Workgroup: DRIP Working Group
Internet-Draft: draft-ietf-drip-auth-29
Published: 15 February 2023
Intended Status: Standards Track
Expires: 19 August 2023
Authors: A. Wiethuechter (Editor) S. Card
AX Enterprize, LLC AX Enterprize, LLC
R. Moskowitz
HTT Consulting

DRIP Entity Tag Authentication Formats & Protocols for Broadcast Remote ID

Abstract

This document describes how to add trust into the Broadcast Remote ID (RID) specification discussed in the DRIP Architecture; first trust in the RID ownership and second in the source of the RID messages. The document defines message types and associated formats (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

Copyright Notice

Table of Contents

1. Introduction

The initial regulations (e.g., [FAA-14CFR]) and standards (e.g., [F3411]) for Unmanned Aircraft (UA) Systems (UAS) Remote Identification and tracking (RID) 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). DRIP's goal as stated in the WG charter 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. Small UA offer little capacity for computation and communication. UAS RID must also be accessible with ubiquitous and inexpensive devices without modification. This limits options.

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]).

Without authentication, an Observer has no basis for trust. As the messages are sent via wireless broadcast, they may be transmitted anywhere within the wireless range and making any claims desired by the sender.

DRIP Specific Authentication Methods, carried in ASTM Authentication Messages (Message Type 0x2) are defined herein. These methods, when properly used, 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 immediately actionable information.

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 [F3411] further support the important use case of Observers who are sometimes offline at the time of observation.

A summary of DRIP requirements [RFC9153] addressed herein is provided in Section 7.

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 (Observer, USS, UTM, etc.) defined in [RFC9153]. Other terms (such as DIME) are from [drip-arch], while others (RAA, HDA, etc.) are from [drip-rid].

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

Legacy Transports:

use of broadcast frames (Bluetooth 4.x) as specified in [F3411].

Extended Transports:

use of extended advertisements (Bluetooth 5.x), service info (Wi-Fi NAN) or vendor specific element information (Wi-Fi BEACON) in broadcast frames as specified in [F3411]. Must use ASTM Message Pack (Message Type 0xF).

3. Background

3.1. Reasoning for IETF DRIP Authentication

[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, 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 [drip-arch], there is a need to have Authentication formats to relay information for Observers to determine trust. 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.

3.1.1. UA Signed Evidence

When an Observer receives a DRIP-based Authentication Message (Section 4.3, Section 4.4, or Section 4.5) containing UA-signed Evidence it SHOULD validate the signature using the HI corresponding to the UA's DRIP Entity Tag (DET).

The UA's HI SHOULD be retrieved from DNS. If not available it may have been revoked. Note that accurate revocation status is a DIME inquiry; DNS non-response is a hint to the DET being expired or revoked. It MAY be retrieved from a local cache, if present. The local cache is typically populated by DNS lookups and/or by received Broadcast Endorsements (Section 3.1.2).

Once the Observer has the registered UA's DET and HI, all further (or cached previous) DRIP-based Authentication Messages using the UA DET can be validated. Signed content, tied to the DET, can now be trusted to have been signed by the holder of the private key corresponding to the DET.

Whether the content is true is a separate question which DRIP cannot address but sanity checks (Section 6) are possible.

3.1.2. DIME Endorsement of UA DET/HI

When an Observer receives a DRIP Link Authentication Message (Section 4.2) containing an Endorsement by the DIME of the UA DET/HI registration (Appendix B), it SHOULD validate the signature using the HI corresponding to the DIME's DET.

The DIME's HI, SHOULD be retrieved from from DNS (Section 5, [drip-registries]), when available. It MAY be cached from a prior DNS lookup or it may be stored in a distinct local store.

3.1.3. DIME Hierarchy Endorsements

An Observer can receive a series of DRIP Link Authentication Messages (Section 4.2), each one pertaining to a DIME's registration in the DIME above it in the hierarchy. Similar to Section 3.1.2, each link in this chain SHOULD be validated.

3.1.4. UAS RID Trust

Section 3.1.1, Section 3.1.2, and Section 3.1.3 complete the trust chain but the chain cannot yet be trusted as having any relevance to the observed UA because replay attacks are trivial. At this point the key nominally possessed by the UA is trusted but the UA has not yet been proven to possess that private key.

It is necessary for the UA to prove possession by dynamically signing data that is unique and unpredictable but easily verified by the Observer. This can be in the form of a DRIP Wrapper or Manifest (Section 4.3, Section 4.4) containing at least one ASTM Vector/Location Message and/or System Message (which contains a timestamp). 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).

3.2. ASTM Authentication Message

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

3.2.1. 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

This document leverages Authentication Type 0x5, Specific Authentication Method (SAM), as the principal authentication container, defining a set of SAM Types in Section 4. This is denoted in every Authentication Page in the Page Header. The SAM Type is denoted as a field in the Authentication Payload (see Section 4.1.1).

