Workgroup: DRIP Working Group
Internet-Draft: draft-ietf-drip-auth-12
Published: 25 May 2022
Intended Status: Standards Track
Expires: 26 November 2022
Authors: A. Wiethuechter (Editor)
AX Enterprize, LLC
S. Card
AX Enterprize, LLC
R. Moskowitz
HTT Consulting

DRIP Entity Tag Authentication Formats & Protocols for Broadcast Remote ID

Abstract

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 26 November 2022.

Table of Contents

1. Introduction

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

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

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

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

This authentication methodology also provides the missing, but US FAA mandated, Error Correction for the Bluetooth 4 transmissions (see Section 4). This is error correction not only for the authentication message itself, but indirectly, to other messages authenticated via the Manifest method (see Section 5.4).

1.1. DRIP Requirements Addressed

The following [drip-requirements] will be addressed:

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

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

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

See Section 6.3 for further clarification.

2. Terminology

2.1. Required Terminology

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

2.2. Definitions

See [drip-requirements] for common DRIP terms.

Legacy Transports: uses broadcast frames (Bluetooth 4).

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

3. Background

3.1. Problem Space and Focus

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

3.1.1. Broadcast RID RF Options

A UA has the option of broadcasting using Bluetooth (4 and 5) or Wi-Fi (BEACON or NAN), see Section 6. With Bluetooth, FAA and other CAA

mandate transmitting simultaneously over both 4 and 5. With Wi-Fi, use of BEACON is recommended. Wi-Fi NAN is another option, depending on CAA.

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

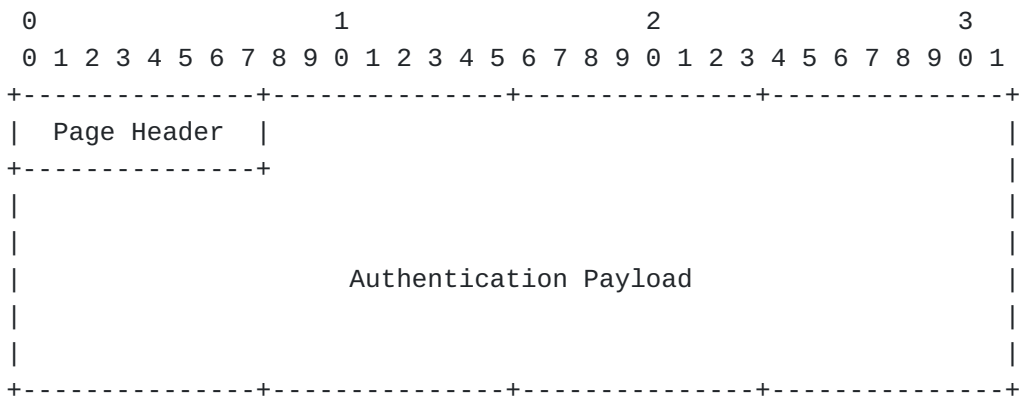
3.2. Reasoning for IETF DRIP Authentication

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

3.3. ASTM Authentication Message

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

3.3.1. Authentication Page



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

Authentication Payload: (23 bytes per page)
    Authentication Payload, including headers. Null padded.

Figure 1: Standard ASTM Authentication Message Page

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

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

3.3.1.1. Authentication Type

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

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

Table 1

3.3.1.1.1. Specific Authentication Method (SAM)

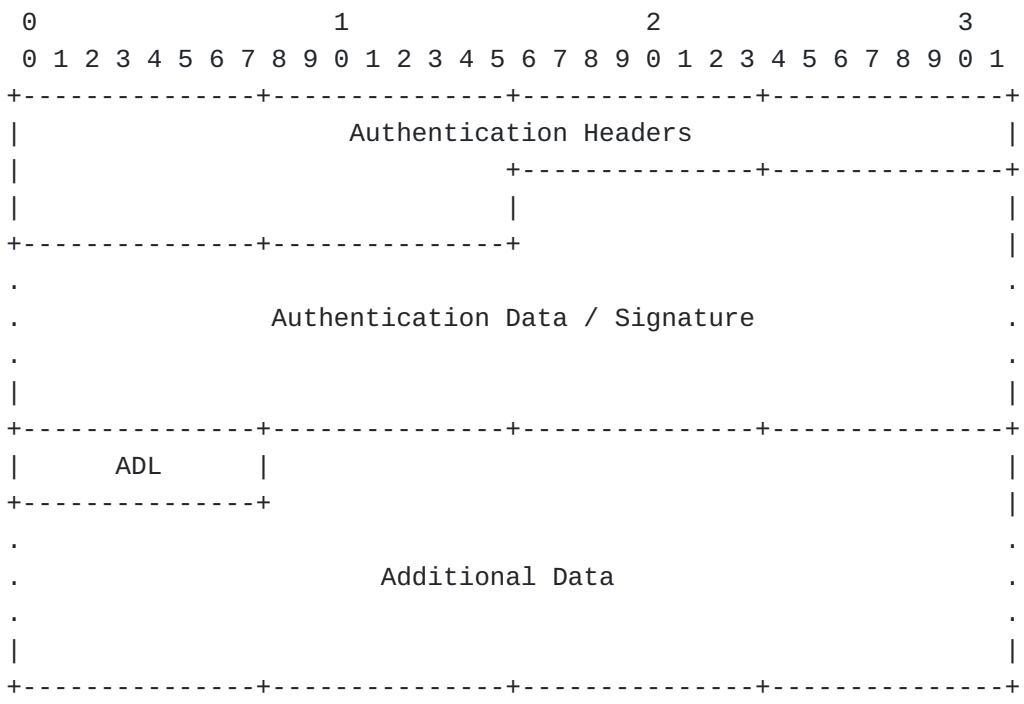
This document leverages Authentication Type 0x5, Specific Authentication Method (SAM), defining a set of SAM Types in Section 5.

