Stream: Internet Engineering Task Force (IETF)
RFC: 9575
Category: Standards Track
Published: June 2024
ISSN: 2070-1721
Authors: A. Wiethuechter, Ed. — *AX Enterprize, LLC*; S. Card — *AX Enterprize, LLC*; R. Moskowitz — *HTT Consulting*

RFC 9575
DRIP Entity Tag (DET) Authentication Formats and Protocols for Broadcast Remote Identification (RID)

Abstract

The Drone Remote Identification Protocol (DRIP), plus trust policies and periodic access to registries, augments Unmanned Aircraft System (UAS) Remote Identification (RID), enabling local real-time assessment of trustworthiness of received RID messages and observed UAS, even by Observers lacking Internet access. This document defines DRIP message types and formats to be sent in Broadcast RID Authentication Messages to verify that attached and recently detached messages were signed by the registered owner of the DRIP Entity Tag (DET) claimed.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The initial regulations (e.g., [FAA-14CFR]) and standards (e.g., [F3411]) for Unmanned Aircraft Systems (UAS) Remote Identification (RID) and tracking do not address trust. However, this is a requirement that needs to be addressed for various different parties that have a stake in the safe operation of National Airspace Systems (NAS). Drone Remote ID Protocol's (DRIP's) goal is to specify how RID can be made trustworthy and available in both Internet and local-only connected scenarios, especially in emergency situations.

UAS often operate in a volatile environment. A small Unmanned Aircraft (UA) offers little capacity for computation and communication. UAS RID must also be accessible with ubiquitous and inexpensive devices without modification. This limits options. Most current small UAS are Internet of Things (IoT) devices even if they are not typically thought of as such. Thus many IoT considerations apply here. Some DRIP work, currently strongly scoped to UAS RID, is likely to be applicable to some other IoT use cases.

Generally, two communication schemes for UAS RID are considered: Broadcast and Network. This document focuses on adding trust to Broadcast RID (Section 3.2 of [RFC9153] and Section 1.2.2 of [RFC9434]). As defined in [F3411] and outlined in [RFC9153] and [RFC9434], Broadcast RID is a one-way Radio Frequency (RF) transmission of Media Access Control (MAC) layer messages over Bluetooth or Wi-Fi.

Senders can make any claims the RID message formats allow. Observers have no standardized means to assess the trustworthiness of message content, nor verify whether the messages were sent by the UA identified therein, nor confirm that the UA identified therein is the one they are visually observing. Indeed, Observers have no way to detect whether the messages were sent by a UA or spoofed by some other transmitter (e.g., a laptop or smartphone) anywhere in direct wireless broadcast range. Authentication is the primary strategy for mitigating this issue.

1.1. DRIP Entity Tag (DET) Authentication Goals for Broadcast RID

ASTM [F3411] Authentication Messages (Message Type 0x2), when used with DET-based formats [RFC9374], enable a high level of trust that the content of other ASTM Messages was generated by their claimed registered source. These messages are designed to provide the Observers with trustworthy and immediately actionable information. Appendix A provides a high-level overview of the various states of trustworthiness that may be used along with these formats.

This authentication approach also provides some error correction (Section 5) as mandated by the United States (US) Federal Aviation Administration (FAA) [FAA-14CFR], which is missing from [F3411] over Legacy Transports (Bluetooth 4.x).

These DRIP enhancements to ASTM's specification for RID and tracking [F3411] further support the important use case of Observers who may be offline at the time of observation.

Section 7 summarizes the DRIP requirements [RFC9153] addressed herein.

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

This document makes use of the terms (CAA, Observer, USS, UTM, etc.) defined in [RFC9153]. Other terms (such as DIME) are from [RFC9434], while others (HI, DET, RAA, HDA, etc.) are from [RFC9374].

In addition, the following terms are defined for this document:

Extended Transports: Use of extended advertisements (Bluetooth 5.x), service info (Wi-Fi Neighbor Awareness Networking (NAN)), or IEEE 802.11 Beacons with the vendor-specific information element as specified in [F3411]. Must use ASTM Message Pack (Message Type 0xF).

Legacy Transports: Use of broadcast frames (Bluetooth 4.x) as specified in [F3411].

Manifest: An immutable list of items being transported (in this specific case over wireless communication).

Observation Session: The period of time during which a given Observer's receiver is processing (even if only intermittently) a series of UAS RID messages, at least some of which use DRIP extensions to [F3411], all nominally from the same UA executing a single flight operation.

Note: For the remainder of this document, *Broadcast Endorsement: Parent, Child* will be abbreviated as *BE: Parent, Child*. For example, *Broadcast Endorsement: RAA, HDA* will be abbreviated as *BE: RAA, HDA*.

3. UAS RID Authentication Background and Procedures

3.1. DRIP Authentication Protocol Description

[F3411] defines Authentication Message framing only. It does not define authentication formats or methods. It explicitly anticipates several signature options but does not fully define those. Annex A1 of [F3411] defines a Broadcast Authentication Verifier Service, which has a heavy reliance on Observer real-time connectivity to the Internet. Fortunately, [F3411] also allows third-party standard Authentication Types using the Type 0x5 Specific Authentication Method (SAM), several of which DRIP defines herein.

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. Per Section 5 of [RFC9434], Authentication formats are needed to relay information for Observers to determine trust. No existing formats (defined in [F3411] or other organizations leveraging this feature) provide functionality to satisfy this goal, resulting in the work reflected in this document.

3.1.1. Usage of DNS

Like most aviation matters, the overall objectives here are security and ultimately safety oriented. Since DRIP depends on DNS for some of its functions, DRIP usage of DNS needs to be protected per best security practices. Many participating nodes will have limited local processing power and/or poor, low-bandwidth QoS paths. Appropriate and feasible security techniques will be highly dependent on the UAS and Observer situation. Therefore, specification of particular DNS security options, transports, etc. is outside the scope of this document (see also Section 9.4).

In DRIP, Observers **MUST** validate all signatures received. This requires that the Host Identity (HI) correspond to a DET [RFC9374]. HI's **MAY** be retrieved from a local cache, if present. The local cache is pre-configured with well-known HIs (such as those of CAA DIMEs) and is further populated by received Broadcast Endorsements (BEs) (Section 3.1.2.1) and DNS lookups (when available).

The Observer **MUST** perform a DNS query, when connectivity allows, to obtain a previously unknown HI. If a query cannot be performed, the message **SHOULD** be cached by the Observer to be validated once the HI is obtained.

A more comprehensive specification of DRIP's use of DNS is out of scope for this document and can be found in [DRIP-REG].

3.1.2. Providing UAS RID Trust

For DRIP, two actions together provide a mechanism for an Observer to trust in UAS RID using Authentication Messages.

First is the transmission of an entire trust chain via Broadcast Endorsements (Section 3.1.2.1). This provides a hierarchy of DIMEs down to and including an individual UA's registration of a claimed DET and corresponding HI (public key). This alone cannot be trusted as having any relevance to the observed UA because replay attacks are trivial.

After an Observer has gathered such a complete key trust chain (from pre-configured cache entries, Broadcast Endorsements received over the air and/or DNS lookups) and verified all of its links, that device can trust that the claimed DET and corresponding public key are properly registered, but the UA has not yet been proven to possess the corresponding private key.

Second is for the UA to prove possession by dynamically signing data that is unique and unpredictable but easily verified by the Observer (Section 3.1.2.2). Verification of this signed data **MUST** be performed by the Observer as part of the received UAS RID information trust assessment (Section 6.4.2).

3.1.2.1. DIME Endorsements of Subordinate DETs

Observers receive DRIP Link Authentication Messages (Section 4.2) containing Broadcast Endorsements by DIMEs of child DET registrations. A series of these Endorsements confirms a path through the hierarchy, defined in [DRIP-REG], from the DET Prefix Owner all the way to an individual UA DET registration.

3.1.2.2. UA-Signed Evidence

To prove possession of the private key associated with the DET, the UA **MUST** sign and send data that is unique and unpredictable but easily validated by the Observer. The data can be an ASTM Message that fulfills the requirements to be unpredictable but easily validated. An Observer receives this UA-signed Evidence from DRIP-based Authentication Messages (Sections 4.3 or 4.4). The Observer must verify the signature (cryptographically, as specified in Section 3.1.1) and validate the signed content (via non-cryptographic means, as specified in Section 6.3).

Whether the content is true is a separate question that DRIP cannot address, but validation performed using observable and/or out-of-band data (Section 6) is possible and encouraged.

