Drone Remote Identification Protocol (DRIP) Architecture
draft-ietf-drip-arch-17

Abstract

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

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 May 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
   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. Architecture Terminology ............................. 10
   2.2. Abbreviations ...................................... 10
   2.3. Additional Definitions ............................... 10
3. Claims, Assertions, Attestations, and Certificates ...... 10
4. HHIT as the DRIP Entity Identifier ....................... 11
   4.1. UAS Remote Identifiers Problem Space ................ 12
   4.2. HHIT as A Trustworthy DRIP Entity Identifier ....... 12
   4.3. HHIT for DRIP Identifier Registration and Lookup ... 14
   4.4. HHIT as a Cryptographic Identifier .................. 14
5. DRIP Identifier Registration and Registries ............. 14
   5.1. Public Information Registry .......................... 15
      5.1.1. Background ...................................... 15
      5.1.2. DNS as the Public DRIP Identifier Registry .... 15
   5.2. Private Information Registry ........................ 15
      5.2.1. Background ...................................... 15
      5.2.2. EPP and RDAP as the Private DRIP Identifier Registry .................................................. 16
      5.2.3. Alternative Private DRIP Registry methods ...... 16
6. DRIP Identifier Trust ................................... 16
7. Harvesting Broadcast Remote ID messages for UTM Inclusion 17
   7.1. The CS-RID Finder ................................... 18
   7.2. The CS-RID SDSP ..................................... 18
8. DRIP Contact ........................................... 18
9. IANA Considerations ..................................... 19
10. Security Considerations ................................ 19
11. Privacy & Transparency Considerations .................. 19
12. References ............................................ 20
   12.1. Normative References ............................... 20
   12.2. Informative References ............................. 20
Appendix A. Overview of Unmanned Aircraft Systems (UAS) Traffic Management (UTM) .............................. 23
   A.1. Operation Concept .................................. 23
1. Introduction

This document describes an architecture for protocols and services to support Unmanned Aircraft System Remote Identification and tracking (UAS RID), 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 [I-D.ietf-drip-reqs]. 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 enabler for a UAS to be identified by Unmanned Aircraft Systems Traffic Management (UTM) and UAS Service Supplier (USS) (Appendix A) or third parties 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 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 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 defines one set of 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 FAA in its 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 the services that can be offered based on 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.

1.2. Overview of Types of UAS Remote ID

1.2.1. Broadcast RID

[F3411] defines a set of 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 RF LOS, typically similar to visual Light-Of-Sight (LOS), 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 7).

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 RID is generated by the UAS (typically some by the UA and some by the GCS, e.g. their respective 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, describing the volumes but not the operations.

* An Observer’s device, expected typically 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 of 2021) typically a direct RF link, but with increasing beyond Visual Line of Sight (BVLOS) operations expected often to be wireless links 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, 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 RID information via the Internet to subscribed Net-RID DP, typically USS. Subscribed Net-RID DP forward RID information via the Internet to subscribed Observer devices. Regulations require and [F3411] describes 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 Discovery and Synchronization Service (DSS). It also prescribes data elements (in JSON) between Observer and 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 RID information is in the scope of [F3411] Network RID.

1.3. Overview of USS Interoperability

With Net-RID, there is direct communication between the UAS and its USS. With Broadcast-RID and UTM, the UAS Operator has either pre-filed a 4D space volume for USS operational knowledge and/or Observers can be providing information about observed UA to a Surveillance Supplemental Data Service Provider (SDSP). USS exchange information via a Discovery and Synchronization Service (DSS) so all USS collectively have knowledge about all activities in a 4D airspace.

The interactions among Observer, UA, and USS are shown in Figure 3.
Editor-note-1: (Stu) re-draw this figure and propose text. Then double check the language in Editor-note-8

1.4. Overview of DRIP Architecture

Figure 4 illustrates a brief summary of the general UAS RID usage scenarios in DRIP.
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 4).

Editor-note-2: Stu: replace figure 4
- Mechanisms to leverage Domain Name System (DNS [RFC1034]), Extensible Provisioning Protocol (EPP [RFC5731]) and Registration Data Access Protocol (RDAP) ([RFC7482]) for publishing public and private information (see Section 5.1 and Section 5.2).

- Specific authentication methods and message payload formats to enable verification that Broadcast RID messages were sent by the claimed sender (Section 6) and that sender is in the claimed registry (Section 5 and Section 6).

- Harvesting broadcast RID messages for UTM inclusion (Section 7).

- Methods for instantly establishing secure communications between an Observer and the pilot of an observed UAS (Section 8).

- Privacy in RID messages (PII protection) (Section 11).

2. Terms and Definitions

2.1. Architecture 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 above.

2.2. Abbreviations

EdDSA:      Edwards-Curve Digital Signature Algorithm
HHIT:       Hierarchical HIT
HIP:        Host Identity Protocol
HIT:        Host Identity Tag

2.3. Additional Definitions

This document uses terms defined in [I-D.ietf-drip-reqs].

3. Claims, Assertions, Attestations, and Certificates

Editor-note-7: (Bob) move section 3 to Section 2.4?
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) its signing over.

4. 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) as self-asserting IPv6 addresses suitable as a UAS ID type and more generally as trustworthy multipurpose remote identifiers.

Self-asserting in this usage is given the Host Identity (HI), the HHIT ORCHID construction and a signature of the HHIT by the HI can both be validated. The explicit registration hierarchy within the HHIT provides registry discovery (managed by a Registrar) to either yield the HI for 3rd-party (who is looking for ID attestation) validation or prove the HHIT and HI have uniquely been registered.
4.1. UAS Remote Identifiers Problem Space

A DRIP entity identifier needs to be "Trustworthy" (See DRIP Requirement GEN-1, ID-4 and ID-5 in [I-D.ietf-drip-reqs]). This means that given a sufficient collection of RID messages, an Observer can establish that the identifier claimed therein uniquely belongs to the claimant: that the only way for any other entity to prove ownership of that identifier would be to obtain information that ought to be available only to the legitimate owner of the identifier (e.g., a cryptographic private key).