3.3.1.2. Page Number

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

3.3.1.3. Authentication Payload Field

The following is shown in its complete format.



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

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

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

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

Figure 2: ASTM Authentication Message Fields

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

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

3.3.2. ASTM Constraints

To keep consistent formatting across the different transports (Legacy and Extended) and their independent restrictions the authentication data being sent is REQUIRED to fit within the page

limit of the most constrained existing transport can support. Under Broadcast RID the transport that can hold the least amount of authentication data is Bluetooth 5 and Wi-Fi BEACON at 9-pages.

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

1. Authentication Data / Signature data MUST fit in 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.

4. Forward Error Correction

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

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 max of 16-pages.

4.1. Encoding

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

4.1.1. Single Page FEC

To generate the parity a simple XOR operation using the previous and current page is used. Only the 23-byte Authentication Page data is used in the XOR operation. For Page 0, a 23-byte null pad is used for the previous page. The resulting parity fills the last 23 bytes of the Additional Data field of Figure 2 with the Additional Data Length field being set to 23 or greater (depending on number of null pad bytes are needed to get onto the next page).

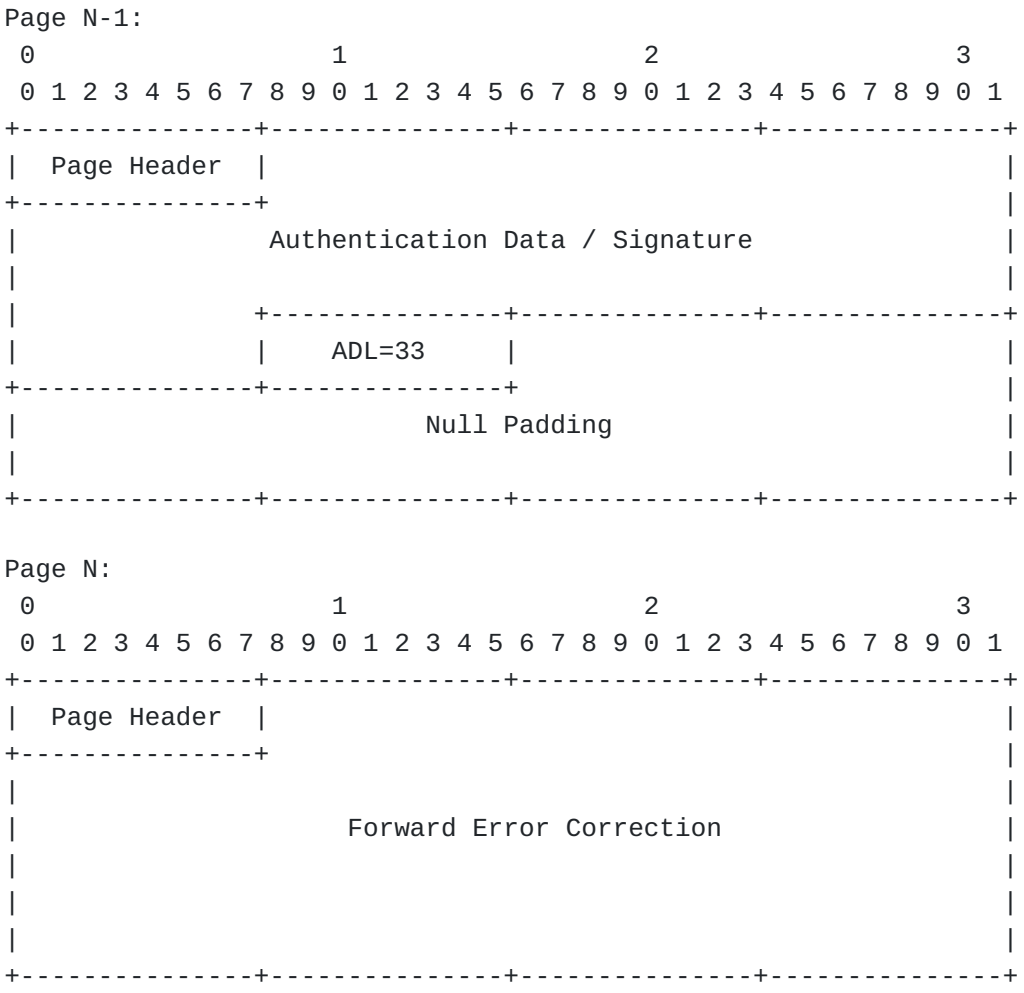


Figure 3: Example Single Page FEC Encoding

4.1.2. Multiple Page FEC

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

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

4.1.2.1. Page Recovery

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

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

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

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

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

```
57 dc6657acd30b2ec4aa582049f52adf9f922e62c469563a
58 6c636a59145a55417a3895fd543f19e94200be4abc5e94
59 02bba5e28f5896d754caf50016a983993b149b5c9e6eeb
```

4.1.2.2. Frame Recovery

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

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

```
10 42012001001000a0014579d8a404d48f2ef9000000000000
11 249600006efeb019ee111ed37a097a0948081c10ffff0000
12 50098960bf8c05042001001000a00145aac6b00abba268b7
12 512001001000a0014579d8a404d48f2ef9bb9a4470ada5b4
12 52ff1352c7402af9d9ebd20034e8d7a12920f4d7e91c1a73
12 53dca7d04e776150825863c512c6eb075a206a95c59b297e
12 54f2935fd416f27b1b42fd5d9dfaa0dec79f32287f41b454
12 557101415def153a770d3e6c0b17ae560809bc634a822c1f
12 563b1064b80a0000000000000000000000000000000000
13 0052656372656174696f6e616c2054657374000000000000
14 02c2ffb019322d1ed3010000c008e40700fc080000000000
15 004e2e4f503132333435360000000000000000000000000000
```

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

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

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

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

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

```
12 57 6c86337bf7ab746f5d62bb7f8de954104b121585d3975f
12 58 6e923f06c1bce165b0e25930d57a63c24f751145e1dd8d
12 59 c115029b42e9979580327a6a14d421c12a33aa2e1a2e51
12 5a 7daaee581016b8012a7b3964f7b2720d387bfa77e94555
12 5b 6f1831cd477ef3a3a85bb403aada89926fb8fc2a14a9ca
12 5c acb4ec2f3a6ed2d8e9f900000000000000000000000000
```

4.2. Decoding

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

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

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

Data Length byte can further confirm this; taking into account any null padding needed for page alignment.