The Authentication Message is structured as a set of pages. 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 maximum 23-byte Authentication Payload. See Section 3.2.3 for more details. Over Bluetooth 4.x, these messages are "fragmented", with each page sent in a separate Bluetooth 4.x broadcast 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 (Bluetooth or Wi-Fi).

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 Figure 1's Authentication Payload, spanning multiple 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                        .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+

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

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

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

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

Figure 2: ASTM Authentication Message Fields

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 5.

3.2.3. ASTM Broadcast RID Constraints

3.2.3.1. Wireless Frame Constraints

A UA has the option of broadcasting using Bluetooth (4.x and 5.x) or Wi-Fi (BEACON or NAN), see Section 6. With Bluetooth, FAA and other Civil Aviation Authorities (CAA) mandate transmitting simultaneously over both 4.x and 5.x. With Wi-Fi, use of BEACON is recommended. Wi-Fi NAN is another option, depending on the CAA. The same application layer information defined in [F3411] MUST be transmitted over all the physical layer interfaces performing the function of RID.

Bluetooth 4.x presents a payload size challenge in that it can only transmit 25-bytes of payload per frame where the others all can support larger payloads per frame. However, the [F3411] messaging framing dictated by Bluetooth 4.x constraints is inherited by [F3411] over other media.

3.2.3.2. 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 denotes the length in bytes of Authentication Data / Signature only) MUST NOT exceed the value of 201. This includes the SAM Type but excludes Additional Data such as FEC.

4. DRIP Authentication Formats

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

When sending data over a medium that does not have underlying Forward Error Correction (FEC), for example Bluetooth 4.x, then Section 5 MUST be used. Appendix A gives a high-level overview of a state machine for decoding and determining a trustworthiness state. Appendix C shows an example of using the formats defined in this section.

4.1. DRIP Authentication Field Definitions

ASTM Message (25-bytes):

Full ASTM Message as defined in [F3411]; specifically Message Types 0x0, 0x1, 0x3, 0x4, and 0x5

ASTM Message Hash (8-bytes):

Hash of a single full ASTM Message using hash operations described in (Section 4.4.3). Multiple hashes MUST be in Message Type order.

Broadcast Endorsement (136-bytes):

DIME HI over UA DET/HI. Generated by a DIME during a UA DET, being used as a Session ID, registration. Used in Section 4.2.

Current Manifest Hash (8-bytes):

Hash of the current Manifest Message (Section 4.4). See Section 4.4.2.

Evidence (0 to 112 bytes):

Opaque evidence data that the UA is endorsing during its flight in Figure 4.

Frame Type (1-byte):

Sub-type for future different DRIP Frame formats. See Section 4.5.1.

Previous Manifest Hash (8-bytes):

Hash of the previously sent Manifest Message (Section 4.4). See Section 4.4.2.

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

The UA DET [drip-rid] in byte form (network byte order) and is part of Figure 4.

UA Signature (64-bytes):

Signature over all 4 preceding fields of Figure 4 using the HI of the UA.

Valid Not After (VNA) Timestamp by UA (4-bytes):

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

Valid Not Before (VNB) Timestamp by UA (4-bytes):

Timestamp denoting recommended time to start trusting data in Figure 4. MUST follow the format defined in [F3411]. That is a Unix-style timestamp but with an epoch of 01/01/2019 00:00:00. MUST be set no earlier than the time the signature is generated.

4.1.1. 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                   .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+

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

SAM Authentication Data (0 to 200 bytes):
    Authentication data (opaque to baseline F3411 but parsed by DRIP).
```

Figure 3: SAM Data Format

4.1.1.1. SAM Type

The SAM Type field is maintained by a TBD registrar and for DRIP four are planned to be allocated:

SAM Type	Description
To Be Assigned	DRIP Link (Section 4.2)
To Be Assigned	DRIP Wrapper (Section 4.3)
To Be Assigned	DRIP Manifest (Section 4.4)
To Be Assigned	DRIP Frame (Section 4.5)

Table 1

4.1.1.2. SAM Authentication Data

This field has a maximum size of 200-bytes, as defined by Section 3.2.3.

4.1.2. UA Signed Evidence

The DRIP Endorsement Structure (DES) [drip-registries] is used to create Signed Evidence by the UA during flight. It is encapsulated by the SAM Authentication Data field of Figure 3.

The DES MUST be 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 the DES as it will not fit in the ASTM Authentication Message.

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

Figure 4: Binary Encoded DRIP Endorsement Structure

UA DRIP Entity Tag:

This is the identity section of the DES and MUST be set to the UA DET (hhit).

Evidence:

The evidence section MUST be filled in with data in the form of an opaque object specified in the DRIP Wrapper, Manifest, or Frame sections.

UA Signature:

The UA private key MUST be used to generate the signature (sig_b16) found in the signature section.

The DES MUST be encoded in the binary form (as defined in [drip-registries]) to create the UA Signed Evidence. The general structure of the binary form can be seen in Figure 4.

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

4.2. DRIP Link

The DRIP Link SAM Type is used to transmit Broadcast Endorsements. For example, the Broadcast Endorsement: DIME, UA is sent (see Section 6.3) as a DRIP Link message. The structure is defined in [drip-registries] and an example of it can be found in Appendix B.

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 validations of the contents when the DIME DET/HI is in the receiver's cache. It also provides the UA HI so that connectivity is not required when performing validation of other DRIP Authentication Messages.