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 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, Session IDs, to be standardized by IETF and other standard development organizations (SDOs) as extensions to ASTM 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 RID [F3411] Authentication Message payload is 201 bytes for most authentication types, but for type 5, also added in this revision, for IETF and other SDOs to develop Specific Authentication Methods as extensions to ASTM RID, one byte is consumed to index the sub-type, leaving only 200 for DRIP authentication payloads, including one or more DRIP entity identifiers and associated authentication data.

4.2. HHIT as A Trustworthy DRIP Entity Identifier

A Remote ID that can be trustworthily used in the RID Broadcast mode can be built from an asymmetric keypair. Rather than using a key signing operation to claim ownership of an ID that does not guarantee name uniqueness, in this method the ID is cryptographically derived directly from the public key. The proof of ID ownership (verifiable attestation, versus mere claim) is guaranteed by signing this cryptographic ID with the associated private key. The association between the ID and the private key is ensured by cryptographically binding the public key with the ID, more specifically the ID results from the hash of the public key. It is statistically hard for another entity to create a public key that would generate (spoof) the ID.
The basic HIT is designed statistically unique through the cryptographic hash feature of second-preimage resistance. The cryptographically-bound addition of the Hierarchy and an HHIT registration process (e.g. based on Extensible Provisioning Protocol, [RFC5730]) 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 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 10).

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 UA 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. The Operators, 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, 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.

An Observer would need Internet access to validate a self-attestations claim. A third-party Certificate can be validated via a small credential cache in a disconnected environment. This third-party Certificate is possible when the third-party also uses HHITs for its identity and the UA has the public key and the Certificate for that HHIT.

Editor-note-3: review the last/above pragraph.
4.3. HHIT for DRIP Identifier Registration and Lookup

Remote ID 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. A first-come-first-serve registration for a HHIT provides deterministic access to any other needed actionable information based on inquiry access authority (more details in Section 5.2).

4.4. HHIT as a Cryptographic Identifier

The only (known to the authors at the time of this writing) extant 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 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.

5. DRIP Identifier Registration and Registries

DRIP registries hold both public and private UAS information 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.
5.1. Public Information Registry

5.1.1. Background

The public registry provides trustable information such as attestations of 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 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 tend to directly provide all public registry information. The Observer may visually "see" these Net-RID UAS, but they may be silent to the Observer. The Net-RID DP is the only source of information based on a query for an airspace volume.

5.1.2. 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).

5.2. Private Information Registry

5.2.1. 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 ID structure compatible with DNS names. The HHIT hierarchy can provide the needed scalability and management structure. It is expected that the private 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.
5.2.2. 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) [RFC7482] [RFC7483]). 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].

5.2.3. 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.

6. DRIP Identifier Trust

Editor-note-5: Section 6 doesn’t use the word "authentication" in the section title, is there a reason to avoid it?

While the DRIP entity identifier is self-asserting, it alone does not provide the "trustworthiness" specified in [I-D.ietf-drip-reqs]. For that it MUST be registered (under DRIP Registries) and be actively used by the party (in most cases the UA). For Broadcast RID this is a challenge to balance the original requirements of Broadcast RID and the efforts needed to satisfy the DRIP requirements all under severe constraints.

An 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) 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 UAs claims.

This process (combining the DRIP entity identifier, Registries and Authentication Formats for Broadcast RID) can satisfy the following DRIP requirement defined in [I-D.ietf-drip-reqs]: GEN-1, GEN-2, GEN-3, ID-2, ID-3, ID-4 and ID-5.
7. Harvesting Broadcast Remote ID messages for UTM Inclusion

Editor-note-6: Section 7 needs to cite the corresponding numbered requirement that it supports.

ASTM anticipated that regulators would require both Broadcast RID and Network RID for large UAS, but allow 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 UAS of essentially all UAS and is now also considering Network RID. The FAA 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 obvious 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 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.

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 RID messages. This Crowd Sourced Remote ID (CS-RID) would be a significant enhancement, beyond baseline DRIP functionality; if implemented, it adds two more entity types.
7.1. The CS-RID Finder

A CS-RID Finder is the gateway for Broadcast Remote ID Messages into the 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 TBD interface to a CS-RID SDSP, similar to but readily distinguishable from that between a GCS and a Net-RID SP.

7.2. The CS-RID SDSP

A CS-RID SDSP aggregates and processes (e.g., estimates UA location using including 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.

Editor-note-8: double check above paragraph after Editor-note-1 is resolved.

8. 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 4 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 6 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). An Observer equipped with HIP can initiate a Base Exchange (BEX) and establish a Bound End to End Tunnel (BEET) protected by IPsec Encapsulating Security Payload (ESP) encryption to a likewise 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; likewise the USS, etc. Certain preconditions are necessary: each party to the communication 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. Given a BEET, 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. This approach satisfies DRIP requirement GEN-6 Contact, supports satisfaction of requirements [I-D.ietf-drip-reqs] GEN-8, GEN-9, PRIV-2, PRIV-5 and REG-3, and is compatible with all other DRIP requirements.

9. IANA Considerations

This document does not make any IANA request.

10. Security Considerations

The security provided by asymmetric cryptographic techniques depends upon protection of the private keys. A manufacturer that embeds a private key in an UA may have retained a copy. A manufacturer whose UA are configured by a closed source application on the GCS which communicates over the Internet with the factory may be sending a copy of a UA or GCS self-generated key back to the factory. Keys may be extracted from a GCS or UA. The 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.

11. 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.

12. References

12.1. Normative References


12.2. Informative References


[FAA_UAS_Concept_Of_Ops]

[Implementing]

[LAANC]


[NPRM]


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 of integrating UAS’s operation 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 UTM research transition team which is the joint workforce by FAA and NASA. 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 the UTM concept 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.

The UTM comprises UAS operation infrastructure, procedures and local regulation compliance policies to guarantee safe UAS integration and operation. The main functionality of a 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 a 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 UTM envisioned functionality. The LAANC program can automate the 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 a 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 (ex. 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 UA. 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. 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
Adam Wiethuechter
AX Enterprize
4947 Commercial Drive
Yorkville, NY, 13495
United States of America
Email: adam.wiethuechter@axenterprize.com

Robert Moskowitz
HTT Consulting
Oak Park, MI, 48237
United States of America
Email: 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
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 an 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 May 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.
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.