3.2. ASTM Authentication Message Framing

The Authentication Message (Message Type 0x2) is unique in the ASTM [F3411] Broadcast standard, as it is the only message that can be larger than the Legacy Transport size. To address this limitation around transport size, it is defined as a set of "pages", each of which fits into a single Legacy Transport frame. For Extended Transports, pages are still used but they are all in a single frame.

> Informational Note: Message Pack (Message Type 0xF) is also larger than the Legacy Transport size but is limited for use only on Extended Transports where it can be supported.

The following subsections are a brief overview of the Authentication Message format defined in [F3411] for better context on how DRIP Authentication fills and uses various fields already defined by ASTM [F3411].

3.2.1. Authentication Page

This document leverages Authentication Type 0x5 (Specific Authentication Method (SAM)) as the principal authentication container, defining a set of SAM Types in Section 4. Authentication Type is encoded in every Authentication Page in the *Page Header*. The SAM Type is defined as a field in the *Authentication Payload* (see Section 3.2.3).

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

Figure 1: Standard ASTM Authentication Message Page

Page Header: (1 octet)

Authentication Type (4 bits) and Page Number (4 bits)

Authentication Payload: (23 octets per page)

Authentication Payload, including headers. Null padded. See Section 3.2.2.

The Authentication Message is structured as a set of pages per Figure 1. There is a technical maximum of 16 pages (indexed 0 to 15) that can be sent for a single Authentication Message, with each page carrying a maximum 23-octet *Authentication Payload*. See Section 3.2.4 for more details. Over Legacy Transports, these messages are "fragmented", with each page sent in a separate Legacy Transport frame.

Either as a single Authentication Message or a set of fragmented Authentication Message Pages, the structure is further wrapped by outer ASTM framing and the specific link framing.

3.2.2. Authentication Payload Field

Figure 2 is the source data view of the data fields found in the Authentication Message as defined by [F3411]. This data is placed into the *Authentication Payload* shown in Figure 1, which spans multiple *Authentication Pages*.

```
 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
+---------------+---------------+---------------+---------------+
|                     Authentication Headers                    |
|                               +---------------+---------------+
|                               |                               |
+---------------+---------------+                               |
.                                                               .
.                Authentication Data / Signature                .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
|      ADL      |                                               |
+---------------+                                               |
.                                                               .
.                       Additional Data                         .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 2: ASTM Authentication Message Fields

Authentication Headers: (6 octets)

As defined in [F3411].

Authentication Data / Signature: (0 to 255 octets)

Opaque authentication data. The length of this payload is known through a field in the *Authentication Headers* (defined in [F3411]).

Additional Data Length (ADL): (1 octet - unsigned)

Length in octets of *Additional Data*. The value of *ADL* is calculated as the minimum of `361 - Authentication Data / Signature Length` and `255`. Only present with *Additional Data*.

Additional Data: (*ADL* octets)

Data that follows the *Authentication Data / Signature* but is not considered part of the *Authentication Data*, and thus is not covered by a signature. For DRIP, this field is used to carry Forward Error Correction (FEC) generated by transmitters and parsed by receivers as defined in Section 5.

3.2.3. SAM Data Format

Figure 3 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
 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                   .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 3: SAM Data Format

SAM Type: (1 octet)

The following SAM Types are allocated to DRIP:

SAM Type	Description
0x01	DRIP Link (Section 4.2)
0x02	DRIP Wrapper (Section 4.3)
0x03	DRIP Manifest (Section 4.4)
0x04	DRIP Frame (Section 4.5)

Table 1: DRIP SAM Types

> Note: ASTM International is the owner of these code points as they are defined in [F3411]. In accordance with Annex 5 of [F3411], the International Civil Aviation Organization (ICAO) has been selected by ASTM as the registrar to manage allocations of these code points. The list is available at [ASTM-Remote-ID].

SAM Authentication Data: (0 to 200 octets)

Contains opaque authentication data formatted as defined by the preceding SAM Type.

3.2.4. ASTM Broadcast RID Constraints

3.2.4.1. Wireless Frame Constraints

A UA has the option to broadcast using Bluetooth (4.x and 5.x), Wi-Fi NAN, or IEEE 802.11 Beacon; see Section 6. With Bluetooth, FAA and other Civil Aviation Authorities (CAA) mandate transmitting simultaneously over both 4.x and 5.x. The same application-layer information defined in [F3411] **MUST** be transmitted over all the physical-layer interfaces performing RID, because Observer transports may be limited. If an Observer can support multiple transports, it should use (display, report, etc.) the latest data regardless of the transport over which that data was received.

Bluetooth 4.x presents a payload-size challenge in that it can only transmit 25 octets of payload per frame, while other transports can support larger payloads per frame. As [F3411] message formats are the same for all media, and their framing was designed to fit within these legacy constraints, Extended Transports cannot send larger messages; instead, the Message Pack format encapsulates multiple messages (each of which fits within these legacy constraints).

By definition Extended Transports provide FEC, but Legacy Transports lack FEC. Thus over Legacy Transports, paged Authentication Messages may suffer the loss of one or more pages. This would result in delivery to the Observer application of incomplete (typically unusable) messages, so DRIP FEC (Section 5) is specified to enable recovery of a single lost page and thereby reduce the likelihood of receiving incompletely reconstructable Authentication Messages. Authentication Messages sent using Extended Transports do not suffer this issue, as the full message (all pages) is sent using a single Message Pack. Furthermore, the use of one-way RF broadcasts prohibits the use of any congestion-control or loss-recovery schemes that require ACKs or NACKs.

3.2.4.2. Paged Authentication Message 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 that the most constrained existing transport can support. Under Broadcast RID, the Extended Transport that can hold the least amount of authentication data is Bluetooth 5.x at 9 pages.

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

1. *Authentication Data / Signature* data **MUST** fit in the first 9 pages (Page Numbers 0 through 8).
2. The *Length* field in the *Authentication Headers* (which encodes the length in octets of *Authentication Data / Signature* only) **MUST NOT** exceed the value of 201. This includes the SAM Type but excludes *Additional Data*.

3.2.4.3. Timestamps

In ASTM [F3411], timestamps are a Unix-style timestamp with an epoch of `2019-01-01 00:00:00 UTC`. For DRIP, this format is adopted for Authentication to keep a common time format in Broadcast payloads.

Under DRIP, there are two timestamps defined: `Valid Not Before (VNB)` and `Valid Not After (VNA)`.

Valid Not Before (VNB) Timestamp: (4 octets)

> Timestamp denoting the recommended time at which to start trusting data. **MUST** follow the format defined in [F3411] as described above. **MUST** be set no earlier than the time the signature (across a given structure) is generated.

Valid Not After (VNA) Timestamp: (4 octets)

Timestamp denoting the recommended time at which to stop trusting data. **MUST** follow the format defined in [F3411] as described above. Has an offset (relative to `VNB`) to avoid replay attacks. The exact offset is not defined in this document. Best practice for identifying an acceptable offset should be used and should take into consideration the UA environment, propagation characteristics of the messages being sent, and clock differences between the UA and Observers. For UA signatures in scenarios typical as of 2024, a reasonable offset would be to set `VNA` approximately 2 minutes after `VNB`; see Appendix B for examples that may aid in tuning this value.

4. DRIP Authentication Formats

All formats defined in this section are contained in the *Authentication Data / Signature* field in Figure 2 and use the Specific Authentication Method (SAM, Authentication Type 0x5). The first octet of the *Authentication Data / Signature* of Figure 2 is used to multiplex among these various formats.

When sending data over a medium that does not have underlying FEC, for example Legacy Transports, then FEC (per Section 5) **MUST** be used.

Examples of Link, Wrapper, and Manifest are shown as part of an operational schedule in Appendix B.2.1.

4.1. UA-Signed Evidence Structure

The *UA-Signed Evidence Structure* (Figure 4) is used by the UA during flight to sign over information elements using the private key associated with the current UA DET. It is encapsulated by the *SAM Authentication Data* field of Figure 3.

This structure is used by the DRIP Wrapper (Section 4.3), Manifest (Section 4.4), and Frame (Section 4.5). DRIP Link (Section 4.2) **MUST NOT** use it, as it will not fit in the ASTM Authentication Message with its intended content (i.e., a Broadcast Endorsement).

```
 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
+---------------+---------------+---------------+---------------+
|                      VNB Timestamp by UA                      |
+---------------+---------------+---------------+---------------+
|                      VNA Timestamp by UA                      |
+---------------+---------------+---------------+---------------+
|                                                               |
.                                                               .
.                            Evidence                           .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
|                               UA                              |
|                        DRIP Entity Tag                        |
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                          UA Signature                         |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 4: Endorsement Structure for UA-Signed Evidence

Valid Not Before (VNB) Timestamp by UA: (4 octets)

See Section 3.2.4.3. Set by the UA.

