#### 4.4. Hierarchy in ORCHID Generation

ORCHIDS, as defined in [RFC7343], do not cryptographically bind an IPv6 prefix nor the ORCHID Generation Algorithm (OGA) ID (the HIT Suite ID) to the hash of the HI. The rationale at the time of developing ORCHID was attacks against these fields are Denial-of-Service (DoS) attacks against protocols using ORCHIDs and thus up to those protocols to address the issue.

HHITs, as defined in Section 3.5, cryptographically bind all content in the ORCHID through the hashing function. A recipient of a DET that has the underlying HI can directly trust and act on all content in the HHIT. This provides a strong, self-attestation for using the hierarchy to find the DET Registry based on the HID (Section 4.5).

#### 4.5. DRIP Entity Tag (DET) Registry

DETs are registered to HDAs. A registration process, [drip-registries], ensures DET global uniqueness (ID-4 in [RFC9153]). It also provides the mechanism to create UAS public/private data that are associated with the DET (REG-1 and REG-2 in [RFC9153]).

#### 4.6. Remote ID Authentication using DETs

The EdDSA25519 HI (Section 3.4) underlying the DET can be used in an 84-byte self-proof attestation (timestamp, HHIT, and signature of these) to provide proof of Remote ID ownership (GEN-1 in [RFC9153]). In practice, the Wrapper and Manifest authentication formats (Sections 6.3.3 and 6.3.4 of [drip-authentication]) implicitly provide this self-attestation. A lookup service like DNS can provide the HI and registration proof (GEN-3 in [RFC9153]).

Similarly, for Observers without Internet access, a 200-byte offline self-attestation could provide the same Remote ID ownership proof. This attestation would contain the HDA's signing of the UA's HHIT, itself signed by the UA's HI. Only a small cache that contains the HDA's HI/HHIT and HDA meta-data is needed by the Observer. However, such an object would just fit in the ASTM Authentication Message (Section 2.2 of [RFC9153]) with no room for growth. In practice, [drip-authentication] provides this offline self-attestation in two authentication messages: the HDA's certification of the UA's HHIT registration in a Link authentication message whose hash is sent in a Manifest authentication message.

Hashes of any previously sent ASTM messages can be placed in a Manifest authentication message (GEN-2 in [RFC9153]). When a Location/Vector Message (i.e., a message that provides UA location, altitude, heading, speed, and status) hash along with the hash of the HDA's UA HHIT attestation are sent in a Manifest authentication message and the Observer can visually see a UA at the claimed location, the Observer has a very strong proof of the UA's Remote ID.

All this behavior and how to mix these authentication messages into the flow of UA operation messages are detailed in [drip-authentication].

#### 5. DRIP Entity Tags (DETs) in DNS

There are two approaches for storing and retrieving DETs using DNS. The following are examples of how this may be done. This will serve as guidance to the actual deployment of DETs in DNS. However, this document does not intend to provide a recommendation. Further DNS-related considerations are covered in [drip-registries].

\* As FQDNs, for example, ".icao.int".

\* Reverse DNS lookups as IPv6 addresses per [RFC8005].

A DET can be used to construct an FQDN that points to the USS that has the public/private information for the UA (REG-1 and REG-2 in [RFC9153]). For example, the USS for the HHIT could be found via the following: assume the RAA is decimal 100 and the HDA is decimal 50. The PTR record is constructed as follows:

```
100.50.det.uas.icao.int    IN PTR      foo.uss.icao.int.
```

The individual DETs may be potentially too numerous (e.g., 60 - 600M) and dynamic (e.g., new DETs every minute for some HDAs) to store in a signed, DNS zone. The HDA SHOULD provide DNS service for its zone and provide the HHIT detail response.

The DET reverse lookup can be a standard IPv6 reverse look up, or it can leverage off the HHIT structure. Using the allocated prefix for HHITs TBD6 [suggested value 2001:30::/28] (See Section 3.1), the RAA is 10 and the HDA is 20, the DET is:

```
2001:30:280:1405:a3ad:1952:ad0:a69e
```

A DET reverse lookup could be to:

```
a69e.ad0.1952.a3ad.1405.280.30.2001.20.10.det.arpa.
```

or:

```
a3ad1952ad0a69e.5.20.10.30.2001.det.remoteid.icao.int.
```

A 'standard' ip6.arpa RR has the advantage of only one Registry service supported.

```
$ORIGIN 5.0.4.1.0.8.2.0.0.3.0.0.1.0.0.2.ip6.arpa.
e.9.6.a.0.d.a.0.2.5.9.1.d.a.3.a    IN    PTR
a3ad1952ad0a69e.20.10.det.rid.icao.int.
```

This DNS entry for the DET can also provide a revocation service. For example, instead of returning the HI RR it may return some record showing that the HI (and thus DET) has been revoked. Guidance on revocation service will be provided in [drip-registries].

## 6. Other UTM Uses of HHITs Beyond DET

HHITs will be used within the UTM architecture beyond DET (and USS in UA ID registration and authentication), for example, as a Ground Control Station (GCS) HHIT ID. Some GCS will use its HHIT for securing its Network Remote ID (to USS HHIT) and Command and Control (C2, Section 2.2.2 of [RFC9153]) transports.

Observers may have their own HHITs to facilitate UAS information retrieval (e.g., for authorization to private UAS data). They could also use their HHIT for establishing a HIP connection with the UA Pilot for direct communications per authorization. Details about such issues are out of the scope of this document).

## 7. Summary of Addressed DRIP Requirements

This document provides the details to solutions for GEN 1 - 3, ID 1 - 5, and REG 1 - 2 requirements that are described in [RFC9153].

## 8. IANA Considerations

### 8.1. New Well-Known IPv6 prefix for DETs

Since the DET format is not compatible with [RFC7343], IANA is requested to allocate a new prefix following this template for the IPv6 Special-Purpose Address Registry.

Address Block:

IANA is requested to allocate a new 28-bit prefix out of the IANA IPv6 Special Purpose Address Block, namely 2001::/23, as per [RFC6890] (TBD6, suggested: 2001:30::/28).

Name:

This block should be named "DRIP Entity Tags (DETs) Prefix".

RFC:

This document.

Allocation Date:

Date this document published.

Termination Date:

Forever.

Source:

False.



Destination:  
False.

Forwardable:  
False.

Globally Reachable:  
False.

Reserved-by-Protocol:  
False.

## 8.2. New IANA DRIP Registry

This document requests IANA to create a new registry titled "Drone Remote ID Protocol" registry. The following two subregistries should be created under that registry.

### Hierarchical HIT (HHIT) Prefixes:

Initially, for DET use, one 28-bit prefix should be assigned out of the IANA IPv6 Special Purpose Address Block, namely 2001::/23, as per [RFC6890]. Future additions to this subregistry are to be made through Expert Review (Section 4.5 of [RFC8126]). Entries with network-specific prefixes may be present in the registry.

HHIT Use	Bits	Value
DET	28	TBD6 (suggested value 2001:30::/28)

### Hierarchical HIT (HHIT) Suite ID:

This 8-bit valued subregistry is a superset of the 4/8-bit "HIT Suite ID" subregistry of the "Host Identity Protocol (HIP) Parameters" registry in [IANA-HIP]. Future additions to this subregistry are to be made through IETF Review (Section 4.8 of [RFC8126]). The following HHIT Suite IDs are defined:

HHIT Suite	Value
RESERVED	0
RSA,DSA/SHA-256	1 [RFC7401]
ECDSA/SHA-384	2 [RFC7401]
ECDSA_LOW/SHA-1	3 [RFC7401]
EdDSA/cSHAKE128	TBD3 (suggested value 5) (RECOMMENDED)
HDA Private Use 1	TBD4 (suggested value 254)
HDA Private Use 2	TBD5 (suggested value 255)

The HHIT Suite ID values 1 - 31 are reserved for IDs that MUST be replicated as HIT Suite IDs (Section 8.4) as is TBD3 here. Higher values (32 - 255) are for those Suite IDs that need not or cannot be accommodated as a HIT Suite ID.

### 8.3. IANA CGA Registry Update

This document requests that this document be added to the reference field for the "CGA Extension Type Tags" registry [IANA-CGA], where IANA registers the following Context ID:

Context ID:

The Context ID (Section 3) shares the namespace introduced for CGA Type Tags. Defining new Context IDs follow the rules in Section 8 of [RFC3972]:

Context ID := 0x00B5 A69C 795D F5D5 F008 7F56 843F 2C40

### 8.4. IANA HIP Registry Updates

This document requests IANA to make the following changes to the IANA "Host Identity Protocol (HIP) Parameters" [IANA-HIP] registry:

Host ID:

This document defines the new EdDSA Host ID with value TBD1 (suggested: 13) (Section 3.4.1) in the "HI Algorithm" subregistry of the "Host Identity Protocol (HIP) Parameters" registry.

Algorithm profiles	Values
EdDSA	TBD1 (suggested value 13) [RFC8032] (RECOMMENDED)

EdDSA Curve Label:

This document specifies a new algorithm-specific subregistry named "EdDSA Curve Label". The values for this subregistry are defined in Section 3.4.1.1. Future additions to this subregistry are to be made through IETF Review (Section 4.8 of [RFC8126]).