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.

Aircraft: In this document whenever the word Aircraft is used it is referring to an Unmanned Aircraft (UA) not a Manned Aircraft.

Legacy Transports: uses broadcast frames (Bluetooth 4.x).

Extended Transports: uses the extended advertisements (Bluetooth 5.X), 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.

3.2. 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 paged.

3.3.1. Authentication Page

```
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
+---------------+---------------+---------------+---------------+
|   Page Header  |                                               |
+---------------+                                               |
|                                                                 |
|                                                                 |
|                     Authentication Payload                     |
|                                                                 |
|                                                                 |
+---------------+---------------+---------------+---------------+
```

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

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)
Contains other header information for the Authentication Message as defined in F3411.

Authentication Data / Signature: (0 to 255 bytes)
Opaque authentication data.

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

Additional Data: (0 to 255 bytes):
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 abstract 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.

Full examples of Authentication Messages (fully paginated; both with and without Additional Data) can be found in Appendix B.
3.3.2. DRIP 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:

1. "Authentication Data / Signature" data MUST fit in a 9-page Authentication Message (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.X, Wi-Fi NaN, and Wi-Fi BEACON). Legacy Transports do not have supporting FEC so with DRIP Authentication the following application level FEC scheme is used.

(Editors Note: add in self-protecting and more-than-self-protecting options, with their justifications)

(Editors Note: Bob M. mentions that the FEC should be page aligned and the ADL includes null padding at the start to page align the data)

4.1. Encoding

4.1.1. Single Page FEC

To generate the parity a simple XOR operation using the previous and current page is used. For Page 0, a 25-byte null pad is used for the previous page. The resulting parity fills the "Additional Data" field of [F3411] with the "Additional Data Length" field being set to 25.

4.1.2. Multi Page FEC

TODO (Reed Solomon)
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 pages 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 Data Length field SHOULD also be performed.

To determine if Single Page FEC or Multi-Page FEC has been used a simple check of the "Additional Data Length (ADL)" field can be used. If the "ADL" is equal to 25, then Single Page FEC is present, anything larger signals Multi-Page FEC.

4.2.1. Single Page FEC

Using the same methods as encoding, an XOR operation is used between the previous and current page (a 25-byte null pad is used as the start). The resulting 25-bytes should be the missing page.

4.2.2. Multi Page FEC

TODO (Reed Solomon)

4.3. FEC Limitations

If more than one page is lost (>1/5 for 5-page messages, >1/10 for 10-page messages) than the error rate of the link is already beyond saving and the application has more issues to deal with.
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)).

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 3: Broadcast Attestation Structure

- **UA DRIP Entity Tag (16 bytes):** The UA DET in byte form (network byte order).
- **Attestation Data (0 to 116 bytes):** Opaque attestation data.
- **Expiration Timestamp by UA (4 bytes):** Timestamp denoting recommended time to trust data to.
- **UA Signature (64 bytes):** Signature over preceding fields using the keypair of the UA.
"Attestation Data" is a field with a maximum of 116-bytes, containing data that the UA is attesting during its flight.

The "Expiration Timestamp" MUST follow the format defined in [F3411]. That is a UNIX timestamp offset by 01/01/2019 00:00:00. An additional offset is then added to push the timestamp a short time into the future to avoid replay attacks. The offset used against the UNIX 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.

6. DRIP Authentication Formats

All formats defined in this section fill 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. Operator ID Signature

The existing ASTM [F3411] Authentication Type 0x2 can be used to send a static Self-Attestation of the Operator.

```
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
+---------------+---------------+---------------+---------------+
|                                                               |
|                            Operator                           |
|                        DRIP Entity Tag                        |
|                                                               |
|                                                               |
|                                                               |
|                     Operator Host Identity                    |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
|                      Expiration Timestamp                     |
+---------------+---------------+---------------+---------------+
|                        Signing Timestamp                      |
+---------------+---------------+---------------+---------------+
```

In this format, the Operator ID is concatenated with the DRIP Entity Tag, followed by the operator host identity, expiration timestamp, and signing timestamp.
Operator Signature

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

Operator Host Identity (32-bytes):
   HI of the Operator.

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

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

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

Figure 4: DRIP Operator ID Signature

6.2.  Message Set Signature

When running under Extended Transports, the existing ASTM [F3411]
Authentication Type 0x3 can be used to sign over the adjacent ASTM
Messages in the Message Pack (Message Type 0xF).

The concatenation of all messages in the Message Pack (excluding Authentication) before signing MUST be in Message Type order and be placed between the UA DRIP Entity Tag and Expiration Timestamp field.
6.3. Specific Authentication Method

For ASTM Specific Authentication Method (Authentication Type 0x5) a special SAM Type field, specified as the first byte of the "Authentication Data / Signature" by [F3411], 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.

```
    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
+---------------+---------------+---------------+---------------+
|   SAM Type    |                                               |
+---------------+                                               |
.                     SAM Authentication Data                     .
.                   .                                            .
+---------------+---------------+---------------+---------------+
```

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 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 the Broadcast Attestation of the Registry (HDA) over the UA. Its structure is defined in [drip-registries] and is as follows:

```
 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
 +---------------+---------------+---------------+---------------+
 |   Link Type   |   DRIP       |   DRIP       |   DRIP       |
 |               | Entity Tag of HDA | Entity Tag of UA | Entity Tag of UA |
 |               |                   |                   |                   |
 |               |                   |                   |                   |
 |               |                   |                   |                   |
 |               |                   |                   |                   |
 |               |                   |                   |                   |
 |               |                   |                   |                   |
 +---------------+---------------+---------------+---------------+
 | Host Identity of UA |
 |                   |
 |                   |
 |                   |
 |                   |
 |                   |
 |                   |
 +---------------+---------------+---------------+---------------+
 | Expiration Timestamp by HDA |
 | Signing Timestamp by HDA |
 +---------------+---------------+---------------+---------------+
 | Signature by HDA |
```
Link Type: (1-byte)
   Multiplexing byte to define different Links being sent.