```
 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
+---------------+---------------+---------------+---------------+
|                                                               |
.                                                               .
.                      Broadcast Endorsement                    .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 5: DRIP Link

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. The hash of such a message SHOULD merely be included in a DRIP Manifest, but an entire such message MAY be encapsulated in a DRIP Wrapper periodically for stronger security.

4.3. DRIP Wrapper

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

The evidence section of the DES (Section 4.1.2) is populated with full (25-byte) [F3411] Broadcast RID messages. The ASTM Messages can be concatenated together to form a contiguous byte sequence (like in Figure 6) or be set in the evidence section as individual Claims.

The minimum number of messages support is 1 and the maximum supported is 4. The 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. Thus it may be preferred in some operation modes to use a DRIP Manifest Section 4.4 instead.

```
 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
+---------------+---------------+---------------+---------------+
|                                                               |
.                                                               .
.                         ASTM Message(s)                       .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 6: DRIP Wrapper Evidence

4.3.1. Wrapped Count & Sanity Check

When decoding a DRIP Wrapper on a receiver, a calculation of the number of messages wrapped and a sanity check can be performed by using the number of bytes (defined as wrapperLength) between the UA DET and the VNB Timestamp by UA such as in Figure 7.

```
if (wrapperLength MOD 25) != 0 {
  return DECODE_FAILURE
}
wrappedCount = wrapperLength / 25;
```

Figure 7: Pseudo-code for Wrapper sanity check and number of messages calculation

4.3.2. Wrapper over Extended Transports

To send the DRIP Wrapper over Extended Transports the messages being wrapped are co-located with the Authentication Message in a ATM

Message Pack (Message Type 0xF). The evidence section of the DES is cleared after signing leaving the following binary structure that is placed into the SAM Authentication Data of Figure 3 and sent in the same Message Pack.

```
 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
+---------------+---------------+---------------+---------------+
|                                                               |
|                              UA                               |
|                        DRIP Entity Tag                        |
|                                                               |
+---------------+---------------+---------------+---------------+
|                   Not Before Timestamp by UA                  |
+---------------+---------------+---------------+---------------+
|                    Not After Timestamp by UA                  |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                          UA Signature                         |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 8: DRIP Wrapper over Extended Transports

To verify the signature the receiver must concatenate all the messages in the Message Pack (excluding Authentication Message found in the same Message Pack) in Message Type order and set the DES evidence section before performing signature verification.

The functionality of a Wrapper in this form is identical to Message Set Signature (Authentication Type 0x3) when running over Extended Transports. What the Wrapper provides is the same format but over both Extended and Legacy Transports allowing the transports to be similar. Message Set Signature also implies using the ASTM validator system architecture which relies on Internet connectivity for verification which the receiver may not have at the time of receipt of an Authentication Message. 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 4 ASTM Messages that can be sent within it. Another limitation is that the format can not be used as a surrogate for messages it is wrapping. This is due to high potential a receiver on the ground does not support DRIP. Thus, when a Wrapper is being used the wrapped data must effectively be sent twice, once as a single framed message (as specified in [F3411]) and then 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. An Observer who has been listening for any length of time SHOULD hash received messages and cross-check them against the Manifest hashes. This is a way to evade the limitation of a maximum of 4 messages in the Wrapper (Section 4.3.3) and greatly reduce overhead.

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 Appendix E).

The evidence section of the DES (Section 4.1.2) is populated with 8-byte hashes of [F3411] Broadcast RID messages (from 2 to 11) and two special hashes (Section 4.4.2). All these hashes can be concatenated to form a contiguous byte sequence or be set in the evidence section individually. The Previous Manifest Hash and Current Manifest Hash MUST always come before the ASTM Message Hashes as seen in Figure 9.

A receiver SHOULD 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 E.

```
 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                             |
+---------------+---------------+---------------+---------------+
|                                                               |
.                                                               .
.                      ASTM Message Hashes                      .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 9: DRIP Manifest Evidence Structure

4.4.1. Hash Count & Sanity Check

When decoding a DRIP Manifest on a receiver, a calculation of the number of hashes and a sanity check can be performed by using the number of bytes (defined as manifestLength) between the UA DET and the VNB Timestamp by UA such as in Figure 10.