Algorithm	Curve	Values	
EdDSA	RESERVED	0	
EdDSA	EdDSA25519	1 [RFC8032]	(RECOMMENDED)
EdDSA	EdDSA25519ph	2 [RFC8032]	
EdDSA	EdDSA448	3 [RFC8032]	(RECOMMENDED)
EdDSA	EdDSA448ph	4 [RFC8032]	
		5-65535	Unassigned

**HIT Suite ID:**

This document defines the new HIT Suite of EdDSA/cSHAKE with value TBD3 (suggested: 5) (Section 3.4.2) in the "HIT Suite ID" subregistry of the "Host Identity Protocol (HIP) Parameters" registry.

HIT Suite	Value	
EdDSA/cSHAKE128	TBD3 (suggested value 5)	(RECOMMENDED)

The HIT Suite ID 4-bit values 1 - 15 and 8-bit values 0x00 - 0x0F MUST be replicated as HHIT Suite IDs (Section 8.2) as is TBD3 here.

**8.5. IANA IPSECKEY Registry Update**

This document requests IANA to make the following change to the "IPSECKEY Resource Record Parameters" [IANA-IPSECKEY] registry:

**IPSECKEY:**

This document defines the new IPSECKEY value TBD2 (suggested: 4) (Section 3.4.1.2) in the "Algorithm Type Field" subregistry of the "IPSECKEY Resource Record Parameters" registry.

**Value Description**

TBD2 (suggested value 4)  
An EdDSA key is present, in the format defined in [RFC8080]

**9. Security Considerations**

The 64-bit hash in HHITs presents a real risk of second pre-image cryptographic hash attack Section 9.4. There are no known (to the authors) studies of hash size to cryptographic hash attacks. A Python script is available to randomly generate 1M HHITs that did not produce a hash collision which is a simpler attack than a first or second pre-image attack.

However, with today's computing power, producing  $2^{64}$  EdDSA keypairs and then generating the corresponding HHIT is economically feasible. Consider that a \*single\* bitcoin mining ASIC can do on the order of  $2^{46}$  sha256 hashes a second or about  $2^{62}$  hashes in a single day. The point being,  $2^{64}$  is not prohibitive, especially as this can be done in parallel.

Now it should be noted that the  $2^{64}$  attempts is for stealing a specific HHIT. Consider a scenario of a street photography company with 1,024 UAs (each with its own HHIT); you'd be happy stealing any one of them. Then rather than needing to satisfy a 64-bit condition on the cSHAKE128 output, you need only satisfy what is equivalent to a 54-bit condition (since there are  $2^{10}$  more opportunities for success).

Thus, although the probability of a collision or pre-image attack is low in a collection of 1,024 HHITs out of a total population of  $2^{64}$ , per Section 9.4, it is computationally and economically feasible. Therefore, the HHIT registration and HHIT/HI registration validation is strongly recommended.

The DET Registry services effectively block attempts to "take over" or "hijack" a DET. It does not stop a rogue attempting to impersonate a known DET. This attack can be mitigated by the receiver of messages containing DETs using DNS to find the HI for the DET. As such, use of DNSSEC by the DET registries is recommended to provide trust in HI retrieval.

The 60-bit hash for DETs with 8-bit OGAs have a greater hash attack risk. As such its use should be restricted to testing and to small, well managed UAS/USS.

Another mitigation of HHIT hijacking is if the HI owner (UA) supplies an object containing the HHIT and signed by the HI private key of the HDA such as detailed in [drip-authentication].

The two risks with hierarchical HITs are the use of an invalid HID and forced HIT collisions. The use of a DNS zone (e.g., "det.arpa.") is a strong protection against invalid HIDs. Querying an HDA's RVS for a HIT under the HDA protects against talking to unregistered clients. The Registry service [drip-registries], through its HHIT uniqueness enforcement, provides against forced or accidental HHIT hash collisions.

Cryptographically Generated Addresses (CGAs) provide an assurance of uniqueness. This is two-fold. The address (in this case the UAS ID) is a hash of a public key and a Registry hierarchy naming. Collision resistance (more important than it implied second-preimage

resistance) makes it statistically challenging to attacks. A registration process [drip-registries] within the HDA provides a level of assured uniqueness unattainable without mirroring this approach.

The second aspect of assured uniqueness is the digital signing (attestation) process of the DET by the HI private key and the further signing (attestation) of the HI public key by the Registry's key. This completes the ownership process. The observer at this point does not know what owns the DET, but is assured, other than the risk of theft of the HI private key, that this UAS ID is owned by something and is properly registered.

#### 9.1. DET Trust in ASTM messaging

The DET in the ASTM Basic ID Message (Msg Type 0x0, the actual Remote ID message) does not provide any assertion of trust. The best that might be done within this Basic ID Message is 4 bytes truncated from a HI signing of the HHIT (the UA ID field is 20 bytes and a HHIT is 16). This is not trustable; that is, too open to a hash attack. Minimally, it takes 84 bytes (Section 4.6) to prove ownership of a DET with a full EdDSA signature. Thus, no attempt has been made to add DET trust directly within the very small Basic ID Message.

The ASTM Authentication Message (Msg Type 0x2) as shown in Section 4.6 can provide practical actual ownership proofs. These attestations include timestamps to defend against replay attacks. But in themselves, they do not prove which UA sent the message. They could have been sent by a dog running down the street with a Broadcast Remote ID module strapped to its back.

Proof of UA transmission comes when the Authentication Message includes proofs for the ASTM Location/Vector Message (Msg Type 0x1) and the observer can see the UA or that information is validated by ground multilateration. Only then does an observer gain full trust in the DET of the UA.

DETs obtained via the Network RID path provides a different approach to trust. Here the UAS SHOULD be securely communicating to the USS, thus asserting DET trust.

#### 9.2. DET Revocation

The DNS entry for the DET can also provide a revocation service. For example, instead of returning the HI RR it may return some record showing that the HI (and thus DET) has been revoked. Guidance on revocation service will be provided in [drip-registries].

### 9.3. Privacy Considerations

There is no expectation of privacy for DETs; it is not part of the privacy normative requirements listed in, Section 4.3.1, of [RFC9153]. DETs are broadcast in the clear over the open air via Bluetooth and Wi-Fi. They will be collected and collated with other public information about the UAS. This will include DET registration information and location and times of operations for a DET. A DET can be for the life of a UA if there is no concern about DET/UA activity harvesting.

Further, the MAC address of the wireless interface used for Remote ID broadcasts are a target for UA operation aggregation that may not be mitigated through MAC address randomization. For Bluetooth 4 Remote ID messaging, the MAC address is used by observers to link the Basic ID Message that contains the RID with other Remote ID messages, thus must be constant for a UA operation. This message linkage use of MAC addresses may not be needed with the Bluetooth 5 or Wi-Fi PHYs. These PHYs provide for a larger message payload and can use the Message Pack (Msg Type 0xF) and the Authentication Message to transmit the RID with other Remote ID messages. However, it is not mandatory to send the RID in a Message Pack or Authentication Message, so allowance for using the MAC address for UA message linking must be maintained. That is, the MAC address should be stable for at least a UA operation.

Finally, it is not adequate to simply change the DET and MAC for a UA per operation to defeat historically tracking a UA's activity.

Any changes to the UA MAC may have impacts to C2 setup and use. A constant GCS MAC may well defeat any privacy gains in UA MAC and RID changes. UA/GCS binding is complicated with changing MAC addresses; historically UAS design assumed these to be "forever" and made setup a one-time process. Additionally, if IP is used for C2, a changing MAC may mean a changing IP address to further impact the UAS bindings. Finally, an encryption wrapper's identifier (such as ESP [RFC4303] SPI) would need to change per operation to insure operation tracking separation.

Creating and maintaining UAS operational privacy is a multifaceted problem. Many communication pieces need to be considered to truly create a separation between UA operations. Simply changing the DET only starts the changes that need to be implemented.

These privacy realities may present challenges for the EU U-space (Appendix A) program.

#### 9.4. Collision Risks with DETs

The 64-bit hash size does have an increased risk of collisions over the 96-bit hash size used for the other HIT Suites. There is a 0.01% probability of a collision in a population of 66 million. The probability goes up to 1% for a population of 663 million. See Appendix C for the collision probability formula.

However, this risk of collision is within a single "Additional Information" value, i.e., a RAA/HDA domain. The UAS/USS registration process should include registering the DET and MUST reject a collision, forcing the UAS to generate a new HI and thus HHIT and reapplying to the DET registration process.

#### 10. References

##### 10.1. Normative References

[NIST.FIPS.202]

Dworkin, M., "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", National Institute of Standards and Technology report, DOI 10.6028/nist.fips.202, July 2015, <<https://doi.org/10.6028/nist.fips.202>>.

[NIST.SP.800-185]

Kelsey, J., Change, S., and R. Perlner, "SHA-3 derived functions: cSHAKE, KMAC, TupleHash and ParallelHash", National Institute of Standards and Technology report, DOI 10.6028/nist.sp.800-185, December 2016, <<https://doi.org/10.6028/nist.sp.800-185>>.

[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>>.

[RFC6890] Cotton, M., Vegoda, L., Bonica, R., Ed., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, DOI 10.17487/RFC6890, April 2013, <<https://www.rfc-editor.org/info/rfc6890>>.

[RFC7343] Laganier, J. and F. Dupont, "An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers Version 2 (ORCHIDv2)", RFC 7343, DOI 10.17487/RFC7343, September 2014, <<https://www.rfc-editor.org/info/rfc7343>>.

- [RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <<https://www.rfc-editor.org/info/rfc7401>>.
- [RFC8005] Laganier, J., "Host Identity Protocol (HIP) Domain Name System (DNS) Extension", RFC 8005, DOI 10.17487/RFC8005, October 2016, <<https://www.rfc-editor.org/info/rfc8005>>.
- [RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <<https://www.rfc-editor.org/info/rfc8032>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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>>.