4.2.1. Single Page FEC

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

4.2.2. Multiple Page FEC

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

4.2.2.1. Page Recovery

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

4.2.2.2. Frame Recovery

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

4.3. FEC Limitations

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

5. DRIP Authentication Formats

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

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

5.1. 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 (12-bytes): Hash of a single full ASTM Message using hash operations described in (Section 5.4.2). Multiple hashes MUST be in Message Type order.

Attestation Data (0 to 112 bytes): Opaque attestation data that the UA is attesting during its flight in Figure 4.

Broadcast Attestation (136-bytes): HDA HI over UA DET/HI. Generated by a DRIP Registry during Session ID registration. Used in Section 5.2.

Current Manifest Hash (12-bytes): See Section 5.4.3.

Frame Type (1-byte): Sub-type for future different DRIP Frame formats. See Section 5.5.1.

Not Before 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 to the time the signature is generated.

Not After 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 ahead of Not Before Timestamp.

Previous Manifest Hash (12-bytes): See Section 5.4.3.

UA DRIP Entity Tag (16-bytes): The UA DET in byte form (network byte order) and is part of Figure 4.

UA Signature (64-bytes): Signature over preceding fields of Figure 4 using the HI of the UA.

5.1.1. Broadcast Attestation Structure

To directly support Broadcast RID a variation of the Attestation Structure format of [drip-registries] SHOULD be used when running DRIP under the various SAM Types (filling the SAM Authentication Data field (Section 5.1.2.2)). The notable changes of the structure is that the timestamps are set by the UA and the Attestor Identity Information is set to the DET of the UA.

When using this structure the UA is always self-attesting its DRIP Entity Tag (DET). The Host Identity of the UA DET can be looked up by mechanisms described in [drip-registries] or by extracting it from Broadcast Attestation (see Section 5.2 and Section 6.3).

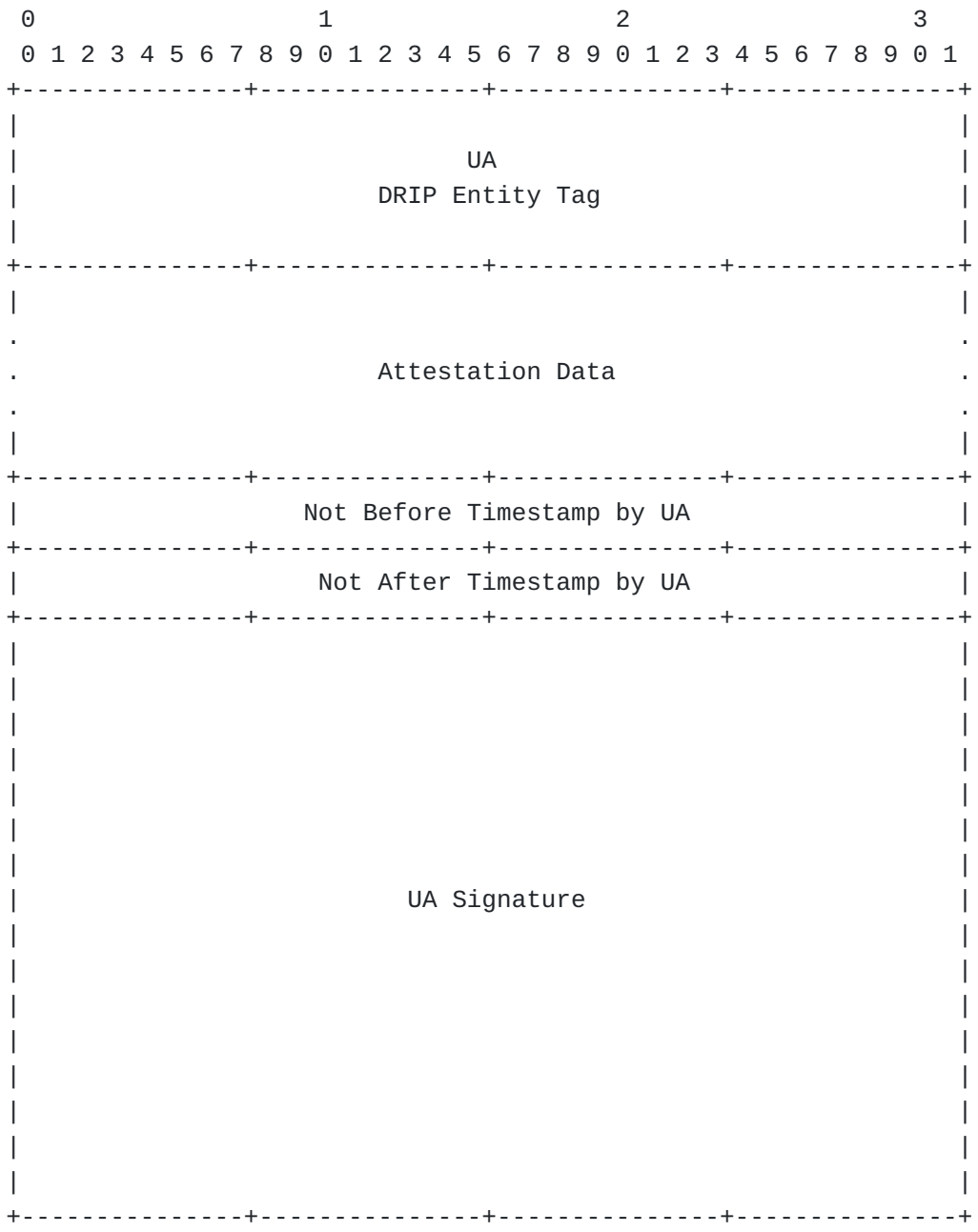
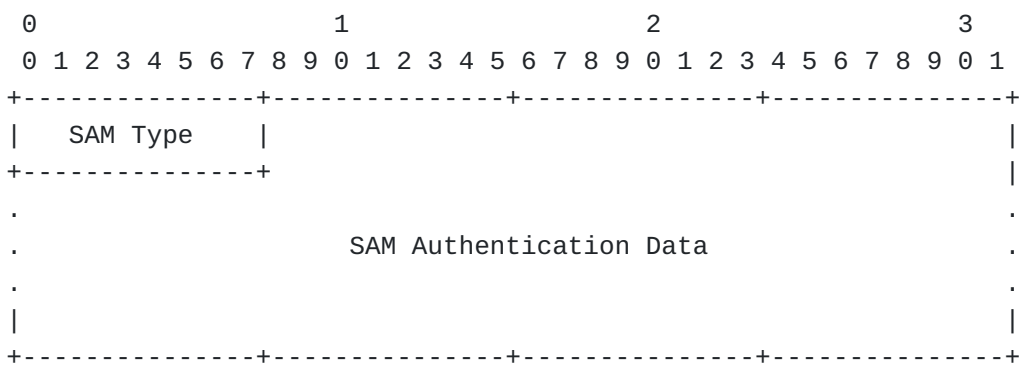