```
hashLength = 8
if (manifestLength MOD hashLength) != 0 {
  return DECODE_FAILURE
}
hashCount = (manifestLength / hashLength) - 2;
```

Figure 10: Pseudo-code for Manifest sanity check and number of hashes calculation

4.4.2. Pseudo-Blockchain Hashes

Two special hashes are included in all Manifests; the Previous Manifest Hash, which links to the previous Manifest, as well as the Current Manifest Hash. This gives a pseudo-blockchain provenance to the Manifest that could be traced back if the Observer was present for extended periods of time.

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 [drip-rid] that is signing the Manifest.

DET's using cSHAKE128 [NIST.SP.800-185] compute the hash as follows:

cSHAKE128(ASTM Message, 8, "", "Remote ID Auth Hash")

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

When building the list of hashes the Previous Manifest Hash is known from the previous Manifest. For the first built Manifest this value is null filled. The Current Manifest Hash is null filled while ASTM Messages are hashed and fill the ASTM Messages Hashes section. When all messages are hashed the Current Manifest Hash is computed over the Previous Manifest Hash, Current Manifest Hash (null filled) and ASTM Messages Hashes. This hash value replaces the null filled Current Manifest Hash and becomes the Previous Manifest Hash for the next Manifest.

4.4.3.1. Legacy Transport Hashing

Under this transport DRIP hashes the full ASTM Message being sent over the Bluetooth Advertising frame. For paged ASTM Messages (currently only Authentication Messages) all the pages are concatenated together and hashed as one object. For all other Message Types each individual 25-byte message is hashed.

4.4.3.2. Extended Transport Hashing

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

4.5. 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.

The population of the evidence section of the DES (Section 4.1.2) is not defined in this document and MUST be openly specified by the implementation (or specification) using it.

```
 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 11: DRIP Frame

4.5.1. Frame Type

Byte to sub-type for future different DRIP Frame formats. It takes the first byte in Figure 11 leaving 111-bytes available for Frame Evidence Data.

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

Table 2

5. 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). The Bluetooth 4.x Legacy Transport does not have supporting FEC so with DRIP Authentication the following application level FEC scheme is used to add FEC. When sending data over a medium that does not have underlying FEC, for example Bluetooth 4.x, then 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 single erased page in an Authentication Message. Other FEC schemes, to protect more than a single page of an Authentication Message or multiple [F3411] Messages, is 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 bytes + number of parity bytes.

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-byte Authentication Payload field of Figure 1 is used in the XOR operations. For Page 0, a 23-byte null pad is used for the previous parity page.

Figure 12 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-bytes of FEC data and in this example 10-bytes of padding 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 12: Example Single Page FEC Encoding

5.2. Decoding

To determine if FEC has been used a simple check of the Last Page Index can be used. In general if the Last Page Index field is one greater than that necessary to hold Length bytes of Authentication Data then FEC has been used. Note however that if Length bytes was exhausted exactly at the end of an Authentication Page then the Additional Data Length will occupy the first byte of the following page the remainder of which under DRIP will be null padded: in this case the Last Page Index will have been incremented by one more for FEC.

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 1-byte in length, so it will roll over (to 0x00) after reaching its maximum value (0xFF). If only 1-page is missing in the Authentication Message the resulting parity bytes 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 are REQUIRED to be sanity checked by DRIP. The pseudo-code in Figure 13 can be used for both checks.

```
function decode_check(auth_pages[], decoded_lpi, decoded_length) {
  // check decoded Last Page Index (LPI) does not exceed maximum LPI
  if (decoded_lpi >= 16) {
    return DECODE_FAILURE
  }

  // check that decoded length does not exceed DRIP maximum
  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 LPI
  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
}
```

Figure 13: Pseudo-code for Decode Checks

Implementations MAY also implement an heuristic extension (Appendix D) to decode if both the first page (Page 0) and last page (Last Page Index) are missing.

5.3. FEC Limitations

The worst case scenario is when the Authentication Data / Signature ends perfectly on a page (Page N-1). This means the Additional Data Length would start the next page (Page N) and have 22-bytes worth of null padding to align the FEC 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. This should be avoided where possible in an effort to maintain efficiency.

6. Requirements & Recommendations

6.1. Legacy Transports

With Legacy Advertisements the goal is to attempt to bring reliable receipt of the paged Authentication Message. FEC (Section 5) MUST be used, per mandated RID rules (for example the US FAA RID Rule [FAA-14CFR]), when using Legacy Advertising methods (such as Bluetooth 4.x).

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

6.2. Extended Transports

Under the ASTM specification, Bluetooth 5.x, Wi-Fi NAN, and Wi-Fi BEACON transport of RID 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.x 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 FEC (Section 5) MUST NOT be used as it is impractical.