## 10.2. Informative References

- [cfrg-comment] "A CFRG review of draft-ietf-drip-rid", September 2021, <[https://mailarchive.ietf.org/arch/msg/cfrg/tAJJq60W6TlUv7\\_pde5cw5TDTCU/](https://mailarchive.ietf.org/arch/msg/cfrg/tAJJq60W6TlUv7_pde5cw5TDTCU/)>.
- [corus] CORUS, "U-space Concept of Operations", September 2019, <<https://www.sesarju.eu/node/3411>>.
- [CTA2063A] ANSI/CTA, "Small Unmanned Aerial Systems Serial Numbers", September 2019, <<https://shop.cta.tech/products/small-unmanned-aerial-systems-serial-numbers>>.
- [drip-architecture] Card, S. W., Wiethuechter, A., Moskowitz, R., Zhao, S., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Architecture", Work in Progress, Internet-Draft, draft-ietf-drip-arch-22, 21 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-drip-arch-22>>.
- [drip-authentication] Wiethuechter, A., Card, S., and R. Moskowitz, "DRIP Entity Tag Authentication Formats & Protocols for Broadcast



Remote ID", Work in Progress, Internet-Draft, draft-ietf-drip-auth-10, 11 May 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-drip-auth-10>>.

[drip-registries]

Wiethuechter, A., Card, S., Moskowitz, R., and J. Reid, "DRIP Entity Tag Registration & Lookup", Work in Progress, Internet-Draft, draft-ietf-drip-registries-02, 30 April 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-drip-registries-02>>.

[F3411] ASTM International, "Standard Specification for Remote ID and Tracking", <<http://www.astm.org/cgi-bin/resolver.cgi?F3411>>.

[FAA\_RID] United States Federal Aviation Administration (FAA), "Remote Identification of Unmanned Aircraft", 2021, <<https://www.govinfo.gov/content/pkg/FR-2021-01-15/pdf/2020-28948.pdf>>.

[IANA-CGA] IANA, "Cryptographically Generated Addresses (CGA) Message Type Name Space", <<https://www.iana.org/assignments/cga-message-types/cga-message-types.xhtml>>.

[IANA-HIP] IANA, "Host Identity Protocol (HIP) Parameters", <<https://www.iana.org/assignments/hip-parameters/hip-parameters.xhtml>>.

[IANA-IPSECKEY]

IANA, "IPSECKEY Resource Record Parameters", <<https://www.iana.org/assignments/ipseckey-rr-parameters/ipseckey-rr-parameters.xhtml>>.

[Keccak] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., and R. Van Keer, "The Keccak Function", <<https://keccak.team/index.html>>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<https://www.rfc-editor.org/info/rfc3972>>.

[RFC4025] Richardson, M., "A Method for Storing IPsec Keying Material in DNS", RFC 4025, DOI 10.17487/RFC4025, March 2005, <<https://www.rfc-editor.org/info/rfc4025>>.

- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <<https://www.rfc-editor.org/info/rfc4034>>.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique Identifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <<https://www.rfc-editor.org/info/rfc4122>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC8004] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", RFC 8004, DOI 10.17487/RFC8004, October 2016, <<https://www.rfc-editor.org/info/rfc8004>>.
- [RFC8080] Sury, O. and R. Edmonds, "Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC", RFC 8080, DOI 10.17487/RFC8080, February 2017, <<https://www.rfc-editor.org/info/rfc8080>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC9063] Moskowitz, R., Ed. and M. Komu, "Host Identity Protocol Architecture", RFC 9063, DOI 10.17487/RFC9063, July 2021, <<https://www.rfc-editor.org/info/rfc9063>>.
- [RFC9153] Card, S., Ed., Wiethuechter, A., Moskowitz, R., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Requirements and Terminology", RFC 9153, DOI 10.17487/RFC9153, February 2022, <<https://www.rfc-editor.org/info/rfc9153>>.
- [RFC9224] Blanchet, M., "Finding the Authoritative Registration Data Access Protocol (RDAP) Service", STD 95, RFC 9224, DOI 10.17487/RFC9224, March 2022, <<https://www.rfc-editor.org/info/rfc9224>>.

## Appendix A. EU U-Space RID Privacy Considerations

The EU is defining a future of airspace management known as U-space within the Single European Sky ATM Research (SESAR) undertaking. Concept of Operation for European UTM Systems (CORUS) project proposed low-level Concept of Operations [corus] for UAS in the EU. It introduces strong requirements for UAS privacy based on European GDPR regulations. It suggests that UAs are identified with agnostic IDs, with no information about UA type, the operators or flight trajectory. Only authorized persons should be able to query the details of the flight with a record of access.

Due to the high privacy requirements, a casual observer can only query U-space if it is aware of a UA seen in a certain area. A general observer can use a public U-space portal to query UA details based on the UA transmitted "Remote identification" signal. Direct remote identification (DRID) is based on a signal transmitted by the UA directly. Network remote identification (NRID) is only possible for UAs being tracked by U-Space and is based on the matching the current UA position to one of the tracks.

This is potentially a contrary expectation as that presented in Section 9.3. U-space will have to deal with this reality within the GDPR regulations. Still, DETs as defined here present a large step in the right direction for agnostic IDs.

The project lists "E-Identification" and "E-Registrations" services as to be developed. These services can use DETs and follow the privacy considerations outlined in this document for DETs.

If an "agnostic ID" above refers to a completely random identifier, it creates a problem with identity resolution and detection of misuse. On the other hand, a classical HIT has a flat structure which makes its resolution difficult. The DET (Hierarchical HIT) provides a balanced solution by associating a registry with the UA identifier. This is not likely to cause a major conflict with U-space privacy requirements, as the registries are typically few at a country level (e.g., civil personal, military, law enforcement, or commercial).

## Appendix B. The 14/14 HID split

The following explains the logic behind selecting to divide the 28 bits of the HID into 2 14-bit components.

At this writing ICAO has 273 member "States", each may want to control RID assignment within its National Air Space (NAS). Some members may want separate RAAs to use for Civil, general Government,

and Military use. They may also want allowances for competing Civil RAA operations. It is reasonable to plan for 8 RAAs per ICAO member (plus regional aviation organizations like in the European Union). Thus at a start a 4,096 RAA space is advised.

There will be requests by commercial entities for their own, RAA allotments. Examples could include international organizations that will be using UAS and international delivery service associations. These may be smaller than the RAA space needed by ICAO member States and could be met with a 2,048 space allotment, but as will be seen, might as well be 4,096 as well.

This may well cover currently understood RAA entities. There will be future new applications, branching off into new areas. So yet another space allocation should be set aside. If this is equal to all that has been reserved, we should allow for 16,384 ( $2^{14}$ ) RAAs.

The HDA allocation follows a different logic from that of RAAs. Per Appendix C, an HDA should be able to easily assign 63M RIDs and even manage 663M with a "first come, first assigned" registration process. For most HDAs this is more than enough, and a single HDA assignment within their RAA will suffice. Most RAAs will only delegate to a couple HDAs for their operational needs. But there are major exceptions that point to some RAAs needing large numbers of HDA assignments.

Delivery service operators like Amazon (est. 30K delivery vans) and UPS (est. 500K delivery vans) may choose, for anti-tracking reasons, to use unique RIDs per day or even per operation. 30K delivery UA could need 11M upwards to 44M RIDs. Anti-tracking would be hard to provide if the HID were the same for a delivery service fleet, so such a company may turn to an HDA that provides this service to multiple companies so that who's UA is who's is not evident in the HID. A USS providing this service could well use multiple HDA assignments per year, depending on strategy.

Perhaps a single RAA providing HDAs for delivery service (or similar behaving) UAS could 'get by' with a 2048 HDA space (11-bits). So the HDA space could well be served with only 12 bits allocated out of the 28-bit HID space. But as this is speculation, and it will take years of deployment experience, a 14-bit HDA space has been selected.

There may also be 'small' ICAO member States that opt for a single RAA and allocate their HDAs for all UA that are permitted in their NAS. The HDA space is large enough that some to use part for government needs as stated above and for small commercial needs. Or the State may use a separate, consecutive RAA for commercial users. Thus it would be 'easy' to recognize State-approved UA by HID high-order bits.

#### Appendix C. Calculating Collision Probabilities

The accepted formula for calculating the probability of a collision is:

$$p = 1 - e^{\{-k^2/(2n)\}}$$

P Collision Probability  
 n Total possible population  
 k Actual population

The following table provides the approximate population size for a collision for a given total population.

Total Population	Deployed Population With Collision Risk of	
	.01%	1%
2 <sup>96</sup>	4T	42T
2 <sup>72</sup>	1B	10B
2 <sup>68</sup>	250M	2.5B
2 <sup>64</sup>	66M	663M
2 <sup>60</sup>	16M	160M

#### Acknowledgments

Dr. Gurtoov is an adviser on Cybersecurity to the Swedish Civil Aviation Administration.

Quynh Dang of NIST gave considerable guidance on using Keccak and the NIST supporting documents. Joan Deamen of the Keccak team was especially helpful in many aspects of using Keccak. Nicholas Gajcowski [cfrg-comment] provided a concise hash pre-image security assessment via the CFRG list.

Many thanks to Michael Richardson and Brian Haberman for the iotdir review, Magnus Nystrom for the secdir review, Elwyn Davies for genart review and DRIP co-chair and draft shepherd, Mohamed Boucadair for his extensive comments and help on document clarity.