Figure 4: Broadcast Attestation Structure

5.1.2. SAM Data Format

Figure 5 is the general format to hold authentication data when using SAM and is placed inside the Authentication Data / Signature field in Figure 2.



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

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

Figure 5: SAM Data Format

5.1.2.1. SAM Type

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

SAM Type	Description
0x01	DRIP Link (Section 5.2)
0x02	DRIP Wrapper (Section 5.3)
0x03	DRIP Manifest (Section 5.4)
0x04	DRIP Frame (Section 5.5)

Table 2

5.1.2.2. SAM Authentication Data

This field has a maximum size of 200-bytes, as defined by Section 3.3.2. The Broadcast Attestation Structure (Section 5.1.1) SHOULD be used in this space.

5.2. DRIP Link

This SAM Type is used to transmit Broadcast Attestation's. For example, the Broadcast Attestation of the Registry (HDA) over the UA is sent (see Section 6.3) as a DRIP Link message. Its 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 that the UA is currently broadcasting. This message does not require internet connectivity to perform signature validations of the contents when the registry DET/HI is in the receivers cache.

It also provides the UA HI so that connectivity is not required when performing validation of other DRIP Authentication Messages.

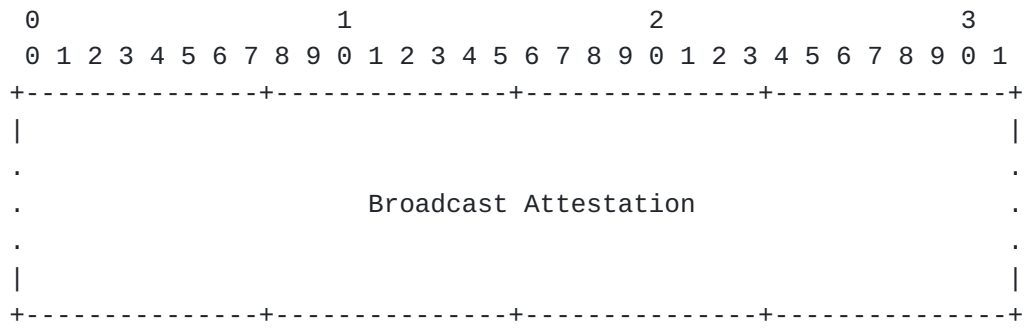