Valid Not After (VNA) Timestamp by UA: (4 octets)

See Section 3.2.4.3. Set by the UA.

Evidence: (0 to 112 octets)

The *Evidence* field **MUST** be filled in with data in the form of an opaque object specified in the DRIP Wrapper (Section 4.3), Manifest (Section 4.4), or Frame (Section 4.5).

UA DRIP Entity Tag: (16 octets)

This is a DET [RFC9374] currently being used by the UA for authentication; it is assumed to be a Specific Session ID (a type of UAS ID typically also used by the UA in the Basic ID Message).

UA Signature: (64 octets)

Signature over the concatenation of preceding fields (*VNB*, *VNA*, *Evidence*, and *UA DET*) using the keypair of the UA DET. The signature algorithm is specified by the Hierarchical Host Identity Tags (HHIT) Suite ID of the DET.

When using this structure, the UA is minimally self-endorsing its DET. The HI of the UA DET can be looked up by mechanisms described in [DRIP-REG] or by extracting it from a Broadcast Endorsement (see Sections 4.2 and 6.3).

4.2. DRIP Link

This SAM Type (Figure 5) is used to transmit Broadcast Endorsements. For example, the *BE: HDA, UA* is sent (see Section 6.3) as a DRIP Link message.

DRIP Link is important as its contents are used to provide trust in the DET/HI pair that the UA is currently broadcasting. This message does not require Internet connectivity to perform signature verification of the contents when the DIME DET/HI is in the Observer's cache. It also provides the UA HI, when it is filled with a `BE: HDA, UA`, so that connectivity is not required when performing signature verification of other DRIP Authentication Messages.

Various Broadcast Endorsements are sent during each UAS flight operation to ensure that the full Broadcast Endorsement chain is available offline. See Section 6.3 for further details.

```
 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
+---------------+---------------+---------------+---------------+
|                    VNB Timestamp by Parent                    |
+---------------+---------------+---------------+---------------+
|                    VNA Timestamp by Parent                    |
+---------------+---------------+---------------+---------------+
|                                                               |
|                              DET                              |
|                            of Child                           |
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                          HI of Child                          |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
|                              DET                              |
|                           of Parent                           |
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                      Signature by Parent                      |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 5: Broadcast Endorsement / DRIP Link

VNB Timestamp by Parent: (4 octets)

See Section 3.2.4.3. Set by Parent Entity.

VNA Timestamp by Parent: (4 octets)

See Section 3.2.4.3. Set by Parent Entity.

DET of Child: (16 octets)

DRIP Entity Tag of Child Entity.

HI of Child: (32 octets)

Host Identity of Child Entity.

DET of Parent: (16 octets)

DRIP Entity Tag of Parent Entity in DIME Hierarchy.

Signature by Parent: (64 octets)

Signature over concatenation of preceding fields (*VNB*, *VNA*, *DET of Child*, *HI of Child*, and *DET of Parent*) using the keypair of the Parent DET.

This DRIP Authentication Message is used in conjunction with other DRIP SAM Types (such as the Manifest or the Wrapper) that contain data (e.g., the ASTM Location/Vector Message, Message Type 0x2) that is guaranteed to be unique, unpredictable, and easily cross-checked by the receiving device.

A hash of the final link (`BE: HDA on UA`) in the Broadcast Endorsement chain **MUST** be included in each DRIP Manifest (Section 4.4).

Note: The Endorsement that proves a DET is registered **MUST** come from its immediate parent in the registration hierarchy, e.g., a DRIP Identity Management Entity (DIME) [DRIP-REG]. In the definitive hierarchy, the parent of the UA is its HHIT Domain Authority (HDA), the parent of an HDA is its Registered Assigning Authority (RAA), etc. It is also assumed that all DRIP-aware entities use a DET as their identifier during interactions with other DRIP-aware entities.

4.3. DRIP Wrapper

This SAM Type is used to wrap and sign over a list of other [F3411] Broadcast RID messages.

The *Evidence* field of the *UA-Signed Evidence Structure* (Section 4.1) is populated with up to four ASTM Messages [F3411] in a contiguous octet sequence. Only ASTM Message Types 0x0, 0x1, 0x3, 0x4, and 0x5 are allowed and must be in Message Type order as defined by [F3411]. These messages **MUST** include the Message Type and Protocol Version octet and **MUST NOT** include the Message Counter octet (thus are fixed at 25 octets in length).

4.3.1. Wrapped Count and Format Validation

When decoding a DRIP Wrapper on a receiver, a calculation of the number of messages wrapped and a validation **MUST** be performed by using the number of octets (defined as `wrapperLength`) between the *VNA Timestamp by UA* and the *UA DET* as shown in Figure 6.

```
<CODE BEGINS>
if (wrapperLength MOD 25) != 0 {
  return DECODE_FAILURE;
}
wrappedCount = wrapperLength / 25;
if (wrappedCount == 0) {
  // DECODE_SUCCESS; treat as DRIP Wrapper over extended transport
}
else if (wrappedCount > 4) {
  return DECODE_FAILURE;
} else {
  // DECODE_SUCCESS; treat as standard DRIP Wrapper
}

<CODE ENDS>
```

Figure 6: Pseudocode for Wrapper Validation and Number of Messages Calculation

4.3.2. Wrapper over Extended Transports

When using Extended Transports, an optimization to DRIP Wrapper can be made to sign over co-located data in an ASTM Message Pack (Message Type 0xF).

To perform this optimization, the *UA-Signed Evidence Structure* is filled with the ASTM Messages to be in the ASTM Message Pack, the signature is generated, and then the *Evidence* field is cleared, leaving the encoded form shown in Figure 7.

```
 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
+---------------+---------------+---------------+---------------+
|                      VNB Timestamp by UA                      |
+---------------+---------------+---------------+---------------+
|                      VNA Timestamp by UA                      |
+---------------+---------------+---------------+---------------+
|                                                               |
|                              UA                               |
|                        DRIP Entity Tag                        |
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                          UA Signature                         |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 7: DRIP Wrapper over Extended Transports

To verify the signature, the receiver **MUST** concatenate all the messages in the Message Pack (excluding the Authentication Message found in the same Message Pack) in ASTM Message Type order and set the *Evidence* field of the *UA-Signed Evidence Structure* before performing signature verification.

The functionality of a Wrapper in this form is equivalent to Message Set Signature (Authentication Type 0x3) when running over Extended Transports. The Wrapper provides the same format but over both Extended and Legacy Transports, which allows the transports to be similar. Message Set Signature also implies using the ASTM validator system architecture, which depends on Internet connectivity for verification that the receiver may not have at the time an Authentication Message is received. This is something the Wrapper, and all DRIP Authentication Formats, avoid when the UA key is obtained via a DRIP Link Authentication Message.

4.3.3. Wrapper Limitations

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

4.4. DRIP Manifest

This SAM Type is used to create message manifests that contain hashes of previously sent ASTM Messages.

By hashing previously sent messages and signing them, we gain trust in a UA's previous reports without retransmitting them. This is a way to evade the limitation of a maximum of four messages in the Wrapper (Section 4.3.3) and greatly reduce overhead.

Observers **MUST** hash all received ASTM Messages and cross-check them against hashes in received Manifests.

Judicious use of a Manifest enables an entire Broadcast RID message stream to be strongly authenticated with less than 100% overhead relative to a completely unauthenticated message stream (see Section 6.3 and Appendix B).

The *Evidence* field of the *UA-Signed Evidence Structure* (Section 4.1) is populated with 8-octet hashes of [F3411] Broadcast RID messages (up to 11) and three special hashes (Section 4.4.2). All of these hashes **MUST** be concatenated to form a contiguous octet sequence in the *Evidence* field. It is **RECOMMENDED** that the maximum number of ASTM Message Hashes used be 10 (see Appendix B.1.1.2).

The *Previous Manifest Hash*, *Current Manifest Hash*, and *DRIP Link (BE: HDA, UA) Hash* **MUST** always come before the *ASTM Message Hashes* as seen in Figure 8.