## Authors' Addresses

Robert Moskowitz  
HTT Consulting  
Oak Park, MI 48237  
United States of America  
Email: rgm@labs.htt-consult.com

Stuart W. Card  
AX Enterprize, LLC  
4947 Commercial Drive  
Yorkville, NY 13495  
United States of America  
Email: stu.card@axenterprize.com

Adam Wiethuechter  
AX Enterprize, LLC  
4947 Commercial Drive  
Yorkville, NY 13495  
United States of America  
Email: adam.wiethuechter@axenterprize.com

Andrei Gurtov  
Linköping University  
IDA  
SE-58183 Linköping  
Sweden  
Email: gurtov@acm.org

drip Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 26 April 2022

A. Wiethuechter  
S. Card  
AX Enterprize, LLC  
R. Moskowitz  
HTT Consulting  
23 October 2021

DRIP Registries  
draft-wiethuechter-drip-registries-01

Abstract

TODO

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 26 April 2022.

Copyright Notice

Copyright (c) 2021 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 the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
2.1. Required Terminology . . . . .	4
2.2. Definitions . . . . .	4
3. Claims, Assertions, Attestations & Certificates . . . . .	4
4. DRIP Attestations & Certificates . . . . .	5
4.1. Attestation Structure . . . . .	5
4.1.1. Attestor Identity Information . . . . .	6
4.1.2. Attestation Data . . . . .	6
4.1.3. Expiration Timestamp . . . . .	7
4.1.4. Signing Timestamp . . . . .	7
4.1.5. Signature . . . . .	7
4.2. Attestations . . . . .	7
4.2.1. Self-Attestation (SA-xx) . . . . .	7
4.2.2. Attestation (A-xy) . . . . .	8
4.2.3. Concise Attestation (CA-xy) . . . . .	9
4.2.4. Mutual Attestation (MA-xy) . . . . .	10
4.2.5. Link Attestation (LA-xy) . . . . .	11
4.2.6. Broadcast Attestation (BA-xy) . . . . .	12
4.3. Certificates . . . . .	14
4.3.1. Attestation Certificate (AC-zxy) . . . . .	14
4.3.2. Concise Certificate (CC-zxy) . . . . .	15
4.3.3. Link Certificate (LC-zxy) . . . . .	15
4.3.4. Mutual Certificate (MC-zxy) . . . . .	16
5. Registries . . . . .	17
5.1. Classes . . . . .	17
5.1.1. Root . . . . .	18
5.1.2. Registered Assigning Authorities . . . . .	18
5.1.3. Hierarchical HIT Domain Authorities . . . . .	18
5.2. Federation . . . . .	19
6. DRIP Fully Qualified Domain Names . . . . .	19
6.1. Serial Number . . . . .	19
6.2. DET . . . . .	19
7. Supported DNS Records . . . . .	20
7.1. HIP RR . . . . .	20
7.2. CERT RR . . . . .	20
7.3. NS RR . . . . .	20
7.4. AAAA RR . . . . .	20
8. Registry Operations . . . . .	20
8.1. Registering an RAA . . . . .	21
8.1.1. Inputs . . . . .	21
8.1.2. DNS Entries . . . . .	21
8.1.3. Database Entries . . . . .	21
8.1.4. Outputs . . . . .	21
8.2. Registering an IRM . . . . .	21
8.2.1. Inputs . . . . .	22



8.2.2.	DNS Entries . . . . .	22
8.2.3.	Database Entries . . . . .	22
8.2.4.	Outputs . . . . .	22
8.3.	Registering an HDA . . . . .	22
8.3.1.	Inputs . . . . .	22
8.3.2.	DNS Entries . . . . .	23
8.3.3.	Database Entries . . . . .	23
8.3.4.	Outputs . . . . .	23
8.4.	Registering an MRA . . . . .	23
8.4.1.	Inputs . . . . .	23
8.4.2.	DNS Entries . . . . .	23
8.4.3.	Database Entries . . . . .	24
8.4.4.	Outputs . . . . .	24
8.5.	Registering a Serial Number . . . . .	24
8.5.1.	Inputs . . . . .	24
8.5.2.	DNS Entries . . . . .	24
8.5.3.	Database Entries . . . . .	24
8.5.4.	Outputs . . . . .	25
8.6.	Registering an Operator . . . . .	25
8.6.1.	Inputs . . . . .	25
8.6.2.	DNS Entries . . . . .	25
8.6.3.	Database Entries . . . . .	25
8.6.4.	Outputs . . . . .	25
8.7.	Registering a Session ID . . . . .	25
8.7.1.	Inputs . . . . .	26
8.7.2.	DNS Entries . . . . .	26
8.7.3.	Database Entries . . . . .	26
8.7.4.	Outputs . . . . .	26
9.	Provisioning . . . . .	27
9.1.	Overview of Transactions . . . . .	27
9.2.	HHIT Delegation . . . . .	28
9.3.	Registry . . . . .	29
9.4.	Manufacturer . . . . .	29
9.5.	Operator . . . . .	30
9.6.	Aircraft . . . . .	31
9.6.1.	Standard Provisioning . . . . .	31
9.6.2.	Operator Assisted Provisioning . . . . .	33
9.6.3.	Initial Provisioning . . . . .	35
10.	Security Considerations . . . . .	35
11.	References . . . . .	35
11.1.	Normative References . . . . .	35
11.2.	Informative References . . . . .	35
	Authors' Addresses . . . . .	36

## 1. Introduction

TODO

## 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. Claims, Assertions, Attestations & Certificates

This section introduces the terms "Claims", "Assertions", "Attestations", and "Certificates" as used in DRIP. In DRIP certificate has a different context compared with security certificates and Public Key Infrastructure used in X.509.

### Claims:

A claim in DRIP is a predicate (e.g., "X is Y", "X has property Y", and most importantly "X owns Y" or "X is owned by Y").

### Assertions:

An assertion in DRIP is a set of claims. This definition is borrowed from JWT [RFC7519] and CWT [RFC8392].

### Attestations:

An attestation in DRIP is a signed assertion. The signer may be the claimant or a related party with stake in the assertion(s). Under DRIP this is normally used when an entity asserts a relationship with another entity, along with other information, and the asserting entity signs the assertion, thereby making it an attestation.

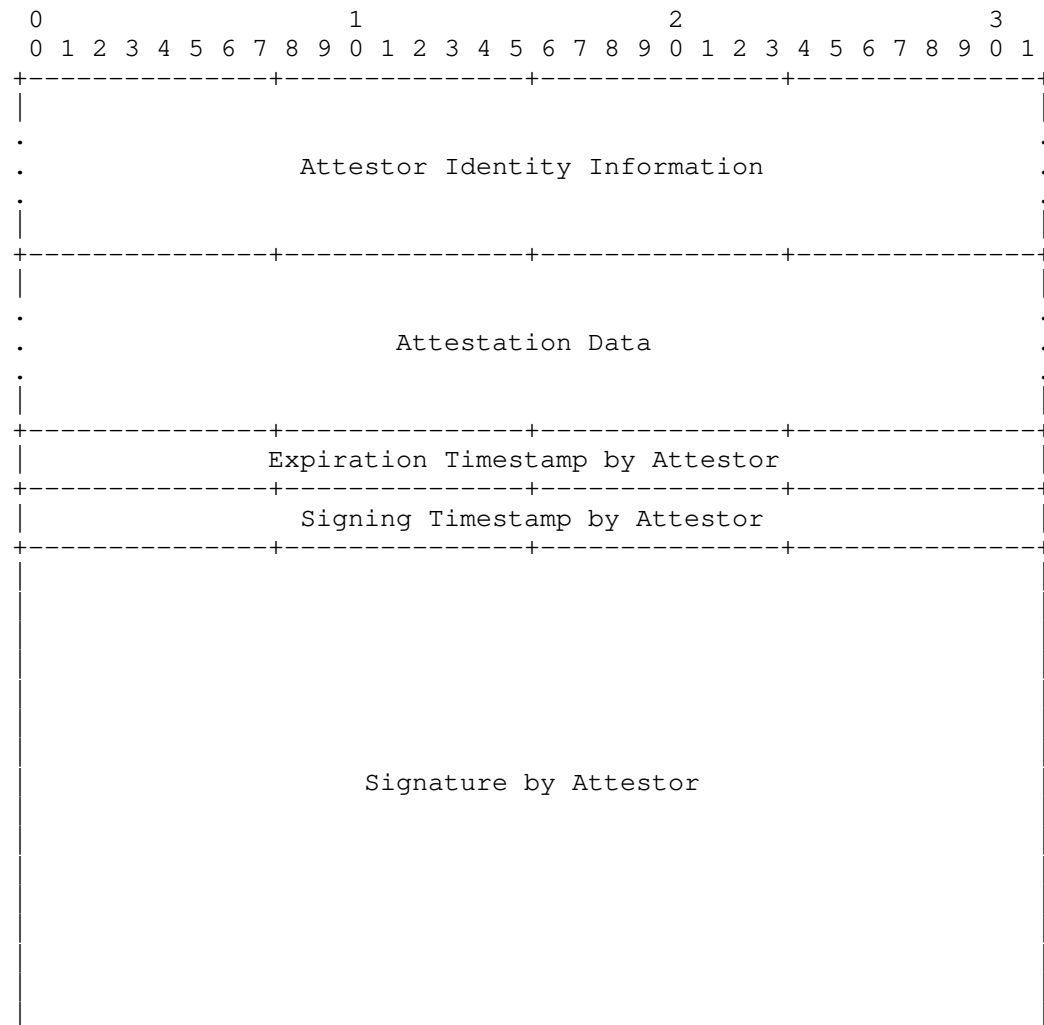
## Certificates:

A certificate in DRIP is an attestation, strictly over identity information, signed by a third party. This third party should be one with no stake in the attestation(s) its signing over.