Figure 6: DRIP Link

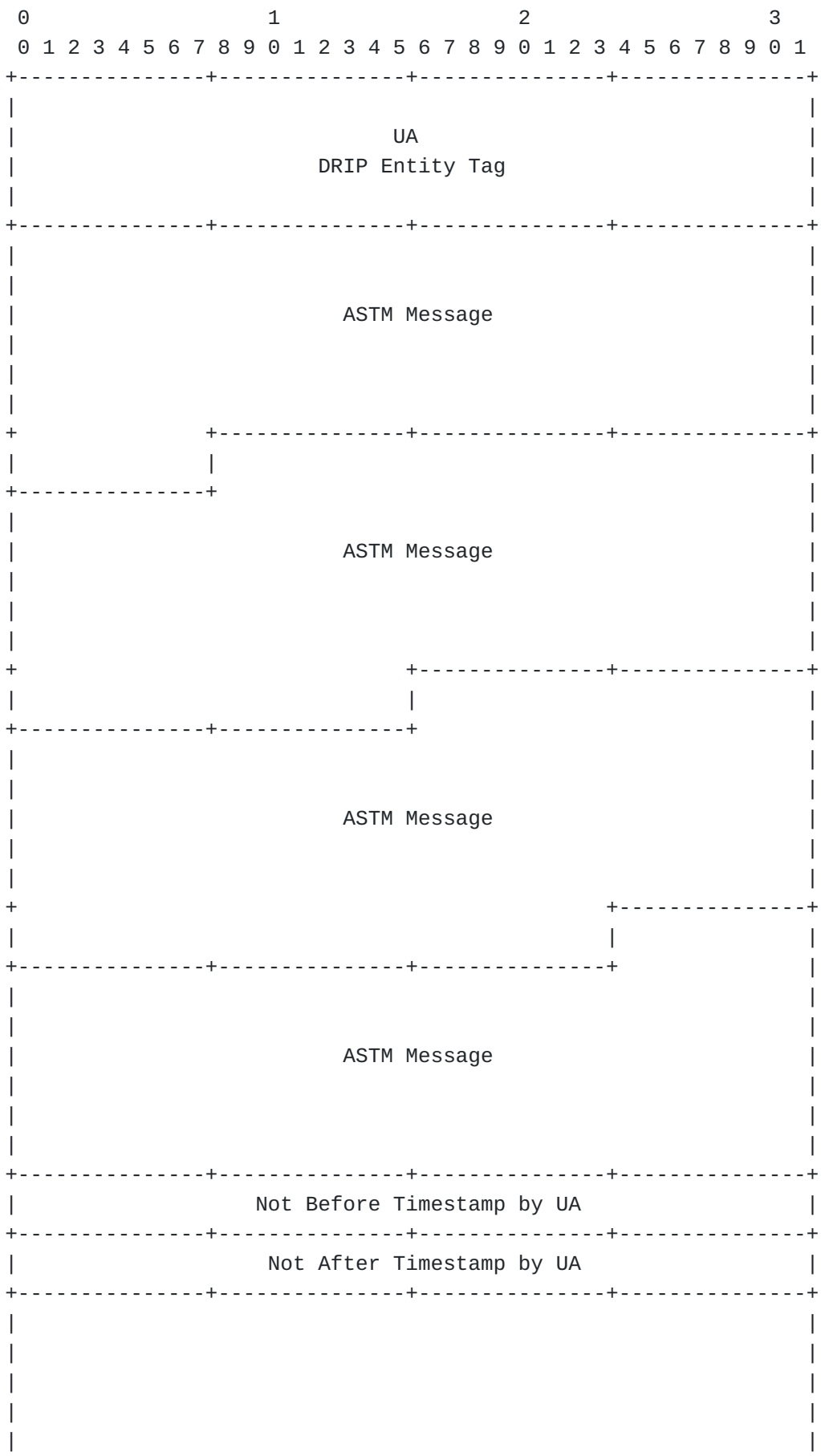
This DRIP Authentication Message is used in conjunction with other DRIP SAM Types (such as Manifest or Wrapper) that contain data that is guaranteed to be unique and easily cross checked by the receiving device. A good candidate for this is using the DRIP Wrapper to encapsulate a Location Message (Message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

5.3. DRIP Wrapper

This SAM Type is used to wrap and sign over a list of other [F3411] Broadcast RID messages. It MUST use the Broadcast Attestation Structure (Section 5.1.1).

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

To determine the number of messages wrapped the receiver can check that the length of the Attestation Data field of the DRIP Broadcast Attestation (Section 5.1.1) is a multiple of 25-bytes.


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
ASTM Message
ASTM Message
ASTM Message
ASTM Message
Not Before Timestamp by UA
Not After Timestamp by UA

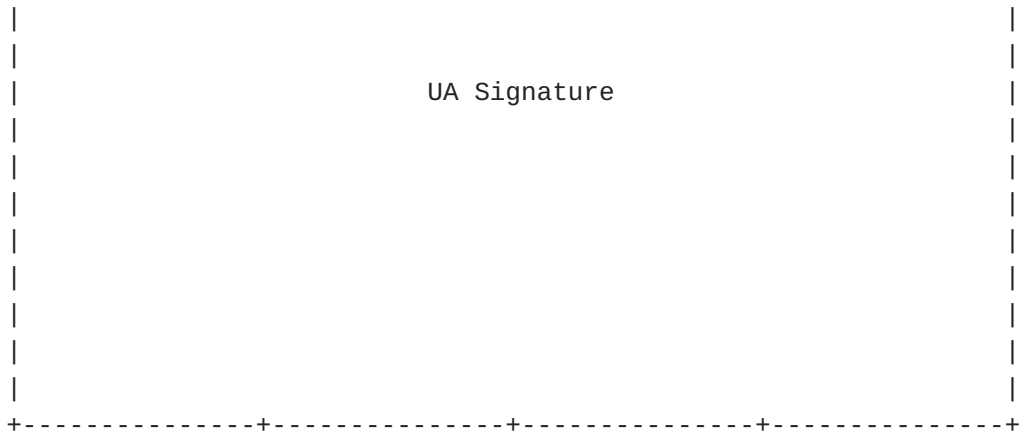

UA Signature

Figure 7: Example 4-Message DRIP Wrapper

5.3.1. 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 Message Pack (0xF). The ASTM Messages are removed from the DRIP Wrapper after signing (as they are redundant) leaving the following structure that is placed into the SAM Authentication Data of Figure 5 and sent in the same Message Pack.