An Observer **MUST** use the Manifest to verify each ASTM Message hashed therein that it has previously received. It can do this without having received them all. A Manifest **SHOULD** typically encompass a single transmission cycle of messages being sent; see Section 6.4 and Appendix B.

```
 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
+---------------+---------------+---------------+---------------+
|                       Previous Manifest                       |
|                              Hash                             |
+---------------+---------------+---------------+---------------+
|                       Current Manifest                        |
|                              Hash                             |
+---------------+---------------+---------------+---------------+
|                     DRIP Link (BE: HDA, UA)                   |
|                              Hash                             |
+---------------+---------------+---------------+---------------+
|                                                               |
.                                                               .
.                      ASTM Message Hashes                      .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 8: DRIP Manifest Evidence Structure

Previous Manifest Hash: (8 octets)

Hash of the previously sent Manifest Message.

Current Manifest Hash: (8 octets)

Hash of the current Manifest Message.

DRIP Link (BE: HDA, UA): (8 octets)

Hash of the DRIP Link Authentication Message carrying `BE: HDA, UA` (see Section 4.2).

ASTM Message Hash: (8 octets)

Hash of a single full ASTM Message using hash operations described in Section 4.4.3.

4.4.1. Hash Count and Format Validation

When decoding a DRIP Manifest on a receiver, a calculation of the number of hashes and a validation can be performed by using the number of octets between the *UA DET* and the *VNB Timestamp by UA* (defined as `manifestLength`) such as shown in Figure 9.

```
<CODE BEGINS>
if (manifestLength MOD 8) != 0 {
  return DECODE_FAILURE
}
hashCount = (manifestLength / 8) - 3;

<CODE ENDS>
```

Figure 9: Pseudocode for Manifest Sanity Check and Number of Hashes Calculation

4.4.2. Manifest Ledger Hashes

The following three special hashes are included in all Manifests:

- the *Previous Manifest Hash* links to the previous Manifest.
- the *Current Manifest Hash* is of the Manifest in which it appears.
- the *DRIP Link (BE: HDA, UA) Hash* ties the endorsed UA key to the Manifest chain.

The Previous and Current hashes act as a ledger of provenance for the Manifest chain, which should be traced back if the Observer and UA were within Broadcast RID wireless range of each other for an extended period of time.

The *DRIP Link (BE: HDA, UA)* is included so there is a direct signature by the UA over the Broadcast Endorsement (see Section 4.2). Typical operation would expect that the list of *ASTM Message Hashes* contain nonce-like data. To enforce a binding between the `BE: HDA, UA` and avoid trivial replay attack vectors (see Section 9.1), at least one *ASTM Message Hash* **MUST** be from an [F3411] message that satisfies the fourth requirement in Section 6.3. At least once per Observation Session, the Observer must process that message as specified in Section 6.3.

4.4.3. Hash Algorithms and Operation

The hash algorithm used for the Manifest is the same hash algorithm used in creation of the DET [RFC9374] that is signing the Manifest. This is encoded as part of the DET using the HHIT Suite ID.

DETs that use cSHAKE128 [NIST.SP.800-185] compute the hash as follows:

```
cSHAKE128(ASTM Message, 64, "", "Remote ID Auth Hash")
```

For ORCHID Generation Algorithms (OGAs) other than "5" (EdDSA/cSHAKE128) [RFC9374], use the construct appropriate for the associated hash. For example, the hash for "2" (ECDSA/SHA-384) is computed as follows:

```
Ltrunc( SHA-384( ASTM Message | "Remote ID Auth Hash" ), 8 )
```

When building a Manifest, this process **MUST** be followed:

1. The *Previous Manifest Hash*
   a. is filled with a random nonce if and only if this is the first manifest being generated;
   b. otherwise, it contains the previous manifest's *Current Manifest Hash*.
2. The *Current Manifest Hash* is filled with null.
3. *ASTM Message Hashes* are filled per Section 4.4.3.1 or Section 4.4.3.2.
4. A hash, as defined above in this section, is calculated over the *Previous Manifest Hash*, *Current Manifest Hash* (null filled), and *ASTM Message Hashes*.
5. The *Current Manifest Hash* (null filled) is replaced with the hash generated in Step r.

4.4.3.1. Legacy Transport Hashing

Under this transport, DRIP hashes the full ASTM Message being sent over the Bluetooth Advertising frame. This is the 25-octet object that starts with the Message Type and Protocol Version octet along with the 24 octets of message data. The hash **MUST NOT** include the Message Counter octet.

For paged ASTM Messages (currently only Authentication Messages), all of the pages are concatenated together in Page Number order and hashed as one object.

4.4.3.2. Extended Transport Hashing

Under this transport, DRIP hashes the full ASTM Message Pack (Message Type 0xF) regardless of its content. The hash **MUST NOT** include the Message Counter octet.

4.5. DRIP Frame

This SAM Type is defined to enable use of the *UA-Signed Evidence Structure* (Section 4.1) in the future beyond the previously defined formats (Wrapper and Manifest) by the inclusion of a single octet to signal the format of *Evidence* data (up to 111 octets).

The content format of *Frame Evidence Data* is not defined in this document. Other specifications **MUST** define the contents and register for a *Frame Type*. At the time of publication (2024), there are no defined Frame Types; only an Experimental range has been defined.

Observers **MUST** check the signature of the structure (Section 4.1) per Section 3.1.2.2 and **MAY**, if the specification of *Frame Type* is known, parse the content in *Frame Evidence Data*.

```
 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
+---------------+---------------+---------------+---------------+
|  Frame Type   |                                               |
+---------------+                                               .
.                      Frame Evidence Data                      .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 10: DRIP Frame

Frame Type: (1 octet)

As shown in Figure 10, the *Frame Type* takes the first octet, which leaves 111 octets available for *Frame Evidence Data*. See Section 8.1 for Frame Type allocations.

5. Forward Error Correction

For Broadcast RID, FEC is provided by the lower layers in Extended Transports. The Bluetooth 4.x Legacy Transport does not support FEC, so the following application-level scheme is used with DRIP Authentication to add some FEC. When sending data over a medium that does not have underlying FEC, for example Bluetooth 4.x, this section **MUST** be used.

The Bluetooth 4.x lower layers have error detection but not correction. Any frame in which Bluetooth detects an error is dropped and not delivered to higher layers (in our case, DRIP). Thus it can be treated as an erasure.

DRIP standardizes a single page FEC scheme using XOR parity across all page data of an Authentication Message. This allows the correction of a single erased page in an Authentication Message. If more than a single page is missing, then handling of an incomplete Authentication Message is determined by higher layers.

Other FEC schemes, to protect more than a single page of an Authentication Message or multiple [F3411] Messages, are left for future standardization if operational experience proves it necessary and/or practical.

The data added during FEC is not included in the *Authentication Data / Signature*, but instead in the *Additional Data* field of Figure 2. This may cause the Authentication Message to exceed 9 pages, up to a maximum of 16 pages.

5.1. Encoding

When encoding, two things are **REQUIRED**:

1. The FEC data **MUST** start on a new Authentication Page. To do this, the results of parity encoding **MUST** be placed in the *Additional Data* field of Figure 2 with null padding before it to line up with the next page. The *Additional Data Length* field **MUST** be set to `number of padding octets + number of parity octets`.
2. The *Last Page Index* field (in Page 0) **MUST** be incremented from what it would have been without FEC by the number of pages required for the *Additional Data Length* field, null padding, and FEC.

To generate the parity, a simple XOR operation using the previous parity page and current page is used. Only the 23-octet *Authentication Payload* field of Figure 1 is used in the XOR operations. For Page 0, a 23-octet null pad is used for the previous parity page.

Figure 11 shows an example of the last two pages (out of N) of an Authentication Message using DRIP Single Page FEC. The *Additional Data Length* is set to 33, as there are always 23 octets of FEC data and there are 10 octets of padding in this example to line it up into Page N.

```
Page N-1:
 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 Data / Signature                |
|                                                               |
|               +---------------+---------------+---------------+
|               |    ADL=33     |                               |
+---------------+---------------+                               |
|                         Null Padding                          |
|                                                               |
+---------------+---------------+---------------+---------------+


Page N:
 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  |                                               |
+---------------+                                               |
|                                                               |
|                     Forward Error Correction                  |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 11: Example Single Page FEC Encoding

5.2. Decoding

Frame decoding is independent of the transmit media. However, the decoding process can determine from the first Authentication Page that there may be a Bluetooth 4.x FEC page at the end. The decoding process **MUST** test for the presence of FEC and apply it as follows.

To determine if FEC has been used, a check of the *Last Page Index* is performed. In general, if the *Last Page Index* field is one greater than that necessary to hold *Length* octets of Authentication Data, then FEC has been used. Note that if *Length* octets are exhausted exactly at the end of an Authentication Page, the *Additional Data Length* field will occupy the first octet of the following page. The remainder of this page will be null padded under DRIP to align the FEC to its own page. In this case, the *Last Page Index* will have been incremented once for initializing the *Additional Data Length* field and once for the FEC page, for a total of two additional pages, as in the last row of Table 5.

To decode FEC in DRIP, a rolling XOR is used on each *Authentication Page* received in the current Authentication Message. A Message Counter, outside of the ASTM Message but specified in [F3411], is used to signal a different Authentication Message and to correlate pages to messages. This Message Counter is only a single octet in length, so it will roll over (to 0x00) after reaching its maximum value (0xFF). If only a single page is missing in the Authentication Message the resulting parity octets should be the data of the erased page.

Authentication Page 0 contains various important fields, only located on that page, that help decode the full ASTM Authentication Message. If Page 0 has been reconstructed, the *Last Page Index* and *Length* fields **MUST** be validated by DRIP. The pseudocode in Figure 12 can be used for both checks.