## 4. DRIP Attestations &amp; Certificates

## 4.1. Attestation Structure

All Attestations and Certificates under DRIP share the following format:



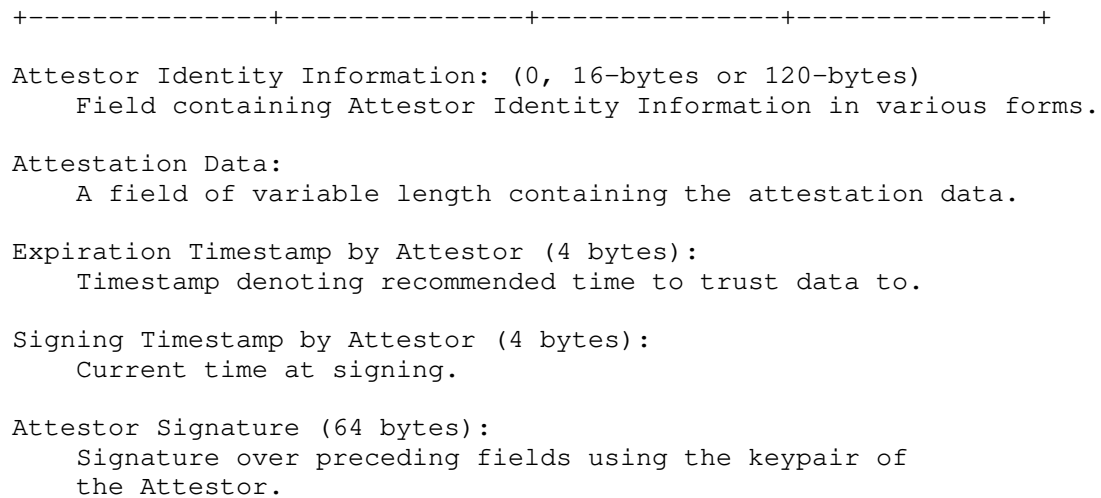


Figure 1: Attestation Structure

#### 4.1.1. 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 4.2.4 and LinkAttestation Section 4.2.5 as their definition clearly states that the signer is the second party with their HHIT or SelfAttestation already embedded in the Attestation Data.

#### 4.1.2. 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.

#### 4.1.3. Expiration Timestamp

TODO

#### 4.1.4. Signing Timestamp

TODO

#### 4.1.5. Signature

TODO

### 4.2. Attestations

#### 4.2.1. 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".



Figure 2: DRIP Self-Attestation

4.2.2. Attestation (A-xy)

(Editors Note: blurb here?)



Figure 3: DRIP Attestation

4.2.3. 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.

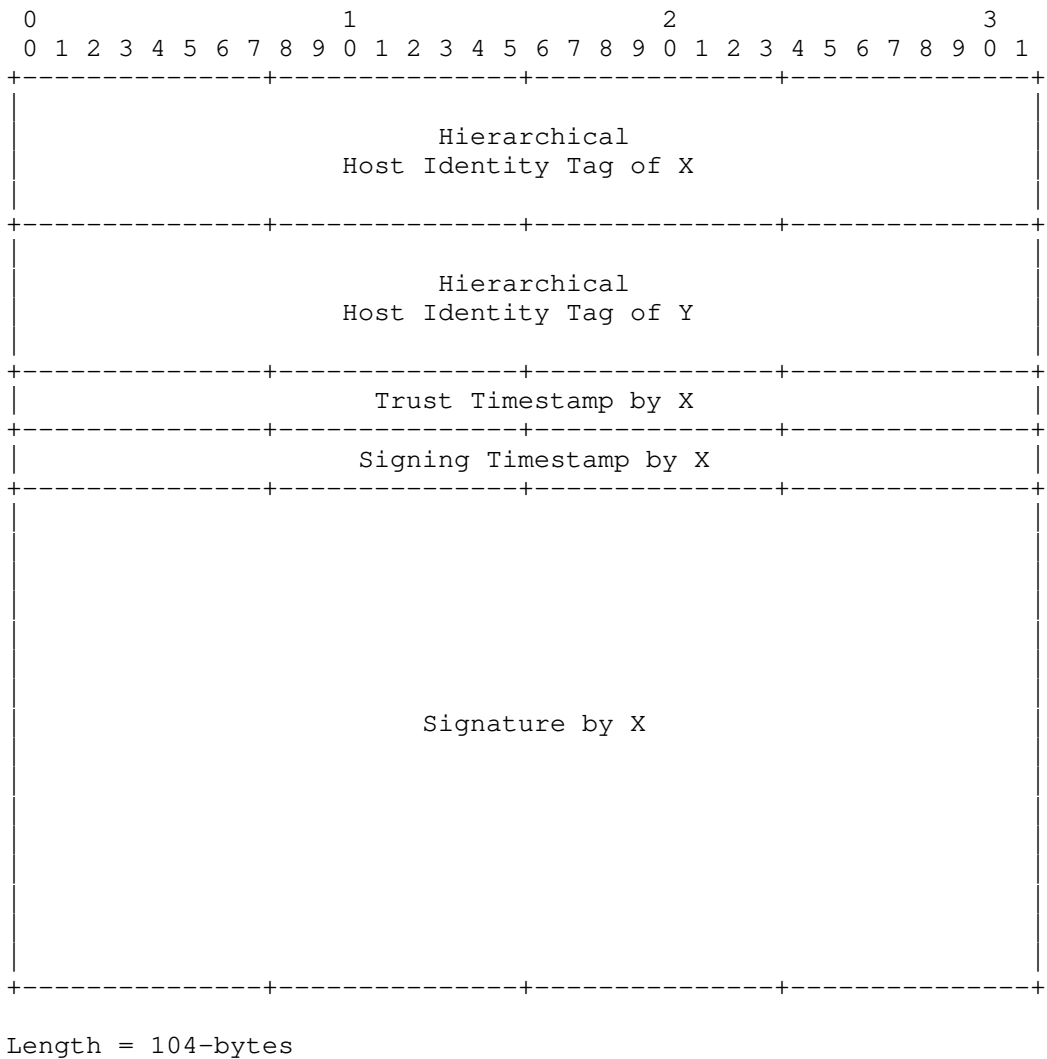


Figure 4: DRIP Concise Attestation

4.2.4. 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 4.1.1) as the data is already present in the "Attestation Data" (Section 4.1.2) in the form of Y's SelfAttestation.



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

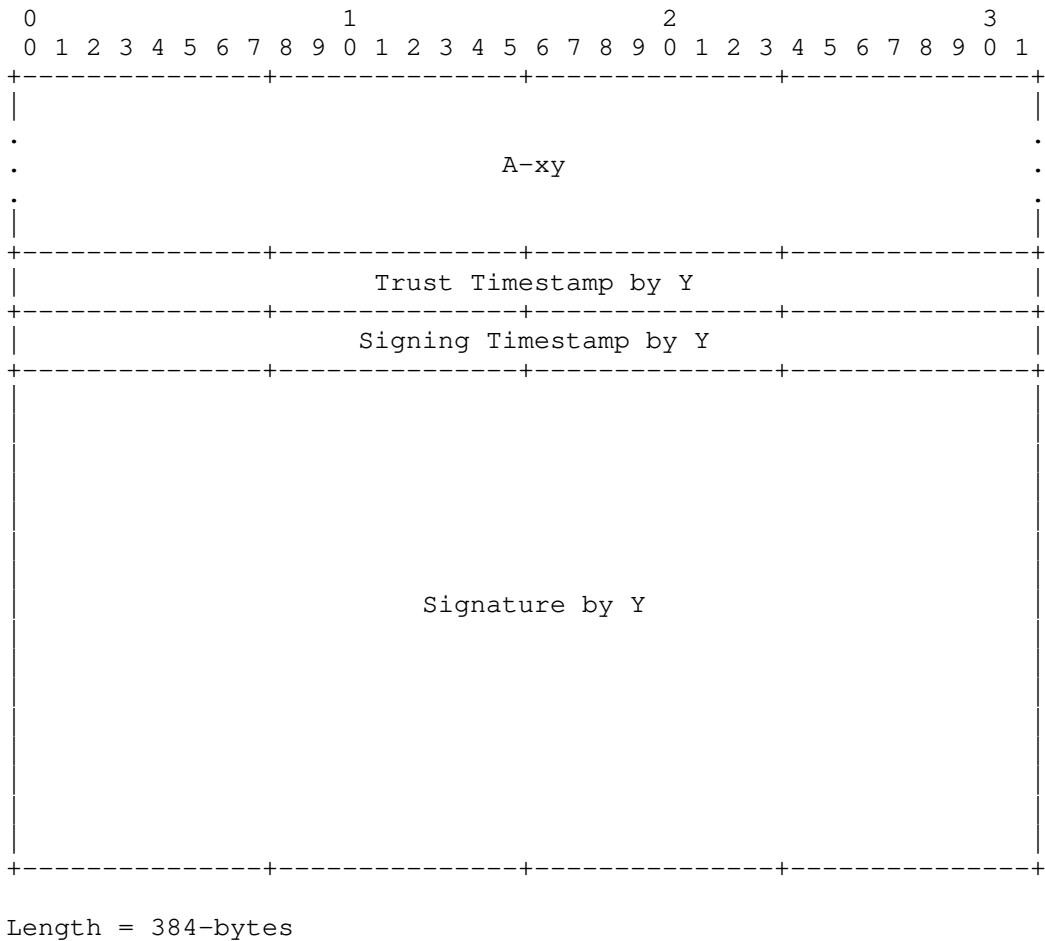


Figure 5: DRIP Mutual Attestation

4.2.5. 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 4.1.1) as the data is already present in the "Attestation Data" (Section 4.1.2) in the form of Y's HHIT.