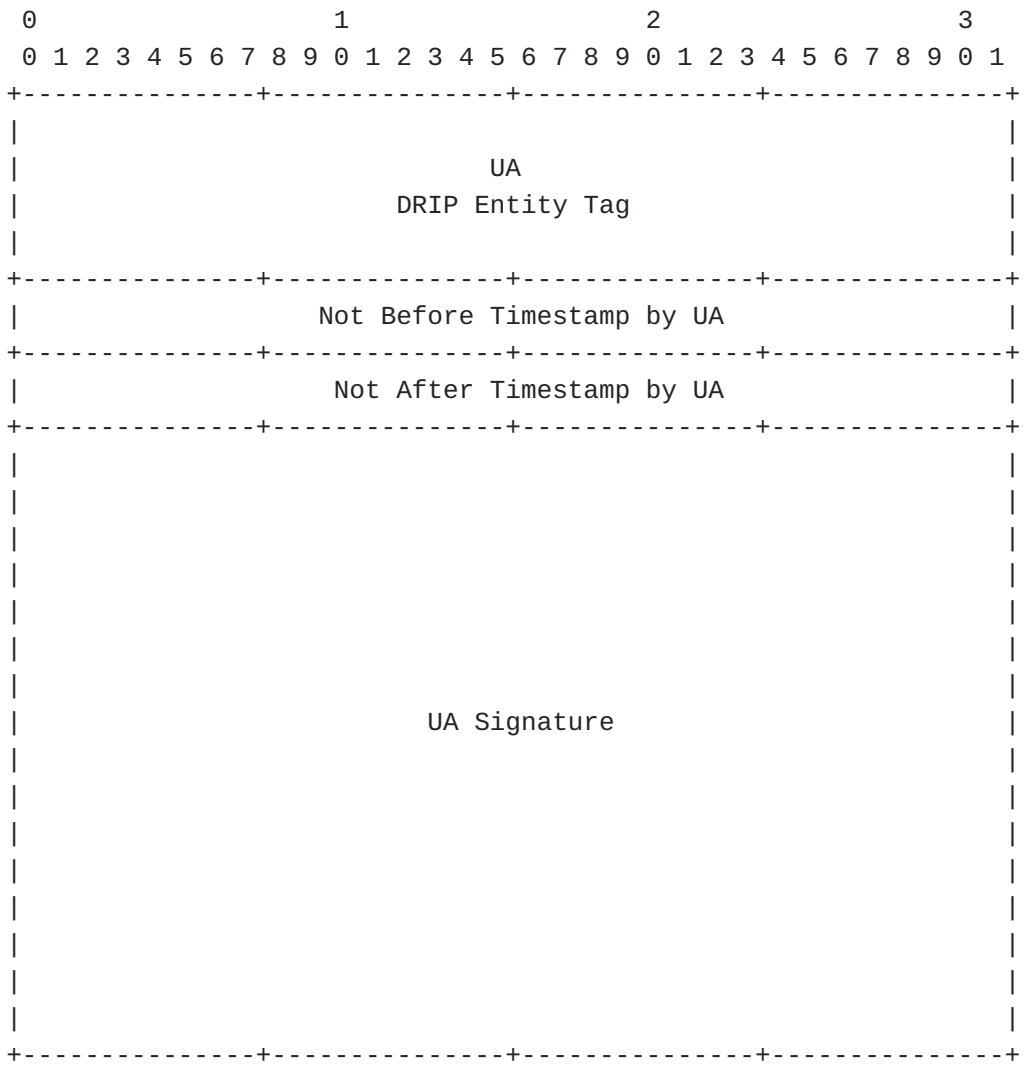


Figure 8: DRIP Wrapper under Extended Transports

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

5.3.2. Wrapper Limitations

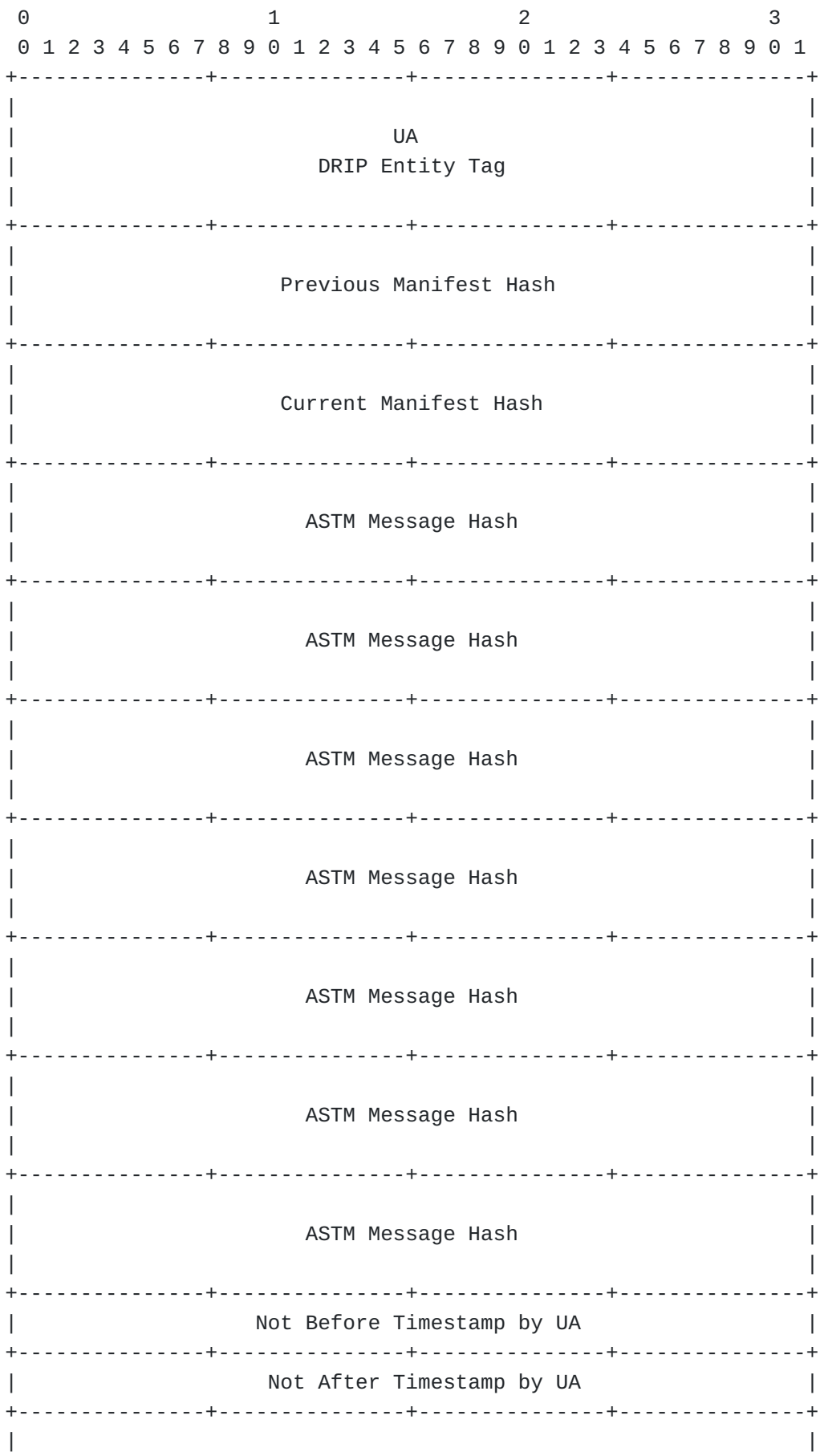
The primary limitation of the Wrapper format is the bounding of up to 4 ASTM Messages that can be sent within it. Another limitation is that the format can not be used as a surrogate for messages it is wrapping. This is due to high potential a receiver on the ground does not support DRIP. Thus when Wrapper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