```
<CODE BEGINS>
function decode_check(auth_pages[], decoded_lpi, decoded_length) {
  // check decoded_lpi does not exceed maximum value
  if (decoded_lpi >= 16) {
    return DECODE_FAILURE
  }

  // check that decoded length does not exceed DRIP maximum value
  if (decoded_length > 201) {
    return DECODE_FAILURE
  }

  // grab the page at index where length ends and extract its data
  auth_data = auth_pages[(decoded_length - 17) / 23].data
  // find the index of last auth byte
  last_auth_byte = (17 + (23 * last_auth_page)) - decoded_length

  // look for non-nulls after the last auth byte
  if (auth_data[(last_auth_byte + 2):] has non-nulls) {
    return DECODE_FAILURE
  }

  // check that byte directly after last auth byte is null
  if (auth_data[last_auth_byte + 1] equals null) {
    return DECODE_FAILURE
  }

  // we set our presumed Additional Data Length (ADL)
  presumed_adl = auth_data[last_auth_byte + 1]
  // use the presumed ADL to calculate a presumed
  //Last Page Index (LPI, a field defined in [F3411])
  presumed_lpi = (presumed_adl + decoded_length - 17) / 23

  // check that presumed LPI and decoded LPI match
  if (presumed_lpi not equal decoded_lpi) {
    return DECODE_FAILURE
  }
  return DECODE_SUCCESS
}