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 7: Example DRIP HDA-UA Broadcast Attestation

This DRIP format MUST be used in conjunction with the DRIP Manifest
with the hash of the DRIP Link message and other dynamic data (such
as the Location Message (Message Type 0x2)).

6.3.2.1. Link Type

+-------------+------------------+
| Link Type   | Description      |
+-------------+------------------+
| 0x00        | Reserved          |
+-------------+------------------+
| 0x01        | HDA to UA        |
+-------------+------------------+
| 0x02        | RAA to HDA       |
+-------------+------------------+
Table 3

6.3.2.2. 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.
UA DRIP Entity Tag (16 bytes):
The UA DET in byte form (network byte order).

ASTM Message (25 bytes):
Full ASTM Message.
Expiration Timestamp by UA (4 bytes):
Timestamp denoting recommended time to trust data to.

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

Figure 8: Example 4-Message DRIP Wrapper

6.3.3.1. Wrapper Limitations

TODO

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.
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Msg Window (1 byte):**

Variable window size - TODO
UA DRIP Entity Tag (16 bytes):
The UA DET in byte form (network byte order).

Previous Manifest Hash (12 bytes):
See Section 6.3.4.3.

Current Manifest Hash (12 bytes):
See Section 6.3.4.3.

ASTM Message Hash (12 bytes):
Hash of a single full ASTM Message. Multiple hashes should be in Message Type order.

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

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

Figure 9: Example DRIP Manifest

6.3.4.1. Hash Algorithms and Operation

The hash algorithm used for the Manifest Message is the same hash algorithm used in creation of the HHIT that is signing the Manifest.

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

cSHAKE128(ASTM 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. Variable Message Window

Windows of number of ASTM Messages the manifest is applicable over.

(Editors Note: needs better text here and justification of inclusion)

6.3.4.3. 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.

6.3.4.4. 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).

Frame Type (1 byte):
    Multiplexing frame type.

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

Attestation Data (0 to 115 bytes):
    Opaque attestation data.

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

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

Figure 10: Example DRIP Frame
6.3.5.1. 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.X).

Under ASTM Bluetooth 4.X 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.X 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.X). 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 an aircraft send the following Authentication Formats to fulfill the [drip-requirements]:

1. DRIP Link using the Broadcast Attestation of USS 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 CAA (satisfies GEN-3)

2. DRIP Link using the Broadcast Attestation of CAA and the USS (satisfies GEN-3)

3. DRIP Link using the Broadcast Attestation of 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. 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 Frame
2. DRIP Wrapper
3. DRIP Manifest
4. DRIP Link

9. IANA Considerations

This document does not require any actions by IANA.

(Editors Note: needed for Link Types?)

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 aircraft 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 (Editors Note: link to registry draft security considerations here).

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 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


12.2. Informative References

[drip-registries]

[drip-requirements]


Appendix A. Authentication Coloring Scheme

For DRIP there are various Authentication states. The below diagram is the recommended state diagram to determine Authentication status:

TODO

Figure 11: DRIP Authentication Colors/State

Each state has a specific color associated with it:
<table>
<thead>
<tr>
<th>State</th>
<th>Color</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Black</td>
<td>No Authentication being received</td>
</tr>
<tr>
<td>Partial</td>
<td>Gray</td>
<td>Authentication being received but missing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pages</td>
</tr>
<tr>
<td>Unsupported</td>
<td>Brown</td>
<td>Authentication Type/SAM Type of received</td>
</tr>
<tr>
<td></td>
<td></td>
<td>message not supported</td>
</tr>
<tr>
<td>Unverifiable</td>
<td>Yellow</td>
<td>Data needed for verification missing</td>
</tr>
<tr>
<td>Verified</td>
<td>Green</td>
<td>Valid verification results</td>
</tr>
<tr>
<td>Trusted</td>
<td>Blue</td>
<td>Valid verification results and HDA is marked</td>
</tr>
<tr>
<td></td>
<td></td>
<td>as trusted</td>
</tr>
<tr>
<td>Questionable</td>
<td>Orange</td>
<td>Inconsistent verification results</td>
</tr>
<tr>
<td>Unverified</td>
<td>Red</td>
<td>Invalid verification results</td>
</tr>
<tr>
<td>Conflicting</td>
<td>Purple</td>
<td>Inconsistent verification results and HDA is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>marked as trusted</td>
</tr>
</tbody>
</table>

Table 4

Appendix B. Example Authentication Messages

B.1. Authentication Data Only

This is an example of an Authentication Message with 52-bytes of Authentication Data.
B.2. Authentication Data & Additional Data

This example has 52-bytes of Authentication Data and 20-bytes of Additional Data.
B.3. DRIP Link Example

This DRIP Link example includes FEC for a single page.

Page 0:

<table>
<thead>
<tr>
<th>Page Header</th>
<th>Authentication Headers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SAM Type</td>
</tr>
<tr>
<td></td>
<td>Broadcast Attestation</td>
</tr>
</tbody>
</table>

Page 1:

<table>
<thead>
<tr>
<th>Page Header</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Broadcast Attestation</td>
</tr>
</tbody>
</table>

Page 2:

<table>
<thead>
<tr>
<th>Page Header</th>
<th></th>
</tr>
</thead>
</table>
Internet-Draft                auth-formats                 November 2021

<table>
<thead>
<tr>
<th>Page Header</th>
<th>Broadcast Attestation /</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>ADL</td>
</tr>
<tr>
<td>/</td>
<td>Forward Error Correction</td>
</tr>
</tbody>
</table>

Page 7:
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

<table>
<thead>
<tr>
<th>Page Header</th>
<th>Forward Error Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>Null Padding</td>
</tr>
</tbody>
</table>