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

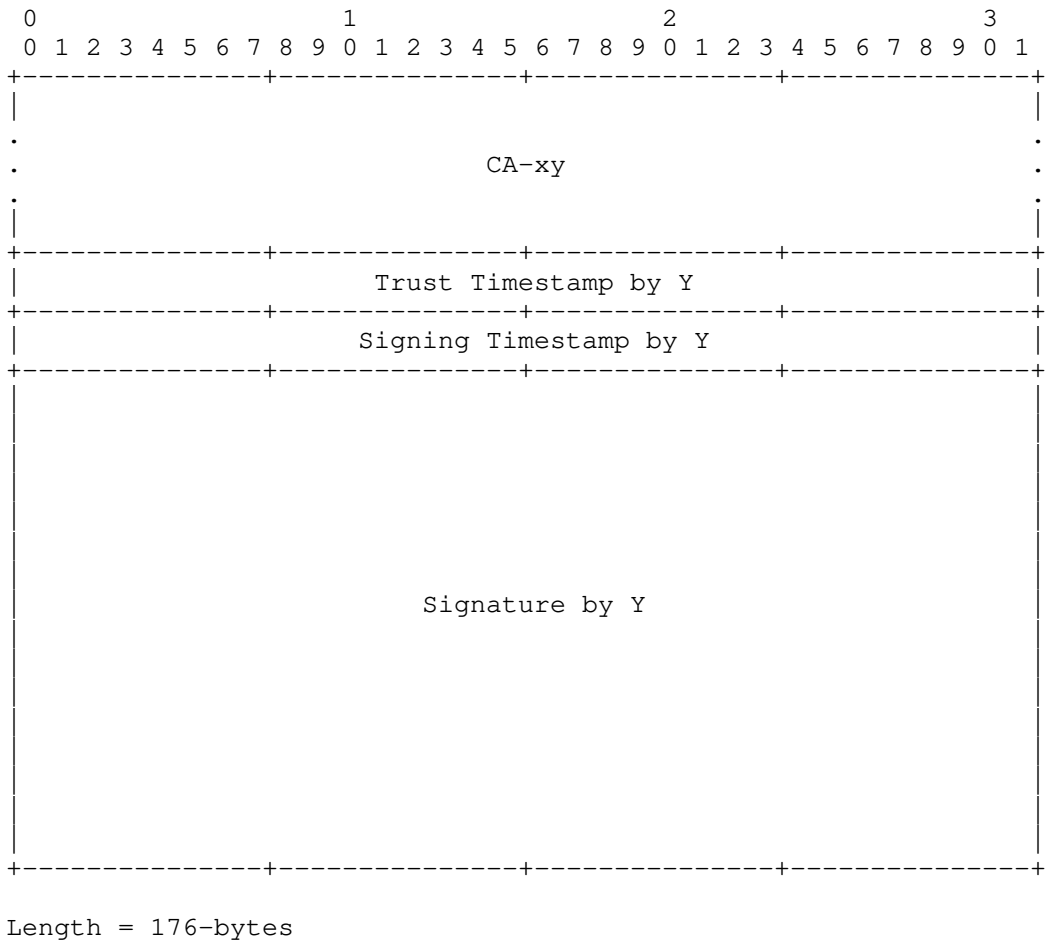


Figure 6: DRIP Link Attestation

4.2.6. 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.