<CODE ENDS>
```

Figure 12: Pseudocode for Decode Checks

5.3. FEC Limitations

The worst-case scenario is when the *Authentication Data / Signature* ends perfectly on a page boundary (Page N-1). This means the *Additional Data Length* would start the next page (Page N) and have 22 octets worth of null padding to align the FEC to begin at the start of the next page (Page N+1). In this scenario, an entire page (Page N) is being wasted just to carry the *Additional Data Length*.

6. Requirements and Recommendations

6.1. Legacy Transports

Under DRIP, the goal is to bring reliable receipt of the paged Authentication Message using Legacy Transports. FEC (Section 5) **MUST** be used, per mandated RID rules (for example, the US FAA RID Rules [FAA-14CFR]), when using Legacy Transports (such as Bluetooth 4.x).

Under [F3411], Authentication Messages are transmitted at the static rate (at least every 3 seconds). Any DRIP Authentication Messages containing dynamic data (such as the DRIP Wrapper) **MAY** be sent at the dynamic rate (at least every 1 second).

6.2. Extended Transports

Under the ASTM specification, Extended Transports of RID must use the Message Pack (Message Type 0xF) format for all transmissions. Under Message Pack, ASTM Messages are sent together (in Message Type order) in a single frame (up to 9 single-frame equivalent messages under Legacy Transports). Message Packs are required by [F3411] to be sent at a rate of 1 per second (like dynamic messages).

Message Packs are sent only over Extended Transports that provide FEC. Thus, the DRIP decoders will never be presented with a Message Pack from which a constituent Authentication Page has been dropped; DRIP FEC could never provide benefit to a Message Pack, only consume its precious payload space. Therefore, DRIP FEC (Section 5) **MUST NOT** be used in Message Packs.

6.3. Authentication

To fulfill the requirements in [RFC9153], a UA **MUST**:

1. send DRIP Link (Section 4.2) using the *BE: Apex, RAA* (partially satisfying GEN-3); at least once per 5 minutes. Apex in this context is the DET prefix owner.
2. send DRIP Link (Section 4.2) using the `BE: RAA, HDA` (partially satisfying GEN-3); at least once per 5 minutes.
3. send DRIP Link (Section 4.2) using the `BE: HDA, UA` (satisfying ID-5, GEN-1 and partially satisfying GEN-3); at least once per minute.

4. send any other DRIP Authentication Format (non-DRIP Link) where the UA is dynamically signing data that is guaranteed to be unique, unpredictable, and easily cross checked by the receiving device (satisfying ID-5, GEN-1 and GEN-2); at least once per 5 seconds.

An Observer's receiver must verify the signature (cryptographically, as specified in Section 3.1.1) on each of the 4 messages sent in the operations specified immediately above and the Observer **MUST** validate the signed content (via non-cryptographic means) of the 4th message sent in the last operation immediately above (the non-DRIP Link message).

These transmission, receiver verification, and Observer validation requirements collectively satisfy GEN-3.

6.4. Operational

UAS operation may impact the frequency of sending DRIP Authentication Messages. When a UA dwells at an approximate location, and the channel is heavily used by other devices, less frequent message authentication may be effective (to minimize RF packet collisions) for an Observer. Contrast this with a UA transiting an area, where authenticated messages **SHOULD** be sufficiently frequent for an Observer to have a high probability of receiving an adequate number for validation during the transit.

A **RECOMMENDED** operational configuration (in alignment with Section 6.3) with rationale can be found in Appendix B. It recommends the following once per second:

- Under Legacy Transport:
  - Two sets of those ASTM Messages required by a CAA in its jurisdiction (example: Basic ID, Location/Vector, and System) and one set of other ASTM Messages (example: Self ID, Operator ID)
  - An FEC-protected DRIP Manifest enabling authentication of those ASTM Messages sent
  - A single page of an FEC-protected DRIP Link
- Under Extended Transport:
  - A Message Pack of ASTM Messages (up to 4) and a DRIP Wrapper (per Section 4.3.2)
  - A Message Pack of a DRIP Link

6.4.1. DRIP Wrapper

If DRIP Wrappers are sent, they **MUST** be sent in addition to any required ASTM Messages in a given jurisdiction. An implementation **MUST NOT** send DRIP Wrappers in place of any required ASTM Messages it may encapsulate. Thus, messages within a Wrapper are sent twice: once in the clear and once authenticated within the Wrapper.

The DRIP Wrapper has a specific use case for DRIP-aware Observers. For an Observer plotting Location/Vector Messages (Message Type 0x2) on a map, display of an embedded Location/Vector Message in a DRIP Wrapper can be marked differently (e.g., via color) to signify trust in the Location/Vector data.

6.4.2. UAS RID Trust Assessment

As described in Section 3.1.2, the Observer **MUST** perform validation of the data being received in Broadcast RID. This is because trust in a key is different from trust that an observed UA possesses that key.

A chain of DRIP Links provides trust in a key. A message, signed by that key, containing data that changes rapidly and is not predictable far in advance (relative to typical operational flight times) but that can be validated by Observers, provides trust that some agent with access to that data also possesses that key. If the validation involves correlating physical world observations of the UA with claims in that data, then the probability is high that the observed UA is (or is collaborating with or observed in real time by) the agent with the key.

At least once per Observation session, after signature verification of any DRIP Authentication Message containing UAS RID information elements (e.g., DRIP Wrapper, Section 4.3), the Observer must use other sources of information to correlate against and perform validation (as specified in Section 6.3). An example of another source of information is a visual confirmation of the UA position.

When correlation of these different data streams does not match in acceptable thresholds, the data **MUST** be rejected as if the signature failed to validate. Acceptable threshold limits and what happens after such a rejection are out of scope for this document.

7. Summary of Addressed DRIP Requirements

The following requirements as defined in [RFC9153] are addressed in this document:

ID-5: Non-spoofability

Addressed using the DRIP Wrapper (Section 4.3), DRIP Manifest (Section 4.4), or DRIP Frame (Section 4.5).

GEN-1: Provable Ownership

Addressed using the DRIP Link (Section 4.2) and DRIP Wrapper (Section 4.3), DRIP Manifest (Section 4.4), or DRIP Frame (Section 4.5).

GEN-2: Provable Binding

Addressed using the DRIP Wrapper (Section 4.3), DRIP Manifest (Section 4.4) or DRIP Frame (Section 4.5).

GEN-3: Provable Registration

Addressed using the DRIP Link (Section 4.2).

8. IANA Considerations

8.1. IANA DRIP Registry

IANA has created the "DRIP SAM Types" and "DRIP Frame Types" registries within the "Drone Remote ID Protocol" registry group.

DRIP SAM Types:
: This registry is a mirror for SAM Types containing the subset of allocations used by DRIP Authentication Messages. Future additions **MUST** be done through ASTM's designated registrar, which is ICAO [ASTM-Remote-ID] at the time of publication of this RFC (2024). The registration procedure for DRIP (only) SAM Types is Standards Action [RFC8126]. Requests for new DRIP SAM Type registrations will be coordinated by IANA and the ASTM-designated registrar of all SAM Types before being documented in Standards Track RFCs. The following values have been allocated to the IETF:

SAM Type	Name	Description
0x01	DRIP Link	Format to hold Broadcast Endorsements
0x02	DRIP Wrapper	Authenticate full ASTM Messages
0x03	DRIP Manifest	Authenticate hashes of ASTM Messages
0x04	DRIP Frame	Format for future DRIP authentication

Table 2: DRIP SAM Types

DRIP Frame Types:
: This 8-bit value registry is for Frame Types in DRIP Frame Authentication Messages. Future additions to this registry are to be made through Expert Review (Section 4.5 of [RFC8126]) for values 0x01 to 0x9F and First Come First Served (Section 4.4 of [RFC8126]) for values 0xA0 to 0xEF. The following values are defined:

Frame Type	Name	Description
0x00	Reserved	Reserved
0x01 - 0xEF	Unassigned	
0xF0-0xFF	Experimental	Reserved for Experimental Use

Table 3: DRIP Frame Types

Criteria that should be applied by the designated experts includes determining whether the proposed registration duplicates existing functionality and whether the registration description is clear and fits the purpose of this registry.

Registration requests **MUST** be sent to drip-reg-review@ietf.org and be evaluated by one or more designated experts within a three-week review period. Within that review period, the designated experts will either approve or deny the registration request, and communicate their decision to the review list and IANA. Denials should include an explanation and, if applicable, suggestions to successfully register the DRIP Frame Type.

Registration requests that are undetermined for a period longer than 28 days can be brought to the IESG's attention for resolution.

9. Security Considerations

9.1. Replay Attacks

[F3411] (regardless of transport) lacks replay protection, as it more fundamentally lacks fully specified authentication. An attacker can spoof the UA sender MAC address and UAS ID, replaying (with or without modification) previous genuine messages, and/or crafting entirely new messages. Using DRIP in [F3411] Authentication Message framing enables verification that messages were signed with registered keys, but when naively used may be vulnerable to replay attacks. Technologies such as Single Emitter Identification can detect such attacks, but they are not readily available and can be prohibitively expensive, especially for typical Observer devices such as smartphones.

Replay attack detection using DRIP requires Observer devices to combine information from multiple Broadcast RID messages and from sources other than Broadcast RID. A complete chain of Link messages (Section 4.2) from an Endorsement root of trust to the claimed sender must be collected and verified by the Observer device to provide trust in a key. Successful signature verification, using that public key, of a Wrapper (Section 4.3) or Manifest (Section 4.4) message, authenticating content that is nonce-like (see below), provides trust that the sender actually possesses the corresponding private key.

The term "nonce-like" describes data that is unique, changes frequently, is not accurately predictable long in advance, and is easily validated (i.e., can be checked quickly at low computational cost using readily available data) by the Observer. A Location/Vector Message is an obvious choice. This is described in Section 3.1.2.2 and Section 6.3 (requirement 4). A Location/Vector Message [F3411] reporting precise UA position and velocity at a precise and very recent time can be checked by the Observer against visual observations of UA within both RF and Visual Line of Sight.

For normative specification of the foregoing, see Sections 3.1.2 and 6.4.2. As non-normative clarification, the requirements are satisfied as follows:

The public key corresponding to a given DET (i.e., the key attested in the DRIP Link (`BE: HDA, UA`) that is the last link in the relevant chain of DRIP Links) is used by an Observer's receiver to try to authenticate some signed message.

If the signature check passes,

> *and* the message was a Wrapper or Manifest,
>
> *and* the wrapped or manifested message contained content that was nonce-like,
>
> *and* the Observer validated that content by non-cryptographic means (e.g., if the wrapped or manifested message was a Location/Vector Message and the UA was visually observed to be in approximately the claimed location at the reported time),

only then can the Observer trust that the currently observed sending UA actually possesses the corresponding private key (and thus owns the corresponding DET).

Messages that pass signature verification with trusted keys could still be replays if they contain only static information (e.g., Broadcast Endorsements (Section 4.2), [F3411] Basic ID, or [F3411] Operator ID), or information that cannot be readily validated (e.g., [F3411] Self-ID). Replay of Link messages is harmless (unless sent so frequently as to cause RF data link congestion) and indeed can increase the likelihood of an Observer device collecting an entire trust chain in a short time window. Replay of other messages ([F3411] Basic ID, [F3411] Operator ID, or [F3411] Self-ID) remains a vulnerability, unless they are combined with messages containing nonce-like data ([F3411] Location/Vector or [F3411] System) in a Wrapper or Manifest. For specification of this last requirement, see Section 4.4.2.

9.2. Wrapper vs Manifest

Implementations have a choice of using Wrapper (Section 4.3), Manifest (Section 4.4), or a combination to satisfy the fourth requirement in Section 6.3.

Wrapper is an attached signature on the full content of one or more [F3411] messages, providing strong authentication. Wrapper is an attached signature of the full content of one or more [F3411] messages, providing strong authentication. However, the size limitation means it cannot support such signatures over other Authentication Messages; thus, it cannot provide a direct binding to any part of the trust chain (Sections 3.1.2 and 6.4.2).

Manifest explicitly provides the binding of the last link in the trust chain (with the inclusion of the hash of the Link containing `BE: HDA, UA`). The use of hashes and their length also allows for a larger number (11 vs 4) of [F3411] messages to be authenticated, making it more efficient compared to the Wrapper. However, the detached signature requires additional Observer overhead in storing and comparing hashes of received messages (some of which may not be received) with those in a Manifest.

Appendix B contains a breakdown of frame counts and an example of a schedule using both Manifest and Wrapper. Typical operation may see (as an example) 2x Basic ID, 2x Location/Vector, 2x System, 1x Operator ID and 1x Self ID broadcast per second to comply with jurisdiction mandates. Each of these messages is a single frame in size. A Link message is 8 frames long (including FEC). This is a base frame count of **16 frames**.

When Wrapper is used, up to four of the previous messages (except the Link) can be authenticated. For this comparison, we will sign all the messages we can in two Wrappers. This results in *20 frames* (with FEC). Due to size constraints, the Link message is left unauthenticated. The total frame count using Wrappers is **36 frames** (wrapper frame count + base frame count).

When Manifest is used, up to 10 previous messages can be authenticated. For this example, all messages (8) are hashed (including the Link) resulting in a single Manifest that is *9 frames* (with FEC). The total frame count using Manifest is **25 frames** (manifest frame count + base frame count).

9.3. VNA Timestamp Offsets for DRIP Authentication Formats

Note the discussion of VNA Timestamp offsets here is in the context of the DRIP Wrapper (Section 4.3), DRIP Manifest (Section 4.4), and DRIP Frame (Section 4.5). For DRIP Link (Section 4.2), these offsets are set by the DIME and have their own set of considerations in [DRIP-REG].

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

9.4. DNS Security in DRIP

As stated in Section 3.1 specification of particular DNS security options, transports, etc. is outside the scope of this document. The main specification for DNS operations in DRIP [DRIP-REG] will specify applicable best common security practices (e.g., from [RFC9364]).

10. References

10.1. Normative References

[F3411] ASTM International, "Standard Specification for Remote ID and Tracking", ASTM F3411-22A, DOI 10.1520/F3411-22A, July 2022, <https://www.astm.org/f3411-22a.html>.

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

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

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

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

[RFC9374] Moskowitz, R., Card, S., Wiethuechter, A., and A. Gurtov, "DRIP Entity Tag (DET) for Unmanned Aircraft System Remote ID (UAS RID)", RFC 9374, DOI 10.17487/RFC9374, March 2023, <https://www.rfc-editor.org/info/rfc9374>.

[RFC9434] Card, S., Wiethuechter, A., Moskowitz, R., Zhao, S., Ed., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Architecture", RFC 9434, DOI 10.17487/RFC9434, July 2023, <https://www.rfc-editor.org/info/rfc9434>.

10.2. Informative References

[ASTM-Remote-ID] International Civil Aviation Organization (ICAO), "Remote ID Number Registration", December 2023, <https://www.icao.int/airnavigation/IATF/Pages/ASTM-Remote-ID.aspx>.

[DRIP-REG] Wiethuechter, A., Ed. and J. Reid, "DRIP Entity Tag (DET) Identity Management Architecture", Work in Progress, Internet-Draft, draft-ietf-drip-registries-16, 31 May 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-drip-registries-16>.

[FAA-14CFR] Federal Aviation Administration (FAA), "Remote Identification of Unmanned Aircraft", January 2021, <https://www.govinfo.gov/content/pkg/FR-2021-01-15/pdf/2020-28948.pdf>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC9364] Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <https://www.rfc-editor.org/info/rfc9364>.

Appendix A. Authentication States

ASTM Authentication has only three states: None, Invalid, and Valid. This is because, under ASTM, the authentication is done by an external service hosted somewhere on the Internet so it is assumed an authoritative response will always be returned. This classification becomes more complex in DRIP with the support of "offline" scenarios where an Observer does not have

Internet connectivity. With the use of asymmetric cryptography, this means that the public key (PK) must somehow be obtained. [DRIP-REG] provides more detail on how these keys are stored on the DNS and how DRIP Authentication Messages can be used to send PKs over Broadcast RID.

There are a few keys of interest: the PK of the UA and the PKs of relevant DIMEs. This document describes how to send the PK of the UA over the Broadcast RID messages. The keys of DIMEs are sent over Broadcast RID using the same mechanisms (see Sections 4.2 and 6.3) but **MAY** be sent at a far lower rate due to potential operational constraints (such as saturation of limited bandwidth). As such, there are scenarios where part of the key-chain may be unavailable at the moment a full Authentication Message is received and processed.

The intent of this informative appendix is to recommend a way to classify these various states and convey it to the user through colors and state names/text. These states can apply to either a single Authentication Message, a DET (and its associated public key), and/or a sender.

Table 4 briefly describes each state and recommends an associated color.

State	Color	Details
None	Black	No Authentication has been or is being received (as yet)
Partial	Gray	Authentication being received but missing pages
Unsupported	Brown	Authentication Type / SAM Type of received message not supported
Unverifiable	Yellow	Data needed for signature verification is missing
Verified	Green	Valid signature verification and content validation
Trusted	Blue	Evidence of Verified and DIME is marked as only registering DETs for trusted entities
Unverified	Red	Invalid signature verification or content validation
Questionable	Orange	Evidence of both"Verified and Unverified for the same claimed sender
Conflicting	Purple	Evidence of both Trusted and Unverified for the same claimed sender

Table 4: Authentication State Names, Colors, and Descriptions

A.1. None: Black

The default state where authentication information has not yet been received and is not currently being received.

A.2. Partial: Gray

A pending state where Authentication Pages are being received, but a full Authentication Message has yet to be compiled.

A.3. Unsupported: Brown

A state wherein authentication data is being or has been received but cannot be used, as the Authentication Type or SAM Type is not supported by the Observer.

A.4. Unverifiable: Yellow

A pending state where a full Authentication Message has been received but other information, such as public keys to verify signatures, is missing.

A.5. Verified: Green

A state where all Authentication Messages that have been received from that claimed sender up to that point pass signature verification and the requirement of Section 6.4.2 has been met.

A.6. Trusted: Blue

A state where all Authentication Messages that have been received from that claimed sender up to that point have passed signature verification, the requirement of Section 6.4.2 has been met, and the public key of the sending UA has been marked as trusted.

The sending UA key will have been marked as trusted if the relevant DIMEs only register DETs (of subordinate DIMEs, UAS operators, and UA) that have been vetted as per their published registration policies, and those DIMEs have been marked, by the owner (individual or organizational) of the Observer, as per that owner's policy, as trusted to register DETs only for trusted parties.

A.7. Questionable: Orange

A state where there is a mix of Authentication Messages received that are Verified (Appendix A.5) and Unverified (Appendix A.8).

State transitions from Verified to Questionable if a subsequent message fails verification, so it would have otherwise been marked Unverified. State transitions from Unverified to Questionable if a subsequent message passes verification or validation, so it would otherwise have been marked Verified. It may transition from either of those states upon mixed results on the requirement of Section 6.4.2.

A.8. Unverified: Red

A state where all Authentication Messages that have been received from that claimed sender up to that point failed signature verification or the requirement of Section 6.4.2.

A.9. Conflicting: Purple

A state where there is a mix of Authentication Messages received that are Trusted (Appendix A.6) and Unverified (Appendix A.8) and the public key of the aircraft is marked as trusted.

State transitions from Trusted to Conflicting if a subsequent message fails verification, so it would have otherwise been marked Unverified. State transitions from Unverified to Conflicting if a subsequent message passes verification or validation and policy checks, so it would otherwise have been marked Trusted. It may transition from either of those states upon mixed results on the requirement of Section 6.4.2.

Appendix B. Operational Recommendation Analysis

The recommendations in Section 6.4 may seem heavy-handed and specific. This informative appendix lays out the math and assumptions made that resulted in those recommendations and provides an example.

In all jurisdictions known to the authors of this document as of its publication (2024), at least the following ASTM Messages are required to be transmitted at least once per second:

- Basic ID (0x1)
- Location (0x2)
- System (0x4)

Europe also requires:

- Operator ID Message (0x5)

Japan requires not one but two Basic ID messages:

- one carrying a manufacturer assigned serial number
- one carrying a CAA assigned registration number

Japan also requires:

- Authentication (0x2) using their own unique scheme

In all jurisdictions, one further message is optional, but highly recommended for carriage of additional information on the nature of the emergency if the Emergency value is sent in the Operational Status field of the Location/Vector Message:

- Self ID (0x3)

To improve the likelihood of successful timely receipt of regulator required RID data elements, most implementations send at a higher rate, whether by repeating the same messages in the same one second interval, or updating message content and sending messages more frequently than once per second. Excessive sending rate, however, could congest the RF spectrum, leading to collisions and counter-intuitively actually reducing the likelihood of timely receipt of RID data.

B.1. Page Counts vs Frame Counts

There are two formulas to determine the number of Authentication Pages required. The following formula is for Wrapper:

```
<CODE BEGINS>
wrapper_struct_size = 89 + (25 * num_astm_messages)
wrapper_page_count = ceiling((wrapper_struct_size - 17) / 23) + 1

<CODE ENDS>
```

The following formula is for Manifest:

```
<CODE BEGINS>
manifest_struct_size = 89 + (8 * (num_astm_hashes + 3))
manifest_page_count = ceiling((manifest_struct_size - 17) / 23) + 1

<CODE ENDS>
```

A similar formula can be applied to Links, as they are of fixed size:

```
<CODE BEGINS>
link_page_count = ceiling((137 - 17) / 23) + 1 = 7

<CODE ENDS>
```

Comparing Wrapper and Manifest Authentication Message page counts against total frame counts, we have the following:

ASTM Messages	Wrapper (w/FEC)	Manifest (w/FEC)	ASTM Messages + Wrapper (w/FEC)	ASTM Messages + Manifest (w/FEC)
0	5 (6)	6 (7)	5 (6)	6 (7)
1	6 (7)	6 (7)	7 (8)	7 (8)
2	7 (8)	6 (7)	9 (10)	8 (9)
3	8 (9)	7 (8)	11 (12)	10 (11)

ASTM Messages	Wrapper (w/FEC)	Manifest (w/FEC)	ASTM Messages + Wrapper (w/FEC)	ASTM Messages + Manifest (w/FEC)
4	9 (10)	7 (8)	13 (14)	11 (12)
5	N/A	7 (8)	N/A	12 (13)
6	N/A	8 (9)	N/A	14 (15)
7	N/A	8 (9)	N/A	15 (16)
8	N/A	8 (9)	N/A	16 (17)
9	N/A	9 (10)	N/A	18 (19)
10	N/A	9 (10)	N/A	19 (20)
11	N/A	9 (11)	N/A	20 (22)

Table 5: Page and Frame Counts

Link shares the same page counts as Manifest with 5 ASTM Messages.

B.1.1. Special Cases

B.1.1.1. Zero ASTM Messages

Zero ASTM Messages (see Table 5) is where Extended Wrapper (Section 4.3.2) without FEC is used in Message Packs. With a maximum of nine "message slots" in a Message Pack, an Extended Wrapper fills five slots; thus it can authenticate up to four ASTM Messages co-located in the same Message Pack.

B.1.1.2. Eleven ASTM Messages

Eleven ASTM Messages (see Table 5) is where a Manifest with FEC invokes the situation mentioned in Section 5.3.

Eleven is the maximum number of ASTM Message Hashes that can be supported resulting in 14 total hashes. This completely fills the *Evidence* field of the *UA-Signed Evidence Structure* making its total size 200 octets. This fits on exactly 9 Authentication Pages (`(201 - 17) / 23 == 8`), so when the ADL is added, it is placed on the next page (Page 10). Per rule 1 in Section 5.1, this means that all of Page 10 is null padded (expect the ADL octet) and FEC data fills Page 11, resulting in a plus-two page count when FEC is applied.

This drives the recommendation is Section 4.4 to only use up to 10 ASTM Message Hashes, not 11.

B.2. Full Authentication Example

This example (Figure 13) is focused on showing that 100% of ASTM Messages can be authenticated over Legacy Transports with up to 125% overhead in Authentication Pages. Extended Transports are not shown in this example, because, for those, Authentication with DRIP is achieved using Extended Wrapper (Section 4.3.2). Two ASTM Message Packs are sent in a given cycle: one containing up to four ASTM Messages and an Extended Wrapper (authenticating the pack), and one containing a Link message with a Broadcast Endorsement and up to two other ASTM Messages.

This example transmit scheme covers and meets every known regulatory case enabling manufacturers to use the same firmware worldwide.