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
DRIP Entity Tag (DET) for Unmanned Aircraft System Remote Identification (UAS RID)

draft-ietf-drip-rid-13

Abstract

This document describes the use of Hierarchical Host Identity Tags (HHITs) as self-attesting IPv6 addresses and thereby a trustable identifier for use as the Unmanned Aircraft System Remote Identification and tracking (UAS RID). Within the context of RID, HHITs will be called DRIP Entity Tags (DET). HHITs self-attest to the included explicit hierarchy that provides Registrar 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 8 May 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.
1.  Introduction ................................................. 3
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) .................. 5
  3.1.  HHIT prefix .............................................. 6
  3.2.  HHIT Suite IDs .......................................... 6
    3.2.1.  8 bit HIT Suite IDs .................................. 6
  3.3.  The Hierarchy ID (HID) .................................. 6
    3.3.1.  The Registered Assigning Authority (RAA) .......... 6
    3.3.2.  The Hierarchical HIT Domain Authority (HDA) ...... 7
  3.4.  Edward Digital Signature Algorithm for HITs ............ 7
    3.4.1.  HOST_ID ............................................. 7
    3.4.2.  HIT_SUITE_LIST ....................................... 8
  3.5.  ORCHIDs for Hierarchical HITs ........................... 9
    3.5.1.  Adding additional information to the ORCHID ...... 9
    3.5.2.  ORCHID Encoding ..................................... 11
    3.5.3.  ORCHID Decoding ..................................... 12
    3.5.4.  Decoding ORCHIDs for HITv2 ......................... 12
4.  Hierarchical HITs as Remote ID DRIP Entity Tags (DET) ...... 13
  4.1.  Nontransferability of HHITs ............................. 13
  4.2.  Encoding HHITs in CTA 2063-A Serial Numbers ............ 14
  4.3.  Remote ID DET as one class of Hierarchical HITs ........ 15
  4.4.  Hierarchy in ORCHID Generation .......................... 15
  4.5.  DRIP Entity Tag (DET) Registry .......................... 15
  4.6.  Remote ID Authentication using DETs ..................... 16
5.  DRIP Entity Tags (DET) in DNS ................................ 16
6.  Other UTM uses of HHITs beyond DET .......................... 17
7.  DRIP Requirements addressed .................................. 18
8.  DET Privacy .................................................. 18
9.  IANA Considerations ......................................... 19
  9.1.  New IPv6 prefix needed for HHITs ....................... 19
10. Security Considerations ....................................... 20
  10.1.  DET Trust ............................................... 21
  10.2.  Collision risks with DETs ............................... 22
11. References .................................................... 22
1. Introduction

[drip-requirements] describes an Unmanned Aircraft System Remote Identification and tracking (UAS ID) as unique (ID-4), non-spoofable (ID-5), and identify a registry where the ID is listed (ID-2); all within a 20 character identifier (ID-1).

This document describes 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 include explicit hierarchy to enable DNS HHIT queries (Host ID for authentication, e.g. [drip-authentication]) and for EPP Registrar discovery [RFC7484] for 3rd-party identification attestation (e.g. [drip-authentication]).

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

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 is in contrast to using general identifiers (e.g. a Universally Unique IDentifier (UUID) [RFC4122] or device serial number) as the subject in an X.509 [RFC5280] certificate.

In a multi-CA (multi Certificate Authority) 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.

Hierarchical HITs provide self-attestation of the HHIT registry. A HHIT can only be in a single registry within a registry system (e.g. Extensible Provisioning Protocol (EPP) [RFC5730] and DNS).

Hierarchical HITs are valid, though non-routable, IPv6 addresses [RFC8200]. As such, they fit in many ways within various IETF technologies.
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 [drip-requirements]. 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 (Hierarchical HIT Domain Authority):
the 16-bit field that identifies the HHIT Domain Authority under an 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 used in HIP.

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

HIP (Host Identity Protocol)
The origin of HI, HIT, and HHIT, required for DRIP.

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. In particular all the functions referenced from
[NIST.FIPS.202] and [NIST.SP.800-185].

KMAC (KECCAK Message Authentication Code [NIST.SP.800-185]):
A PRF and keyed hash function based on KECCAK.

RAA (Registered Assigning Authority):
The 16 bit field identifying the business or organization that
manages a registry of HDAs.

RVS (Rendezvous Server):
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 HIT space. By adding two levels of hierarchical
administration control, the HHIT provides for device registration/
ownership, thereby enhancing the trust framework for HITs.

HHITs represent the HI in only a 64 bit hash and uses the other 32
bits 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 in Section 3.5.1.

A HHIT is built from the following fields:

* IANA prefix (max 28 bit)
* 32 bit Hierarchy ID (HID)
* 4 (or 8) bit HIT Suite ID
* ORCHID hash (96 - prefix length - Suite ID length bits, e.g. 64)
  See Section 3.5

The Context ID for the ORCHID hash is:
A python script is available for generating HHITs [hhit-gen].

3.1. HHIT prefix

A unique IANA IPv6 prefix, no larger than 28 bit, for HHITs is recommended. It clearly separates the flat-space HIT processing from HHIT processing per Section 3.5.

Without a unique prefix, the first 4 bits of the RRA would be interpreted as the HIT Suite ID per HIPv2 [RFC7401].

3.2. HHIT Suite IDs

The HIT Suite IDs specifies the HI and hash algorithms. Any HIT Suite ID can be used for HHITs. The 8 bit format is supported (only when the first 4 bits are ZERO), but this reduces the ORCHID hash length.

3.2.1. 8 bit HIT Suite IDs

Support for 8 bit HIT Suite IDs is allowed in Section 5.2.10 of [RFC7401], but not specified in how ORCHIDs are generated with these longer OGAs. Section 3.5 provides the algorithmic flexibility, allowing for HDA custom HIT Suite IDs as follows:

<table>
<thead>
<tr>
<th>HIT Suite</th>
<th>Four-bit ID</th>
<th>Eight-bit encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDA Assigned 1</td>
<td>NA</td>
<td>TBD3 (suggested value 0x0E)</td>
</tr>
<tr>
<td>HDA Assigned 2</td>
<td>NA</td>
<td>TBD4 (suggested value 0x0F)</td>
</tr>
</tbody>
</table>

This feature 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.