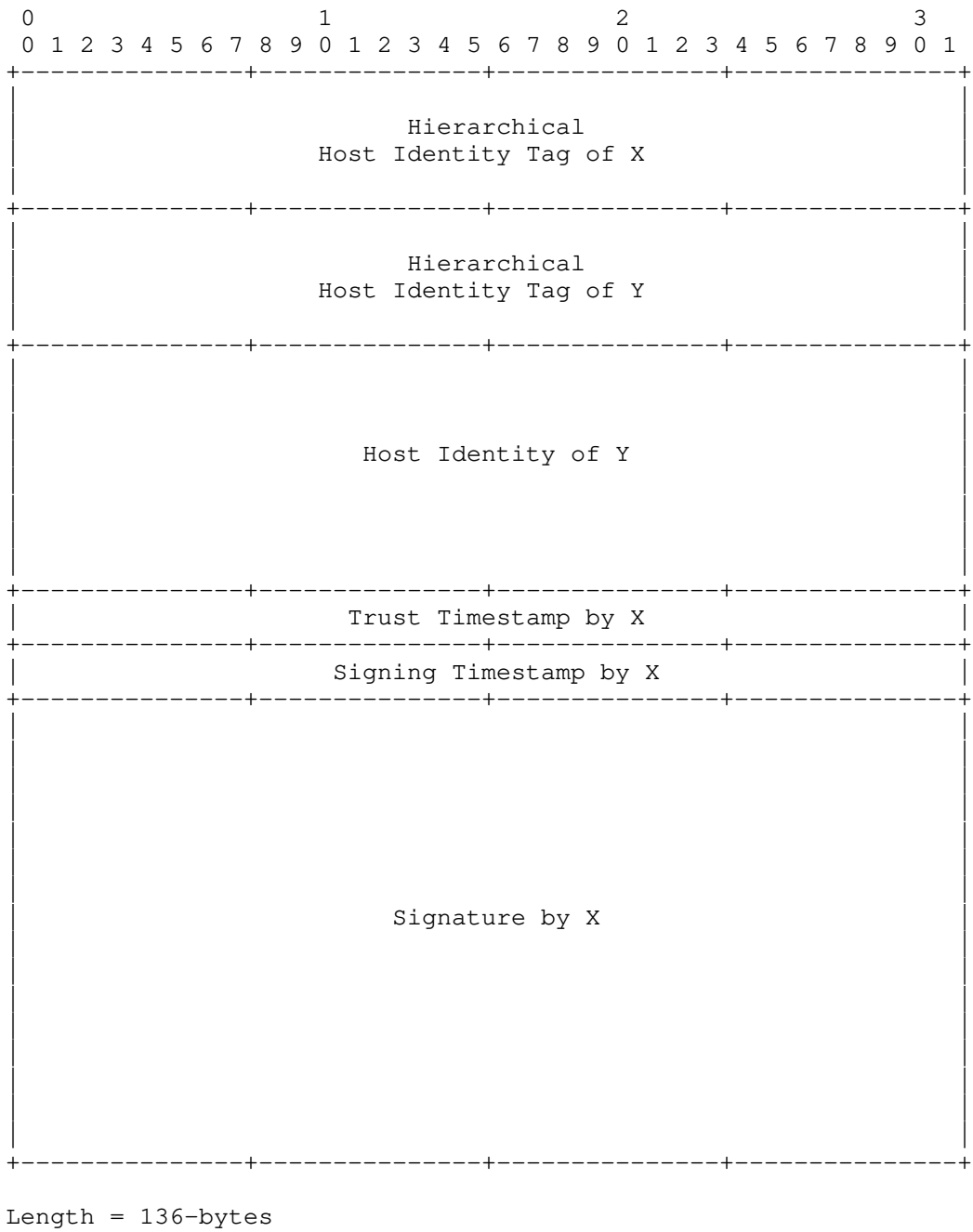


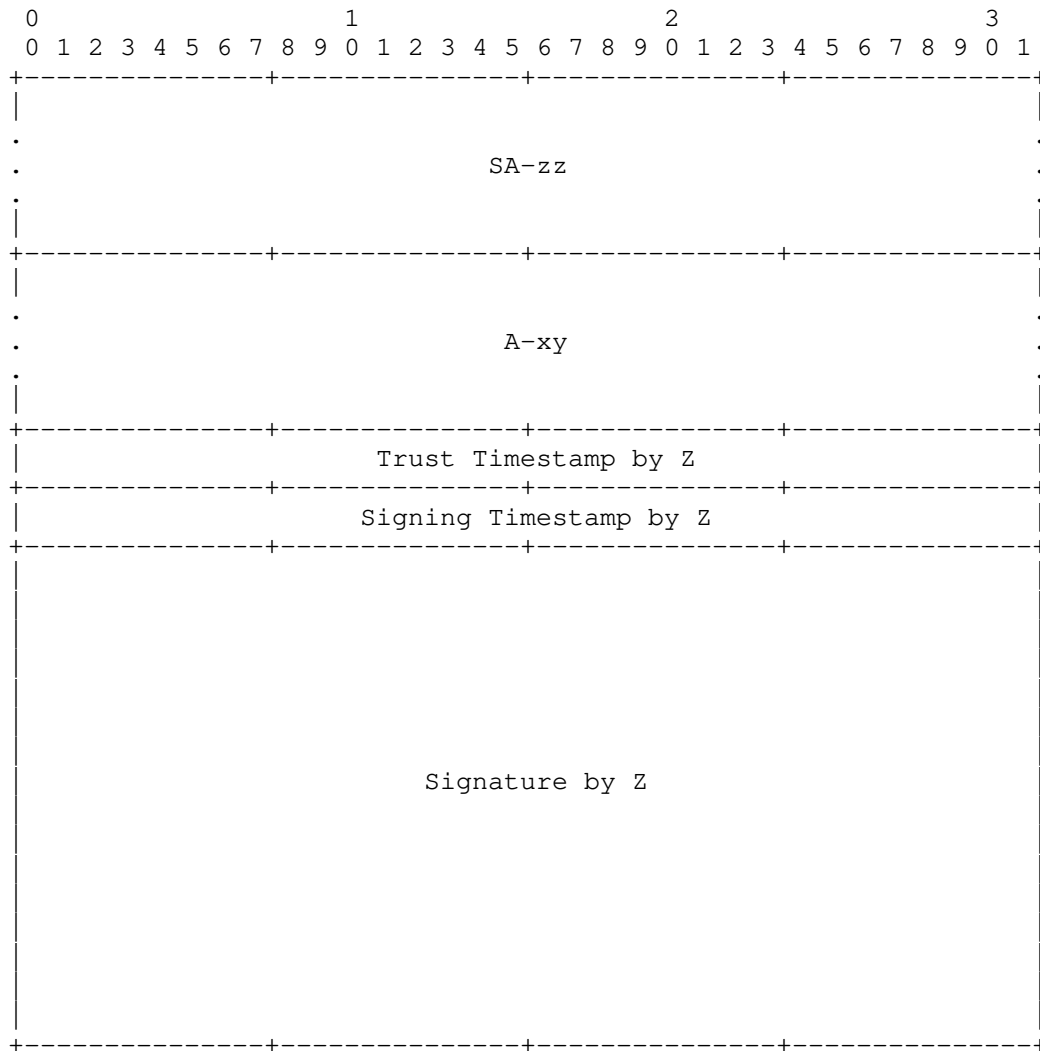
Figure 7: DRIP Broadcast Attestation

### 4.3. 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.

#### 4.3.1. Attestation Certificate (AC-zxy)



Length = 504-bytes

Figure 8: DRIP Attestation Certificate

4.3.2. Concise Certificate (CC-zxy)



Figure 9: DRIP Concise Certificate

4.3.3. Link Certificate (LC-zxy)

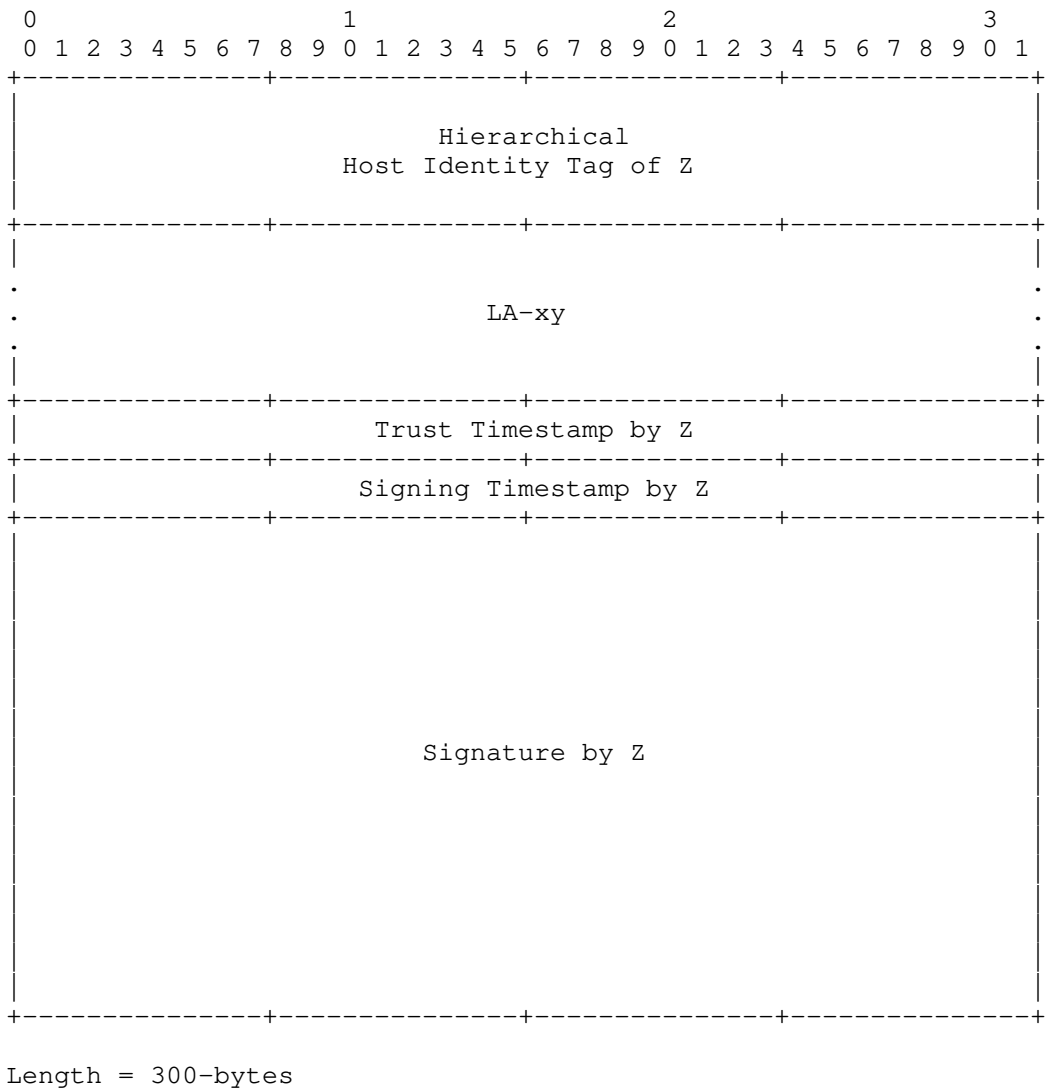


Figure 10: DRIP Link Certificate

4.3.4. Mutual Certificate (MC-zxy)



Figure 11: DRIP Mutual Certificate

5. Registries

5.1. Classes

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

#### 5.1.1. 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)

#### 5.1.2. Registered Assigning Authorities

TODO

Hold RAA values of 2+ and HDA value of 0.

Most are contemplated to be Civil Aviation Authorities (CAAs) then delegate HDAs to manage their NAS.

##### 5.1.2.1. ICAO Registry of Manufacturer's (IRM)

A special registry 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)

#### 5.1.3. Hierarchial HIT Domain Authorities

##### 5.1.3.1. Manufacturer's Registry of Aircraft (MRA)

A registry 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 values of 1 and HDA value of 1+.



#### 5.1.3.2. 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 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 value of 1+.

#### 5.2. 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?)

#### 6. DRIP Fully Qualified Domain Names

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

##### 6.1. Serial Number

Serial Number: 8653FZ2T7B8RA85D19LX  
ICAO Mfr Code: 8653  
Length Code: F  
ID: FZ2T7B8RA85D19LX  
FQDN: Z2T7B8RA85D19LX.F.8653.mfr.remoteid.aero

##### 6.2. DET

DET: 2001:0030:00a0:0145:a3ad:1952:0ad0:a69e  
ID: a3ad:1952:0ad0:a69e  
OGA: 5  
HDA: 0014 = 20  
RAA: 000a = 10  
FQDN: a3ad19520ad0a69e.5.20.10.det.remoteid.aero

(Editors Note: do we want to convert HDA/RAA to int or leave as hex?)

(Editors Note: DNS is case-sensitive in my experience, do we do all upper case?)

(Editors Note: do we support condensed ipv6 forms? - instinct is no as dns case-sensitive so it would be considered a different fqdn entirely)

## 7. Supported DNS Records

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

### 7.1. 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).

### 7.2. CERT RR

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

### 7.3. 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

### 7.4. AAAA RR

DRIP requires the use of IPv6.

## 8. 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

#### 8.1. Registering an RAA

Specifically handled by the Root Registry (Section 5.1.1).

##### 8.1.1. Inputs

Required:

1. SelfAttestation of RAA
2. IP Address of RAA

##### 8.1.2. 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 = ???

##### 8.1.3. Database Entries

##### 8.1.4. Outputs

#### 8.2. Registering an IRM

Specifically handled by the Root Registry (Section 5.1.1).

### 8.2.1. Inputs

Required:

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

### 8.2.2. 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 = ???

### 8.2.3. Database Entries

### 8.2.4. Outputs

Required:

1. Attestation: Root on IRM

## 8.3. Registering an HDA

Specifically handled by an RAA (Section 5.1.2).

### 8.3.1. Inputs

Required:

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

### 8.3.2. 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 ..."

### 8.3.3. Database Entries

### 8.3.4. Outputs

## 8.4. Registering an MRA

Specifically handled by the IRM Registry (Section 5.1.2.1).

### 8.4.1. Inputs

Required:

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

### 8.4.2. 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 = ???

#### 8.4.3. Database Entries

(HDA value, MRA Details)

#### 8.4.4. Outputs

Required:

1. Attestation: IRM on MRA

#### 8.5. Registering a Serial Number

Specifically handled by a MRA (Section 5.1.3.1).

#### 8.5.1. Inputs

Required:

1. Serial Number
2. Aircraft Metadata

Optional:

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

#### 8.5.2. DNS Entries

Required on MRA:

A/AAAA with Serial Number FQDN (Section 6.1)

Optional on MRA:

HIP RR of Aircraft with DET FQDN (Section 6.2) ("`<sn_det_fqdn>` HIP ...")