6.3. Authentication

It is REQUIRED that a UA send the following DRIP Authentication Formats to fulfill the requirements in [RFC9153]:

1. SHOULD: send DRIP Link (Section 4.2) using the Broadcast Endorsement: DIME:Apex, DIME:RAA (satisfying GEN-3); at least once per 5 minutes

2. MUST: send DRIP Link (Section 4.2) using the Broadcast Endorsement: DIME:RAA, DIME:HDA (satisfying GEN-3); at least once per 5 minutes

3. MUST: send DRIP Link (Section 4.2) using the Broadcast Endorsement: DIME:HDA, UA (satisfying ID-5, GEN-1 and GEN-3); at least once per minute

4. MUST: send any other DRIP Authentication Format (RECOMMENDED: DRIP Manifest (Section 4.4) or DRIP Wrapper (Section 4.3)) 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

6.4. 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 Broadcast Endorsement: DIME, UA.

A separate issue is the frequency of transmitting the DRIP Link Authentication Message for the Broadcast Endorsement: DIME, UA when using the Manifest. 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 Broadcast Endorsement: DIME, UA to mitigate potential packet loss.

The following operational configuration is RECOMMENDED (in alignment with Section 6.3):

1. Per CAA requirements, generate and transmit a set of ASTM Messages (example; Basic ID, Location and System).

2. Under Extended Transports, generate and include in the same Message Pack as the CAA required ASTM Messages a DRIP Wrapper as specified in Section 4.3.2.

3. Under Legacy Transports, generate and transmit every 5 seconds a DRIP Manifest (Section 4.4) hashing as many sets of recent CAA required ASTM Messages. The system MAY periodically replace

the DRIP Manifest with a DRIP Wrapper (Section 4.3) containing at least a Location Message (Message Type 0x2).

4. Under both Legacy or Extended Transports, generate and transmit a DRIP Link's (Section 4.2) containing; Broadcast Endorsement: DIME:HDA, UA every minute, Broadcast Endorsement: DIME:RAA, DIME:HDA every 5 minutes, Broadcast Endorsement: DIME:Apex, DIME:RAA every 5 minutes.

The reasoning and math behind this recommendation can be found in Appendix E.

6.4.1. 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]. Sending them within the DRIP Wrapper makes 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 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 Location Messages (Message Type 0x2) on a map display an embedded Location Message in a DRIP Wrapper can be marked differently (e.g. via color) to signify trust in the Location data.

6.4.2. UAS RID Trust Assessment

As described in Section 3.1.4, the receiver MUST perform verification of the data being received in Broadcast RID.

After signature validation 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 verification. An example of another source of information is a visual confirmation of the UA position.

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

7. Summary of Addressed DRIP Requirements

The following [RFC9153] requirements 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

This document requests a new subregistry for Frame Type under the DRIP registry.

DRIP Frame Type: This 8-bit valued subregistry is for Frame Types in DRIP Frame Authentication Messages. Future additions to this subregistry are to be made through Expert Review (Section 4.5 of [RFC8126]). The following values are defined:

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

9. Security Considerations

9.1. 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 DRIP Link messages can easily be replayed by an attacker who has copied them from previous broadcasts.

If an attacker (who is smart and spoofs more than just the UAS ID/data payloads) willing replays an DRIP Link message they have in principle actually helped by ensuring the DRIP Link is sent more frequently and be received by potential Observers.

The primary mitigation is the UA is REQUIRED to send more than DRIP Link messages, specifically the Manifest and/or Wrapper messages that sign over changing data ASTM Messages (e.g. Location/Vector Messages) using the DET private key. An UA sending these messages then actually signing these and other messages using the DET key provides the Observer with data that proves realtime signing. 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.

9.2. VNA Timestamp Offsets for DRIP Authentication Formats

Note the discussion of VNA Timestamp offsets here is in 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-registries].

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 recommended that 3-5 minutes should be sufficient to serve this purpose in any scenario, but is not limited by design.

10. 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).

* Many thanks to Rick Salz for the secdir review.

* Thanks to Matt Joras for a genart review.

11. References

11.1. Normative References

[drip-arch] Card, S. W., Wiethuechter, A., Moskowitz, R., Zhao, S., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Architecture", Work in Progress, Internet-Draft, draft-ietf-drip-arch-29, 16 August 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-drip-arch-29>.

[F3411]
"F3411-22a: Standard Specification for Remote ID and Tracking", July 2022.

[NIST.SP.800-185] Kelsey, J., Change, S., Perlner, R., and NIST, "SHA-3 derived functions: cSHAKE, KMAC, TupleHash and ParallelHash", NIST Special Publications (General) 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>.

11.2. Informative References

[drip-registries] Wiethuechter, A. and J. Reid, "DRIP Entity Tag (DET) Identity Management Architecture", Work in Progress, Internet-Draft, draft-ietf-drip-registries-07, 5 December 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-drip-registries-07>.

[drip-rid] Moskowitz, R., Card, S. W., Wiethuechter, A., and A. Gurtov, "DRIP Entity Tag (DET) for Unmanned Aircraft System Remote ID (UAS RID)", Work in Progress, Internet-Draft, draft-ietf-drip-rid-37, 2 December 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-drip-rid-37>.

[FAA-14CFR] "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>.