3.3. The Hierarchy ID (HID)

The Hierarchy ID (HID) provides the structure to organize HITs into administrative domains. HIDs are further divided into 2 fields:

* 16 bit Registered Assigning Authority (RAA)
* 16 bit Hierarchical HIT Domain Authority (HDA)

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) could be an RAA.
The RAA is a 16 bit field (65,536 RAAs) assigned by a numbers management organization, perhaps ICANN’s IANA service. An RAA must provide a set of services to allocate HDAs to organizations. It must 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 HID RVS servers.

As HHITs may be used in many different domains, RAA should be allocated in blocks with consideration on the likely size of a particular usage. Alternatively, different Prefixes can be used to separate different domains of use of HHTs.

This DNS zone may be a PTR for its RAA. It may be a zone in a HHIT specific DNS zone. Assume that the RAA is 100. The PTR record could be constructed:

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

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

An HDA may be an ISP or any third party that takes on the business to provide RVS and other needed services for HIP enabled devices.

The HDA is an 16 bit field (65,536 HDAs per RAA) assigned by an RAA. An HDA should maintain a set of RVS servers that its client HIP-enabled customers use. How this is done and scales to the potentially millions of customers is outside the scope of this document. 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.

### 3.4. Edward Digital Signature Algorithm for HITs

Edwards-Curve Digital Signature Algorithm (EdDSA) [RFC8032] are specified here for use as Host Identities (HIs) per HIPv2 [RFC7401]. Further the HIT_SUITE_LIST is specified as used in [RFC7343].

See Section 3.2 for use of the HIT Suite for this document.

#### 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 5.2.19 of [RFC7401].
For hosts that implement EdDSA as the algorithm, the following EdDSA curves are available:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Curve</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>EdDSA</td>
<td>RESERVED</td>
<td>0</td>
</tr>
<tr>
<td>EdDSA</td>
<td>EdDSA25519</td>
<td>1 [RFC8032]</td>
</tr>
<tr>
<td>EdDSA</td>
<td>EdDSA25519ph</td>
<td>2 [RFC8032]</td>
</tr>
<tr>
<td>EdDSA</td>
<td>EdDSA448</td>
<td>3 [RFC8032]</td>
</tr>
<tr>
<td>EdDSA</td>
<td>EdDSA448ph</td>
<td>4 [RFC8032]</td>
</tr>
</tbody>
</table>

3.4.2. HIT_SUITE_LIST

The HIT_SUITE_LIST parameter contains a list of the supported HIT suite IDs of the Responder. Based on the HIT_SUITE_LIST, the 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, and the relationship between the four-bit ID value used in the OGA ID field and the eight-bit encoding within the HIT_SUITE_LIST ID field is clarified:

<table>
<thead>
<tr>
<th>HIT Suite</th>
<th>4-bit ID</th>
<th>8-bit encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESERVED</td>
<td>0</td>
<td>0x00</td>
</tr>
<tr>
<td>EdDSA/cSHAKE128</td>
<td>TBD2 (suggested value 5)</td>
<td>0x50 (RECOMMENDED)</td>
</tr>
</tbody>
</table>

The following table provides more detail on the above HIT Suite combinations. The input for each generation algorithm is the encoding of the HI as defined in this Appendix.

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).
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 addendum to ORCHIDv2 [RFC7343], as it can produce ORCHIDv2 output.

3.5.1. Adding additional information to the ORCHID

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

<table>
<thead>
<tr>
<th>Index</th>
<th>Hash function</th>
<th>HMAC</th>
<th>Signature algorithm family</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>cSHAKE128</td>
<td>KMAC128</td>
<td>EdDSA</td>
<td>EdDSA HI hashed with cSHAKE128, output is variable</td>
</tr>
</tbody>
</table>

Table 1: HIT Suites
ORCHID := Prefix | OGA ID | Encode_96( Hash )

where:
Prefix : A constant 28-bit-long bitstring value
         (IANA IPv6 assigned).
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.

This addendum will be constructed as follows:
ORCHID := Prefix (p) | Info (n) | OGA ID (o) | Hash (m)

where:
Prefix (p) : An IANA IPv6 assigned prefix (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.
Hash (m) : An extraction function in which output is m bits.
p + n + o + m = 128 bits

With a 28 bit IPv6 Prefix, the remaining 100 bits can be divided in
any manner between the additional information, OGA ID, and the hash
output. Care must be taken in determining the size of the hash
portion, taking into account risks like pre-image attacks. Thus 64
bits as used in Hierarchical HITs may be as small as is acceptable.
Note that if a 8 bit OGA is used, the hash may be 4 bits shorter.
This may result in a greater risk of pre-image attacks and a
corresponding greater need to manage HHIT registration and require
look up of the HI from a trusted source.
3.5.2. ORCHID Encoding

This addendum 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, 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 32 bits for the HID and 4 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 the cSHAKE function call for this addendum is:

\[
cSHAKE128(\text{Input}, \ L, \ "\", \ \text{Context ID})
\]

\[
\begin{align*}
\text{Input} & := \text{Prefix} | \text{Additional Information} | \text{OGA ID} | \text{HOST_ID} \\
\text{L} & := \text{Length in bits of hash portion of ORCHID}
\end{align*}
\]

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 Prefix of /28 and Additional Information of ZERO length):

\[
\text{Hash}[L](\text{Context ID} | \ \text{Input})
\]

\[
\begin{align*}
\text{Input} & := \text{Prefix} | \text{Additional Information} | \text{OGA ID} | \text{HOST_ID} \\
\text{L} & := \text{Length in bits of hash portion of ORCHID} \\
\text{Hash}[L] & := \text{An extraction function in which output is obtained by extracting the middle } L\text{-bit-long bitstring from the argument bitstring.}
\end{align*}
\]

Hierarchical HIT uses the same context as all other HIPv2 HIT Suites as they are clearly separated by the distinct HIT Suite ID.

3.5.2.1. Encoding ORCHIDs for HITv2

This section is included to provide backwards compatibility for ORCHIDv2 [RFC7343] as used for HITv2 [RFC7401].