CERT RRs of SelfAttestation and BroadcastAttestation

#### 8.5.3. Database Entries

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

#### 8.5.4. Outputs

Optional:

1. BroadcastAttestation: Mfr on Aircraft

#### 8.6. Registering an Operator

Specifically handled by a RIDR (Section 5.1.3.2).

##### 8.6.1. Inputs

Required:

1. SelfAttestation: Operator on Operator
2. Operator PII

Optional: TODO

##### 8.6.2. DNS Entries

Optional on RIDR:

HIP RR of Operator

CERT RRs SelfAttestation of Operator, A-ro

##### 8.6.3. Database Entries

TODO

##### 8.6.4. 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

#### 8.7. Registering a Session ID

Specifically handled by a RIDR (Section 5.1.3.2).

#### 8.7.1. 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)

#### 8.7.2. DNS Entries

Required on RIDR:

HIP RR of Aircraft with DET FQDN (Section 6.2) ("`<session_det_fqdn>`  
HIP ...")

CERT RRs for SelfAttestation of Aircraft, BroadcastAttestation

#### 8.7.3. Database Entries

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

#### 8.7.4. 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

## 9. 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 9.6.2). The methods to securely inject and store keypair information in a "secure element" of the Aircraft is out of scope of this document.

### 9.1. 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

## 9.2. 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 the 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.

### 9.3. 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.

### 9.4. Manufacturer

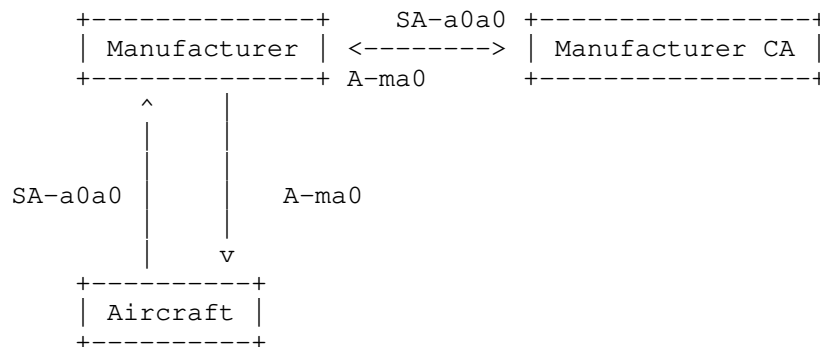


Figure 12: Manufacturer Provision

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.

#### 9.5. Operator

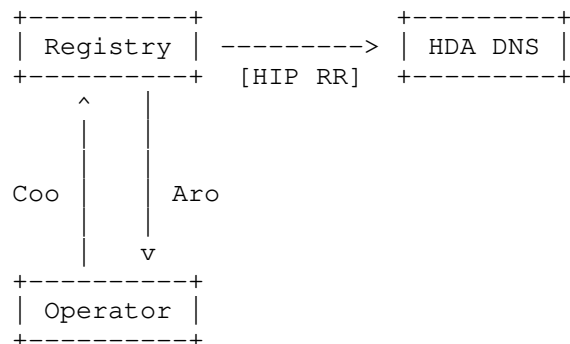


Figure 13: Operator Provision

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.

## 9.6. Aircraft

### 9.6.1. Standard Provisioning

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

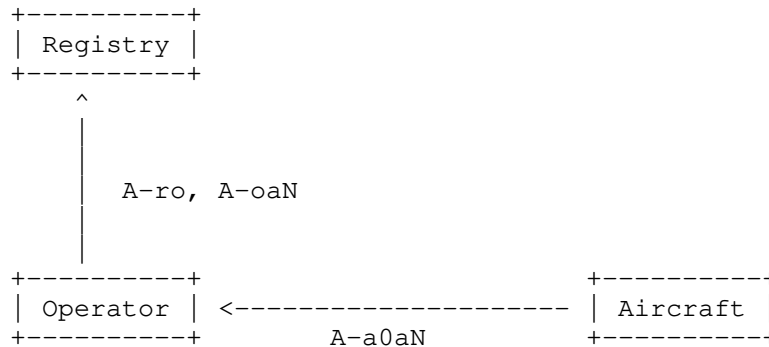


Figure 14: Standard Provision: Step 1

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).

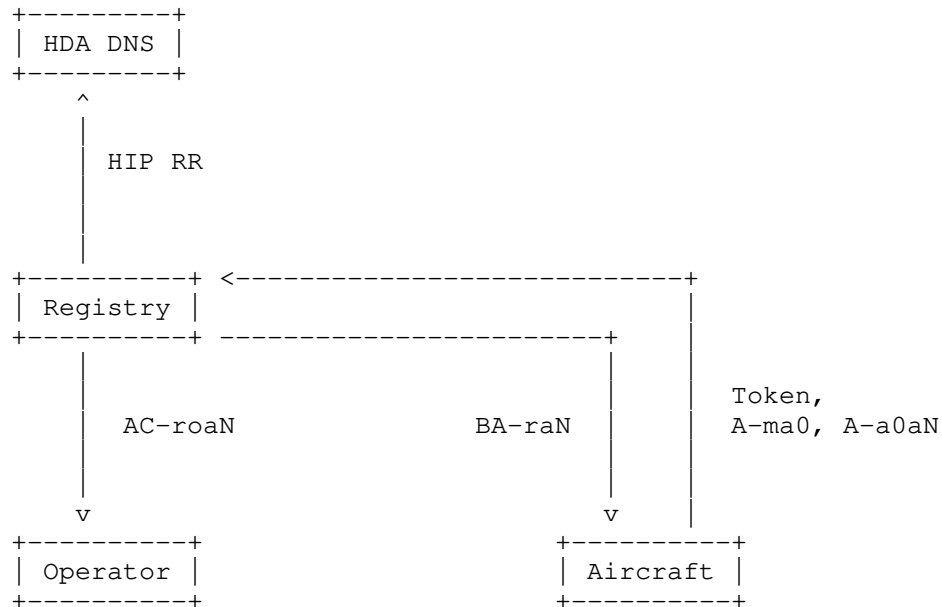


Figure 16: Standard Provision: Step 3

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).

#### 9.6.2. 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.

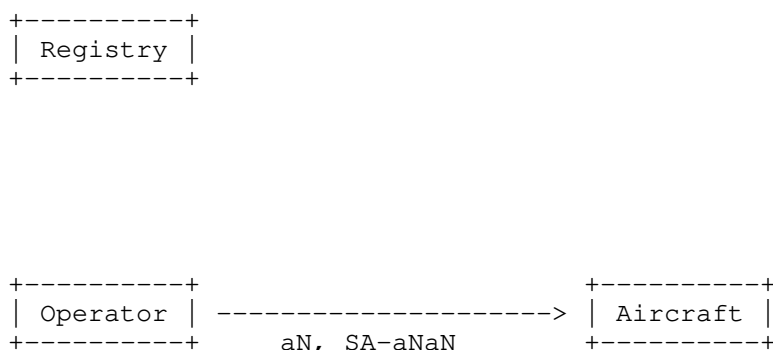


Figure 17: Operator Assisted Provision: Step 1

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.

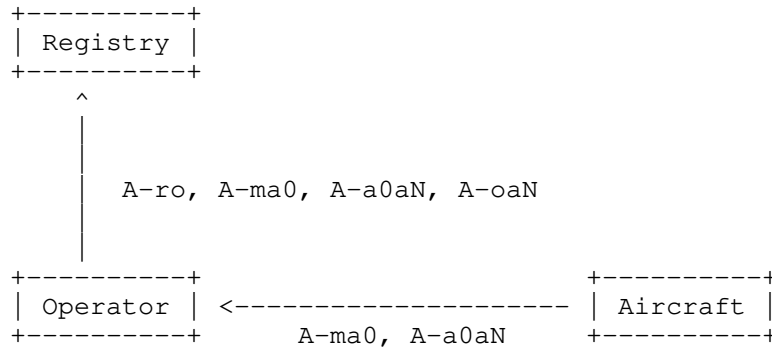


Figure 18: Operator Assisted Provision: Step 2

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).

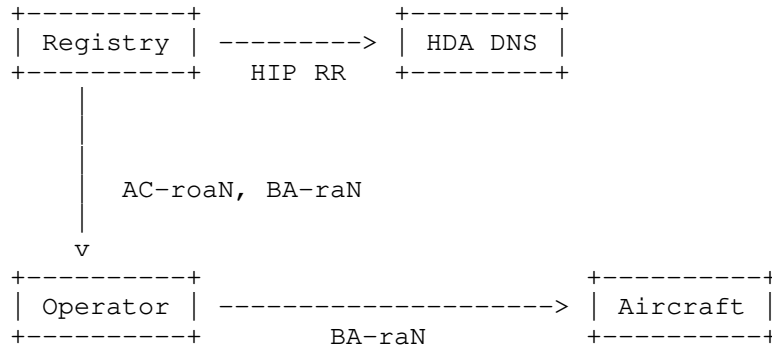


Figure 19: Operator Assisted Provision: Step 3

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.



### 9.6.3. 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.

## 10. Security Considerations

TODO

## 11. References

### 11.1. 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>>.

### 11.2. 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](mailto:adam.wiethuechter@axenterprize.com)

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

Email: [stu.card@axenterprize.com](mailto:stu.card@axenterprize.com)

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

Email: [rgm@labs.htt-consult.com](mailto:rgm@labs.htt-consult.com)