```
+---------------------------------------------------------+
|                      Frame Slots                        |
| 00 - 04           | 05 - 07      | 08 - 16 | 17       |
+-------------------+--------------+---------+--------+
| {A|B|C|D},V,S,I,O | {A|B|C|D},V,S | M[0,8]  | L/W[0] |
+-------------------+--------------+---------+--------+
| {A|B|C|D},V,S,I,O | {A|B|C|D},V,S | M[0,8]  | L/W[1] |
+-------------------+--------------+---------+--------+
| {A|B|C|D},V,S,I,O | {A|B|C|D},V,S | M[0,8]  | L/W[2] |
+-------------------+--------------+---------+--------+
| {A|B|C|D},V,S,I,O | {A|B|C|D},V,S | M[0,8]  | L/W[3] |
+-------------------+--------------+---------+--------+
| {A|B|C|D},V,S,I,O | {A|B|C|D},V,S | M[0,8]  | L/W[4] |
+-------------------+--------------+---------+--------+
| {A|B|C|D},V,S,I,O | {A|B|C|D},V,S | M[0,8]  | L/W[5] |
+-------------------+--------------+---------+--------+
| {A|B|C|D},V,S,I,O | {A|B|C|D},V,S | M[0,8]  | L/W[6] |
+-------------------+--------------+---------+--------+
| {A|B|C|D},V,S,I,O | {A|B|C|D},V,S | M[0,8]  | L/W[7] |
+-------------------+--------------+---------+--------+

A = Basic ID Message (0x0) ID Type 1
B = Basic ID Message (0x0) ID Type 2
C = Basic ID Message (0x0) ID Type 3
D = Basic ID Message (0x0) ID Type 4
V = Location/Vector Message (0x1)
I = Self ID Message (0x3)
S = System Message (0x4)
O = Operator ID Message (0x5)

L[y,z] = DRIP Link Authentication Message (0x2)
W[y,z] = DRIP Wrapper Authentication Message (0x2)
M[y,z] = DRIP Manifest Authentication Message (0x2)
  y = Start Page
  z = End Page

# = Empty Frame Slot
* = Message in DRIP Manifest Authentication Message
```

Figure 13: Example of a Fully Authenticated Legacy Transport Transmit Schedule

Every common required message (Basic ID, Location/Vector, and System) is sent twice along with Operator ID and Self ID in a single second. The Manifest is over all messages (8) in slots `00 - 04` and `05 - 07`.

In two seconds, either a Link or Wrapper is sent. The content and order of Links and Wrappers runs as follows:

```
Link: HDA on UA
Link: RAA on HDA
Link: HDA on UA
Link: Apex on RAA
Link: HDA on UA
Link: RAA on HDA
Link: HDA on UA
Wrapper: Location/Vector (0x1), System (0x4)
Link: HDA on UA
Link: RAA on HDA
Link: HDA on UA
Link: Apex on RAA
Link: HDA on UA
Link: RAA on HDA
Link: HDA on UA
Wrapper: Location/Vector (0x1), System (0x4)
Link: IANA on UAS RID Apex
```

After perfect receipt of all messages for a period of 8 seconds, all messages sent during that period have been authenticated using the Manifest (except for the Authentication Messages themselves). Within 136 seconds, the entire Broadcast Endorsement chain is received and can be validated. Interspersed in this schedule are 4 messages sent not only in their basic [F3411] form, but also in DRIP Wrapper messages, together with their attached signatures, to defend against the possibility of attack against the detached signatures provided by the Manifest messages.

B.2.1. Raw Example

Assuming the following DET and HI:

```
2001:3f:fe00:105:a29b:3ff4:2226:c04e
b5fef530d450dedb59ebafa18b00d7f5ed0ac08a81975034297bea2b00041813
```

The following ASTM Messages are to be sent in a single second:

```
0240012001003ffe000105a29b3ff42226c04e000000000000
12000000000000000000000000000000000000000060220000
32004578616d706c652053656c6620494400000000000000000
4200000000000000000001000000000000000000010ea510900
52004578616d706c65204f70657261746f7220494400000000
0240012001003ffe000105a29b3ff42226c04e000000000000
12000000000000000000000000000000000000000060220000
4200000000000000000001000000000000000000010ea510900
```

This is a Link with FEC that would be spread out over 8 seconds:

```
2250078910ea510904314b8564b17e66662001003ffe000105
2251a29b3ff42226c04eb5fef530d450dedb59ebafa18b00d7
2252f5ed0ac08a81975034297bea2b000418132001003ffe00
22530105b82bf1c99d87273103fc83f6ecd9b91842f205c222
2254dd71d8e165ad18ca91daf9299a73eec850c756a7e9be46
2255f51dddfa0f09db7bfdde14eec07c7a6dd1061c1d5ace94
2256d9ad97940d280000000000000000000000000000000000
2257a03b0f7a6feb0d198167045058cfc49f73129917024d22
```

This is a Wrapper with FEC that would be spread out over 8 seconds:

```
2250078b10ea510902e0dd7c6560115e671200000000000000
22510000000000000000000000000000602200004200000000
225200000000001000000000000000000010ea5109002001003f
2253fe000105a29b3ff42226c04ef0ecad581a030ca790152a
22542f08df5762a463e24a742d1c530ec977bbe0d113697e2b
2255b909d6c7557bdaf1227ce86154b030daadda4a6b8474de
22569a62f6c375020826000000000000000000000000000000
2257f5e8eebcb04f8c2197526053e66c010d5d7297ff7c1fe0
```

This is the Manifest with FEC sent in the same second as the original messages:

```
225008b110ea510903e0dd7c6560115e670000000000000000
2251d57594875f8608b4d61dc9224ecf8b842bd4862734ed01
22522ca2e5f2b8a3e61547b81704766ba3eeb651be7eafc928
22538884e3e28a24fd5529bc2bd4862734ed012ca2e5f2b8a3
2254e61547b81704766ba3eeb62001003ffe000105a29b3ff4
22552226c04efb729846e7d110903797066fd96f49a77c5a48
2256c4c3b330be05bc4a958e9641718aaa31aeabad368386a2
22579ed2dce2769120da83edbcdc0858dd1e357755e7860317
2258e7c06a5918ea62a937391cbfe0983539de1b2e688b7c83
```


Acknowledgments

The authors acknowledge the following individuals:

- Ryan Quigley, James Mussi, and Joseph Stanton of AX Enterprize, LLC for early prototyping to find holes in earlier drafts of this specification.
- Carsten Bormann for the simple approach of using bit-column-wise parity for erasure (dropped frame) FEC.
- Soren Friis for pointing out that Wi-Fi implementations would not always give access to the MAC Address, as was 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).
- Gabriel Cox (chair of the working group that produced [F3411]) for reviewing the specification for the SAM Type request as the ASTM Designated Expert.
- Mohamed Boucadair (Document Shepherd) for his many patches and comments.

- Eric Vyncke (DRIP AD) for his guidance regarding the document's path to publication.

The authors also thank the following reviewers:

- Rick Salz (secdir)
- Matt Joras (genart)
- Di Ma (dnsdir)
- Gorry Fairhurst (tsvart)
- Carlos Bernardos (intdir)
- Behcet Sarikaya (iotdir)
- Martin Duke (IESG)
- Roman Danyliw (IESG)
- Murray Kucherawy (IESG)
- Erik Kline (IESG)
- Warren Kumari (IESG)
- Paul Wouters (IESG)

Authors' Addresses

Adam Wiethuechter (EDITOR)
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