For HITv2s, the Prefix MUST be 2001:20::/28. Info is length ZERO (not included), and OGA ID is length 4. Thus the HI Hash is length 96. 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 addendum, the decoding of an ORCHID is determined by the Prefix and OGA ID (HIT Suite 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 the HHIT document.

3.5.4. Decoding ORCHIDs for HITv2

This section is included to provide backwards compatibility for ORCHIDv2 [RFC7343] as used for HITv2 [RFC7401].

HITv2s are identified by a Prefix of 2001:20::/28. The next 4 bits are the OGA ID. is length 4. The remaining 96 bits are the HI Hash.
4. Hierarchical HITs as Remote ID DRIP Entity Tags (DET)

Hierarchical HITs are a refinement on the Host Identity Tag (HIT) of HIPv2 [RFC7401]. HHITs require a new Overlay Routable Cryptographic Hash Identifier (ORCHID [RFC7343]) mechanism as described in Section 3.5. HHITs for UAS ID (DET) also use the new EdDSA/SHAKE128 HIT suite defined in Section 3.4 (GEN-2 in [drip-requirements]). This hierarchy, cryptographically embedded within the HHIT, provides the information for finding the UA’s HHIT registry (ID-3 in [drip-requirements]).

ASTM Standard Specification for Remote ID and Tracking [F3411] specifies four UAS ID types:

TYPE-1  A static, manufacturer assigned, hardware serial number per ANSI/CTA-2063-A "Small Unmanned Aerial System Serial Numbers" [CTA2063A].

TYPE-2  A CAA assigned (presumably static) ID.

TYPE-3  A UTM system assigned UUID [RFC4122]. These can be dynamic, but do not need to be.

TYPE-4  Specific Session ID (SSI)

Note that Types 1 - 3 allow for a UAS ID with a maximum length of 20 bytes, the SSI (Type 4) uses the first byte of the ID for the SSI value, thus restricting the UAS ID to a maximum of 19 bytes. The SSI values initially assigned are:

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

ID 2  3GPP - IEEE 1609.2-2016 HashedID8

4.1. Nontransferablity of HHITs

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

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

where:

MFR Code : 4 character code assigned by ICAO.

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

MFR SN : Alphanumeric code (0-9, A-Z except 0 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 HIT Suite ID and ORCHID hash will take 14 characters (see below), leaving 1 character to distinguish encoded DETs from other manufacturer use of CTA 2063-A Serial Numbers.

A character in a CTA 2063-A Serial Number "shall include any combination of digits and uppercase letters, except the letters 0 and 1, but may include all digits". This would allow for a Base34 encoding of the binary HIT Suite ID and ORCHID hash. Although, programatically, 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 68 bits (HIT Suite ID | ORCHID hash) of the HHIT SHALL be left-padded with 2 bits of ZERO. This 70 bit number will be encoded into 14 characters using the digit/letters above. The manufacturer MUST use a Length Code of F (15). The first character after the Length Code MUST be 'Z', followed by the 14 characters of the encoded HIT Suite ID and ORCHID hash.

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:
A mapping service (e.g. DNS) MUST provide a trusted (e.g. via DNSSEC) conversion of the 4 character Manufacturer Code to high-order 60 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. The HHIT DET will still be used in the Authentication Messages.

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 9.

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 rational at the time of developing ORCHID was attacks against these fields are 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.

4.5. DRIP Entity Tag (DET) Registry

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

The two levels of hierarchy within the DET allows for CAAs to have their own Registered Assigning Authority (RAA) for their National Air Space (NAS). Within the RAA, the CAAs can delegate HDAs as needed. There may be other RAAs allowed to operate within a given NAS; this is a policy decision by the CAA.
4.6. Remote ID Authentication using DETs

The EdDSA25519 Host Identity (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 [drip-requirements]). In practice, the Wrapper and Manifest authentication formats in the ASTM Authentication Message (Msg Type 0x2) [drip-authentication] implicitly provide this self-attestation. A lookup service like DNS can provide the HI and registration proof (GEN-3 in [drip-requirements]).

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 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 [drip-requirements]). When a Location/Vector Message (Msg Type 0x1) 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 (DET) in DNS

There are two approaches for storing and retrieving the DET using DNS. These are:

* As FQDNs in the .aero TLD.
* 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 [drip-requirements]). For example, the USS for the HHIT could be found via the following: Assume the RAA is 100 and the HDA is 50. The PTR record is constructed as:
The individual DETs are potentially too numerous (e.g. 60 - 600M) and
dynamic (new DETs every minute for some HDAs) to actually store in a
signed, DNS zone. The HDA SHOULD provide DNS service for its zone
and provide the HHIT detail response. A secure connection (e.g. DNS
over TLS) to the authoritative zone may be a viable alternative to
DNSSEC.

The DET reverse lookup can be a standard IPv6 reverse look up, or it
can leverage off the HHIT structure. Assume a prefix of
2001:30::/28, the RAA is 10 and the HDA is 20 and the DET is:


A DET reverse lookup could be to:

a69e.ad0.1952.a3ad.145.a0.30.2001.20.10.det.arpa.

or:

a3ad1952ad0a69e.5.20.10.30.2001.det.remoteid.aero.

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

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

6. Other UTM uses of HHITs beyond DET

HHITs might be used within the UTM architecture beyond DET (and USS
in UA ID registration and authentication). This includes a GCS HHIT
ID. The GCS may use its HIIT if it is the source of Network Remote
ID for securing the transport and for secure C2 transport (e.g.
[drip-secure-nrid-c2]).

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 (this use is
currently outside the scope). Further, they can be used by FINDER
observers, (e.g. [crowd-sourced-rid]).
7. DRIP Requirements addressed

This document in the previous sections provides the details to solutions for GEN 1 - 3, ID 1 - 5, and REG 1 - 2 as described in [drip-requirements].

8. DET Privacy

There is no expectation of privacy for DETs; it is not part of the Privacy Normative Requirements, Section 4.3.1, of [drip-requirements]. DETs are broadcast in the clear over the open air via Bluetooth and WiFi. 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 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 WiFi 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 UAS RID only starts the changes that need to be implemented.