5.4. DRIP Manifest

This SAM Type is used to create message manifests. It MUST use the Broadcast Attestation Structure (Section 5.1.1).

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

The Attestation Data field is filled with 12-byte hashes of previous [F3411] Broadcast messages. A receiver does not need to have received every message in the manifest to verify it. A manifest SHOULD typically encompass a single transmission cycle of messages being sent, see Section 6.4.


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
Previous Manifest Hash
Current Manifest Hash
ASTM Message Hash
ASTM Message Hash
ASTM Message Hash
ASTM Message Hash
ASTM Message Hash
ASTM Message Hash
ASTM Message Hash
Not Before Timestamp by UA
Not After Timestamp by UA

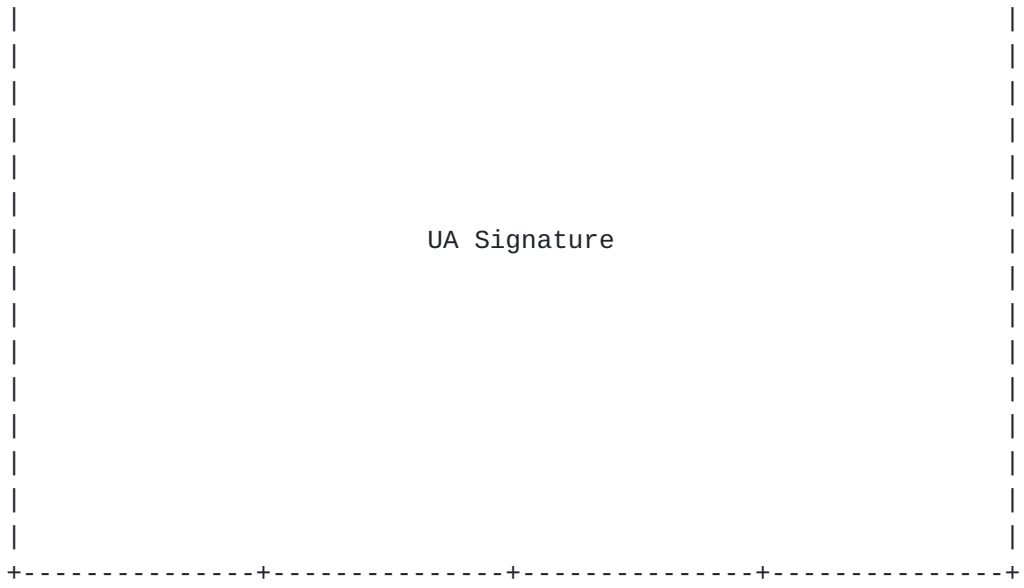

UA Signature

Figure 9: Example DRIP Manifest

5.4.1. Hash Count

The number of hashes in the Manifest can be variable (3-9). An easy way to determine the number of hashes is to take the length of the data between the end of the UA DRIP Entity Tag and Not Before Timestamp by UA and divide it by the hash length (12). If this value is not rational, the message is invalid.

5.4.2. Message Hash Algorithms and Operation

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

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

cSHAKE128(ASTM Message, 96, "", "Remote ID Auth Hash")

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

5.4.2.1. Legacy Transport Hashing

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

5.4.2.2. Extended Transport Hashing

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

5.4.3. Pseudo-Blockchain Hashes

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

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

Cycling:

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

5.4.4. Manifest Limitations

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

Another limitation is the length of hash, which is discussed in Section 9.1.

5.5. DRIP Frame

This SAM Type is for when the authentication data does not fit in other defined formats under DRIP and is reserved for future expansion under DRIP if required. This SAM Type MUST use the Broadcast Attestation Structure (Section 5.1.1).