Appendix A. Authentication State Diagrams & Color Scheme

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

There are two keys of interest: the PK of the UA and the PK of the DIME. This document gives a clear way to send the PK of the UA over the Broadcast RID messages. The key of the DIME can be sent over Broadcast RID using the same mechanisms (see Section 4.2 and Section 6.3) but is not required due to potential operational constraints of sending multiple DRIP Link messages. As such there are scenarios where you may have part of the key-chain but not all of it.

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

A.1. State Colors

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

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

Table 3

A.2. State Diagrams

This section gives some RECOMMENDED state flows that DRIP should follow. Note that the state diagrams do not have all error conditions mapped.

A.2.1. Notations

```
                  o--------------o
                  |   PROCESS    |
                  o--------------o

                  +--------------+
                  |    STATE     |
                  +--------------+

                   ooooo
                  o  N  o    Transition N
                   ooooo

                  +----->    Transition Option False/No

                  ----->     Transition Option True/Yes
```

Figure 14: Diagram Notations

A.2.2. General

```
o---------------------o      ooooo        +------+
|        Start        |---->o  1  o+----->| None |
o---------------------o      ooooo        +------+
                               |
                               v
                             ooooo        +-------------+
                            o  2  o+----->| Unsupported |
                             ooooo        +-------------+
                               |             ^
                               v             |
          +---------+        ooooo           |
          | Partial |<-----+o  3  o          |
          +---------+        ooooo           |
                               |             |
                               v             +
                             ooooo         ooooo        o-------------o
                            o  4  o------>o  5  o------>| SAM Decoder |
                             ooooo         ooooo        o-------------o
                               +
                               |
                               v
                        o------------------o
                        | AuthType Decoder |
                        o------------------o
```

Figure 15: Standard Authentication Colors/State

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

Table 4

A.2.3. DRIP SAM

```
o-------------o      ooooo        o-----------------------------o
| SAM Decoder |---->o  6  o------>| DRIP Wrapper/Manifest/Frame |
o-------------o      ooooo        o-----------------------------o
                       +                 |              ^
                       |                 |              |
                       v                 v              |
               o-----------o    o--------------------o |
               | DRIP Link |--->| Update State Cache | |
               o-----------o    o--------------------o |
                                  |                    |
                                  v                    |
       o--------------o         ooooo       o----------------------o
       | NOP / Return |<------+o  7  o----->| Extract Message from |
       o--------------o         ooooo       | Verification Queue   |
                                            o----------------------o
```

Figure 16: DRIP SAM Decoder

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

Table 5

A.2.4. DRIP Link

```
o-----------o       ooooo         ooooo        +--------------+
| DRIP Link |----->o  8  o+----->o  9  o+----->| Unverifiable |
o-----------o       ooooo         ooooo        +--------------+
                      |             |
                      |-------------'
                      v
                    ooooo        +------------+
                   o  10 o+----->| Unverified |
                    ooooo        +------------+
                      |
                      v
                o----------------------o
                | Add UA DET/PK        |
                | to Key Cache         |
                o----------------------o
                      |
                      v
                    ooooo         +----------+
                   o  11 o+------>| Verified |
                    ooooo         +----------+
                      |              ^
                      v              |
                o--------------------------o
                | Mark UA DET/PK           |
                | as Trusted in Key Cache  |
                o--------------------------o
```

Figure 17: DRIP Link State Decoder

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

Table 6

A.2.5. DRIP Wrapper/Manifest/Frame

```
o-----------------------------o         +--------------+
| DRIP Wrapper/Manifest/Frame |         | Unverifiable |
o-----------------------------o         +--------------+
           |                                   ^
           v                                   |
         ooooo         ooooo        o--------------------o
        o  12 o+----->o  13 o+----->| Add Message to     |
         ooooo         ooooo        | Verification Queue |
           |             |          o--------------------o
           |             |
           |-------------'
           v
         ooooo         ooooo         ooooo         +------------+
        o  14 o+----->o  15 o+----->o  16 o+----->| Unverified |
         ooooo         ooooo         ooooo         +------------+
           |             |             |
           v             v             |
         ooooo        +-------------+  |
        o  17 o+----->| Conflicting |  |
         ooooo        +-------------+  |
           |                           |
           v                           v
         ooooo                  +--------------+
        o  18 o---------------->| Questionable |
         ooooo                  +--------------+
           +
           |
           v
         ooooo        +----------+
        o  19 o+----->| Verified |
         ooooo        +----------+
           |
           v
        +---------+
        | Trusted |
        +---------+
```

Figure 18: DRIP Wrapper/Manifest/Frame State Decoder

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

Transition	Transition Query	Next State/Process/ Transition (Yes, No)
	Has past Messages of this type been marked as Questionable or Verified?	
17	Has past Messages of this type been marked as Conflicting?	Conflicting, 18
18	Has past Messages of this type been marked as Questionable or Unverified?	Questionable, 19
19	Is UA DET/PK marked as Trusted in Key Cache?	Trusted, Verified