9. IANA Considerations

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

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

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.

HIT Suite ID:
This document defines the new HIT Suite of EdDSA/cSHAKE with value TBD2 (suggested: 5) (see Section 3.4.2) in the "HIT Suite ID" subregistry of the "Host Identity Protocol (HIP) Parameters" registry.

HIT Suite ID eight-bit encoding:
This document defines the first eight-bit encoded HIT Suite IDs as defined in Section 5.2.10 of [RFC7401]. These are the new HDA domain HIT Suites with values TBD3 and TBD4 (suggested: 0x0E and 0x0F) (see Section 3.2.1). IANA is requested to expand the "HIT Suite ID" subregistry of the "Host Identity Protocol (HIP) Parameters" registry to show both the four-bit and eight-bit values as shown in Section 5.2.10 of [RFC7401] and add these new values that only have eight-bit representations.

9.1. New IPv6 prefix needed for HHITs

Because HHIT format is not compatible with [RFC7343], IANA is requested to allocate a new 28-bit prefix out of the IANA IPv6 Special Purpose Address Block, namely 2001:0000::/23, as per [RFC6890] (suggested: 2001:30::/28).
The 64-bit hash in HHITs presents a real risk of second pre-image cryptographic hash attack Section 10.2. 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 you have $2^{10}$ 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 10.2, it is computationally and economically feasible. Thus 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 the DET using DNS to find the HI for the DET. As such, use of DNSSEC and DNS over TLS by the DET registries is recommended.

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 discussed in Section 4.6.

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 that 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.

10.1. DET Trust

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 actually 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 [crowd-sourced-rid]. Only then does an observer gain full trust in the DET of the UA.

DETs obtained via the Network Remote ID path provides a different approach to trust. Here the UAS SHOULD be securely communicating to the USS (see [drip-secure-nrid-c2]), thus asserting DET trust.
10.2. 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 B 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.

11. References

11.1. Normative References


11.2. Informative References

[cfrg-comment]


[crowd-sourced-rid]


[drip-authentication]

[drip-registries]

[drip-requirements]


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 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.

The project lists "E-Identification" and "E-Registrations" services as to be developed. These services can follow the privacy mechanism proposed in this document. 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 Hierarchical HITs provide 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. 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.

<table>
<thead>
<tr>
<th>Deployed Population</th>
<th>With Collision Risk of .01%</th>
<th>1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Population</td>
<td>4T</td>
<td>42T</td>
</tr>
<tr>
<td>( 2^{96} )</td>
<td>1B</td>
<td>10B</td>
</tr>
<tr>
<td>( 2^{72} )</td>
<td>250M</td>
<td>2.5B</td>
</tr>
<tr>
<td>( 2^{68} )</td>
<td>66M</td>
<td>663M</td>
</tr>
<tr>
<td>( 2^{64} )</td>
<td>16M</td>
<td>160M</td>
</tr>
</tbody>
</table>

Acknowledgments

Dr. Gurtov 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.

Authors’ Addresses
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.
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 & Certificates

4.1. Attestation Structure

All Attestations and Certificates under DRIP share the following format:

```
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
+---------------+---------------+---------------+---------------+
|                                                               |
| Attestor Identity Information                                 |
|                                                               |
+---------------+---------------+---------------+---------------+
| Attestation Data                                           |
|                                                               |
+---------------+---------------+---------------+---------------+
| Expiration Timestamp by Attestor                            |
|                                                               |
+---------------+---------------+---------------+---------------+
| Signing Timestamp by Attestor                               |
|                                                               |
+---------------+---------------+---------------+---------------+
| Signature by Attestor                                      |
```

Wiethuechter, et al. Expires 26 April 2022
Attestor Identity Information: (0, 16-bytes or 120-bytes)
   Field containing Attestor Identity Information in various forms.

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

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

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

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

Figure 1: Attestation Structure

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".
4.2.2. Attestation (A-xy)

(Editors Note: blurb here?)
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.
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.

Length = 104-bytes

Figure 4: DRIP Concise Attestation
The unique size of this attestation (384-bytes) allows for easy detection and subsequent decoding without issue.

Length = 384-bytes

Figure 5: DRIP Mutual Attestation

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.

Length = 176-bytes

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
4.3.2. Concise Certificate (CC-zxy)

Length = 192-bytes

4.3.3. Link Certificate (LC-zxy)
4.3.4. Mutual Certificate (MC-zxy)
5. Registries

5.1. Classes

Under DRIP there are 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?)
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 thr RAA and HDA for the target Registry. The method for discovering a Registry’s RAA and HDA is out of scope here. This allows for the device to generate an HHIT to send to the Registry to be accepted (thus generating the required Host Identity Claim) or denied.

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

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

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

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

```
+--------------------------+-+--------------------------+
| Manufacturer             |<| Manufacturer CA          |
+--------------------------+|+--------------------------+
                  ^            ^
                  v            v
SA-a0a0            A-ma0
                  ^            ^
                  v            v
+--------------------------+
| Aircraft                 |
+--------------------------+
```

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

```
+----------+            +---------+
| Registry | ---------> | HDA DNS |
+----------+  [HIP RR]  +---------+

Coo |   | Aro

v

+----------+
| 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.

```
+----------+
| Registry |
+----------+

^  

| A-ro, A-oaN |

+----------+              +----------+
| Operator | <------------------- | Aircraft |
| +----------+        A-a0aN          +----------+
```

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.
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).
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.

```
+----------+
 | Registry |
 +----------+
```

```
+----------+                        +----------+
 | Operator | ---------------------> | Aircraft |
 +----------+     aN, SA-aNaN        +----------+
```

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.
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).

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


11.2. Informative References


[hhit-registries] Moskowitz, R., Card, S. W., and A. Wiethuechter, "Hierarchical HIT Registries", Work in Progress, Internet-


Authors’ Addresses

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

Email: adam.wiethuechter@axenterprize.com

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

Email: stu.card@axenterprize.com

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

Email: rgm@labs.htt-consult.com