Table 7

Appendix B. Broadcast Endorsement: DIME, UA

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+---------------+---------------+
|                                                               |
|                             DRIP                              |
|                      Entity Tag of DIME                       |
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
|                             DRIP                              |
|                       Entity Tag of UA                        |
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                      Host Identity of UA                      |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
|                     VNB Timestamp by DIME                     |
+---------------+---------------+---------------+---------------+
|                     VNA Timestamp by DIME                     |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                      Signature by DIME                        |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
```

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

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

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

VNB Timestamp by DIME (4-bytes):
    Current time at signing.

VNA Timestamp by DIME (4-bytes):
    Timestamp denoting recommended time to trust DIME Endorsement
    of UA DET and HI (may be minutes to months in the future).

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

Figure 19: Example DRIP Broadcast Endorsement: DIME, UA

Appendix C. Example TX/RX Flow

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

```
        +--------------------+
  .-----| Unmanned Aircraft |-----.
  |     +--------------------+     |
  | 0              | 1              | 2
  |                |                |

  O                O                O
--|--            --|--            --|--
 / \              / \              / \
  A                B                C


Broadcast Paths: Messages Received
0: None
1: DRIP Link or DRIP Wrapper or DRIP Manifest
2: DRIP Link and DRIP Wrapper or DRIP Manifest

Observers: Authentication State
A: None
B: Unverifiable
C: Verified, Trusted, Unverified, Questionable, or Conflicting
```

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

Appendix D. Additional FEC Decoding Heuristic

With Section 5, if Page 0 and the FEC page are missing from the Authentication Message there is a heuristic that can be applied instead of FEC decoding to obtain the Authentication Data. This is based on the structure of the DRIP Authentication Messages and additional information sent over the broadcast or via lookup in DNS.

Looking at Page 0 (Figure 20) of any DRIP Authentication Format the payload data is always a DET. For DRIP Link (Section 4.2) this DET is of the DIME while for DRIP Wrapper (Section 4.3), Manifest (Section 4.4) and Frame (Section 4.5) it is the DET of the UA.

```
 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 Headers    +---------------+
|                                               |   SAM Type    |
+---------------+---------------+---------------+---------------+
|                                                               |
|                             DRIP                              |
|                          Entity Tag                           |
|                                                               |
+---------------+---------------+---------------+---------------+

Page Header: (1-byte)
    Authentication Type (4-bits)
    Page Number (4 bits)

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

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

DRIP Entity Tag: (16-bytes)
  DET of an entity in network byte order
```

Figure 20: Example Page 0 from DRIP Authentication Message

Under DRIP, the Basic ID Message (Message Type 0x1) SHOULD be using Specific Session ID (ID Type 4) subtype IETF DRIP Entity ID (Type 1). This DET of the UA can be used in place of the missing DET in a DRIP Wrapper, Manifest and Frame. For DRIP Link, which is missing the DET of the DIME, the lookup properties of the DET enables the discovery, via DNS, the DIME's DET.

These DETs obtained via other means can replace the missing payload of Authentication Page 0 and enable the full decoding and verification of the DRIP Authentication Message.

When the missing DET is supposed to be of the UA the DET MAY be sourced from the Basic ID Message (Message Type 0x1). Under DRIP, this SHOULD be set to the DET missing in the Authentication Data.

Appendix E. Operational Recommendation Analysis

The recommendations found in (Section 6.4) may seem heavy handed and specific. This appendix lays out the math and assumptions made to come to the recommendations listed there. This section is solely based on operations using Bluetooth 4.x; as such all calculations of frame counts for DRIP included FEC using Section 5.

E.1. Methodology

In the US, the required ASTM Messages to be transmitted every second are: Basic ID (0x1), Location (0x2) and System (0x4). Typical implementations will most likely send at a higher rate (2x sets per cycle) resulting in 6 frames sent per cycle.

> Informational Note: in Europe the Operator ID Message (0x5) is also included; pushing the frame count to 8 per cycle. For Japan two Basic ID (0x0), Location (0x1) and Authentication (0x2) are required.

To calculate the frame count of a given DRIP Authentication Message the following formula is used:

```
1 + ceiling((((16 + 8 + 64) + (Item Size * Item Count) + 2) - 16)
/ 23) + 1
```

The leading 1 is counting for the Page 0 which is always present. The DET (16-bytes), timestamps (8-bytes) and signature (64-bytes) all make up the required fields for DRIP. Item Size (in bytes) is size of each item in a given format; for a Wrapper it is 25 (a full ASTM Message), while for a Manifest it is 8 (a single hash). 2 more is added to account for the SAM Type and the ADL byte. The value 16 is the number of bytes not counted (as they are part of Page 0 which is already counted for). 23 is the number of bytes per Authentication Page (pages 1 - 15). After dividing by 23 the value is raised to the nearest whole value as we can only send full frames, not partial. The final 1 is counting for a single page of FEC applied in DRIP under Bluetooth 4.x.

> Informational Note: for DRIP Link the Item Size is 48 and Item Count is 1; resulting in a frame count of 8

Comparing DRIP Wrapper and Manifest Authentication Message frame counts we have the following:

Authenticated Frames	**Wrapper Frames**	**Manifest Frames**	**Total Wrapper Frames**	**Total Manifest Frames**
1	7	7	8	8
2	8	7	10	9
3	9	7	12	10
4	10	8	14	12
5	N/A	8	N/A	13
6	N/A	8	N/A	14
7	N/A	9	N/A	16
8	N/A	9	N/A	17

Authenticated Frames	Wrapper Frames	Manifest Frames	Total Wrapper Frames	Total Manifest Frames
9	N/A	10	N/A	19
10	N/A	10	N/A	20
11	N/A	10	N/A	21

Table 8: Frame Counts

Note that for Manifest Frames the calculations use an Item Count that is 2 + Authentication Frames. This is to account for the two special hashes.

The values in Total Frames is calculated by adding in the Item Count (to either the Wrapper Frames or Manifest Frames column) to account for the ASTM Messages being sent outside the Authentication Message.

E.2. ASTM Maximum Schedule Example

For this example we will assume the following ASTM Messages are in play:

*1x Basic ID (0x0) set as ID Type for Serial Number (0x1)

*1x Basic ID (0x0) set as ID Type for CAA Assigned ID (0x2)

*1x Basic ID (0x0) set as ID Type for UTM Assigned ID (0x3)

*1x Basic ID (0x0) set as ID Type for Specific Session ID (0x4)

*2x Location (0x1)

*1x Self ID (0x3)

*2x System (0x4)

*2x Operator ID (0x5)

This message set is uses all single frame ASTM Messages, sending a set of them (Location, System and Operator ID) at a rate of 2 per second. Two Basic ID's are sent in a single second and rotate between the 4 defined (1x per type). A single Self ID is sent every second. All messages in a given second if appear more than once are exact duplicates.

```
+-----------------------------------------------------------+
|                        Frame Slots                        |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 |
+----+----+----+----+----+----+----+----+----+----+----+----+
| A* | V* | S  | O  | B  | V  | S* | O  | I  |   L/W[0,2]   |
+----+----+----+----+----+----+----+----+----+----+----+----+
| C* | V  | S* | O  | D* | V* | S  | O  | I* |   L/W[3,5]   |
+----+----+----+----+----+----+----+----+----+----+----+----+
| A  | V* | S  | O* | B* | V  | S  | O  | I  |L/W[6,7] | ## |
+----+----+----+----+----+----+----+----+----+----+----+----+
| C  | V  | S  | O  | D  | V  | S  | O  | I  |    M[0,2]    |
+----+----+----+----+----+----+----+----+----+----+----+----+
| A  | V  | S  | O  | B  | V  | S  | O  | I  |    M[3,5]    |
+----+----+----+----+----+----+----+----+----+----+----+----+
| C  | V  | S  | O  | D  | V  | S  | O  | I  |    M[6,8]    |
+----+----+----+----+----+----+----+----+----+----+----+----+
| A  | V  | S  | O  | B  | V  | S  | O  | I  |M[9]| ## | ## |
+----+----+----+----+----+----+----+----+----+----+----+----+

# = Empty Frame Slot
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)
  Wrapping Location (0x1) and System (0x4)
M(x)[y,z] = DRIP Manifest Authentication Message (0x2)
  x = Number Hashes
  y = Start Page
  z = End Page
* = Message in DRIP Manifest Authentication Message
```

Figure 21: Example Transmit Schedule

Manifest messages in the schedule are filled with unique messages from previously transmitted messages before the new Manifest is sent. In Figure 21 this is denoted by the * symbol as being part of the Manifest. For the figure 24 messages are eligible for the Manifest in the very first cycle of transmission. In future iterations, 56 messages are eligible across the 7 seconds it takes to send the previous Manifest and the next Link/Wrapper. Care should be given into the selection of messages for a Manifest as there is a limit of 11 hashes.

Informational Note: the term "unique message" above is used as in the example schedule the 2nd Location and System messages MAY be exact copies of the previous Location and System messages sent in the same second. Duplicates of this kind SHOULD NOT be included in a Manifest.

In the schedule the Wrapper and the Link messages switch back and forth the contents of them are changing in the following order:

```
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 (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 (0x1), System (0x4)
Link: ??? on Apex
```

Any messages not required for a local jurisdiction can be removed from the schedule. It is RECOMMENDED this empty frame slot is left empty to help with timing due to RF constraints/concerns. For example in the US the Self ID(0x3) and Operator ID (0x5) are not required and can be ignored in the above figures. In the US only one Basic ID (0x0) is selected at any given time, opening up 3 more slots.

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