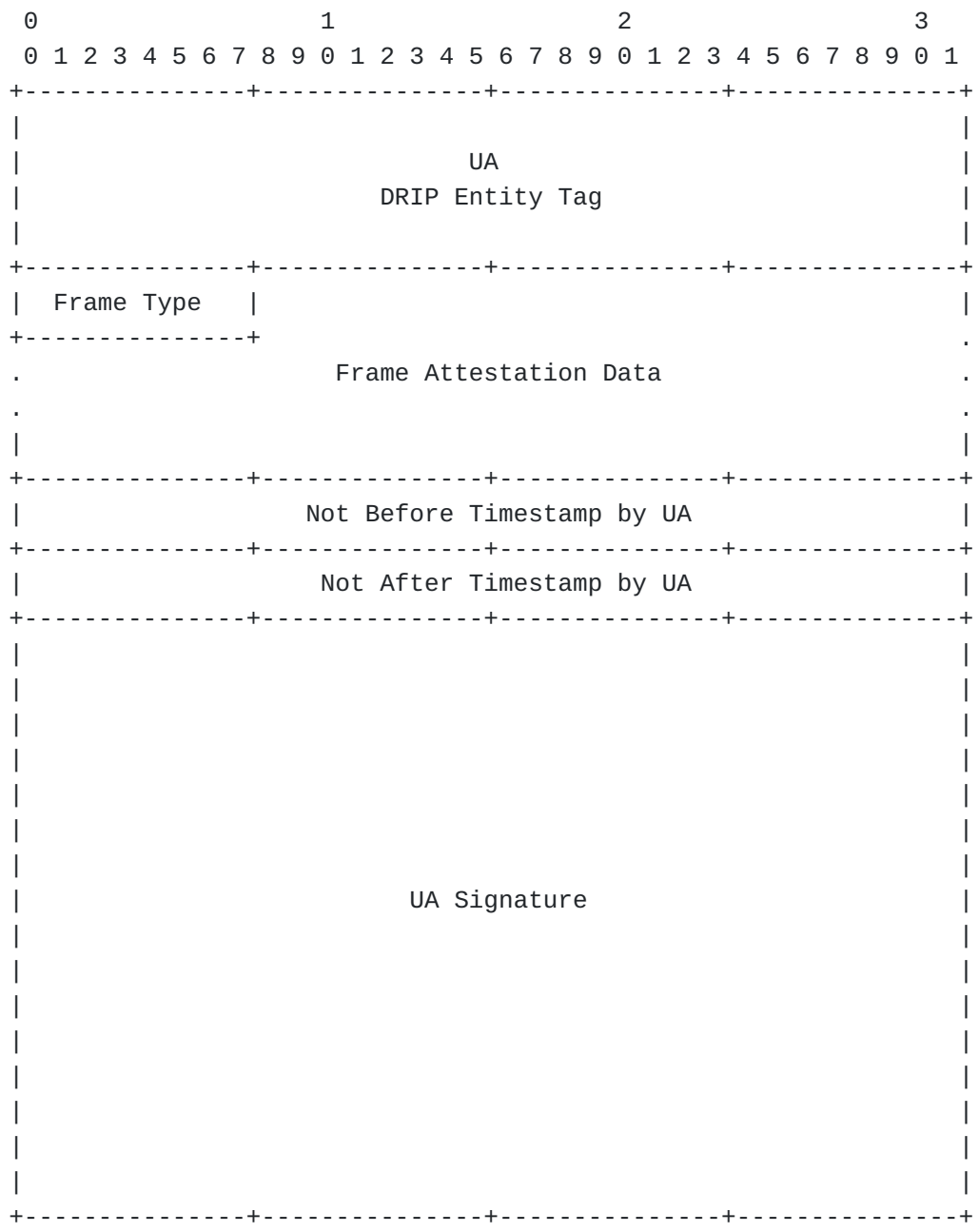


Figure 10: Example DRIP Frame

5.5.1. Frame Type

Byte to sub-type for future different DRIP Frame formats. It takes the first byte of Attestation Data in Section 5.1.1 leaving 111-bytes for Frame Attestation Data.

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

Table 3

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 4) MUST be used, per mandated Remote ID rules (for example the US FAA Remote ID Rule [faa-rid]), when using Legacy Advertising methods (such as Bluetooth 4).

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

6.2. Extended Transports

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

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

6.3. Authentication

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

1. DRIP Link using the Broadcast Attestation of HDA and the UA (satisfying GEN-1 and GEN-3)

2. Any other DRIP Authentication Format (RECOMMENDED: DRIP Manifest or DRIP Wrapper) where the UA is dynamically signing data that is guaranteed to be unique and easily cross checked by the receiving device (satisfying GEN-1 and GEN-2)

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

1. DRIP Link using the Broadcast Attestation of HID Root and the RAA (CAA) (satisfies GEN-3)

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

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

4. Any other DRIP Authentication Format (RECOMMENDED: DRIP Manifest or DRIP Wrapper) where the UA is dynamically signing data that is guaranteed to be unique and easily cross checked by the receiving device (satisfying GEN-1 and GEN-2)

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 HDA UA assertion.

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

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

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 format would seem to be a more efficient use of the transport channel.

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

7. ICAO Considerations

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

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

8. IANA Considerations

8.1. Update IANA DRIP Registry

This document requests a new subregistry for Frame Type.

DRIP Frame Type: This 8-bit valued subregistry is for Frame Types to be later allocated. 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

Table 4

9. Security Considerations

9.1. Manifest Hash Length

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

1. Hash lengths smaller than 8-bytes (for example 4-bytes) were originally contemplated but ruled out by comments by various cryptographers. The main concern raised in this forum was that the length of hash would not provide strong resistance against collision rate. The authors also after further review agreed with this and also realized operationally it was not

necessarily viable. While 4-byte hashes would allow more messages to be filled into a single DRIP Manifest payload (up to 22 individual hashes) the length of time for the UA to stay in a single place where the Observer would receive all the originally messages to rehash to verify such a message was impractical.

2. Hash lengths larger than 8-bytes (for example 12 or 16-bytes) were also considered by the authors. These got the approval of the cryptographers but the number of hashes to send became much lower (only 5 individual hashes). While this lower number is a more reasonable number of original messages the Observer would have to capture it would also mean that potentially more DRIP Manifests would need to be sent. Overall the increase length of the hash did not operationally justify the cost.

3. Simplifying the current design and locking it into using the same hash as the HHIT instead of allowing for agility in either hash algorithm or length seemed more realistic to the authors today.

9.2. Replay Attacks

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

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

2. It is RECOMMENDED to send more than just DRIP Link messages, specifically those that sign over changing data using the current session keypair, and those messages are sent more frequently. An UA beaconing these messages then actually signing other messages using the keypair validates the data receiver by an Observer. An UA who does not either run DRIP themselves or does not have possession of the same private key, would be clearly exposed upon signature verification.

9.3. Trust Timestamp Offsets

Note the discussion of Trust Timestamp Offsets here is in context of the DRIP Wrapper (Section 5.3) and DRIP Manifest (Section 5.4) messages. For DRIP Link (Section 5.2) messages these offsets are set by the Attestor (typically a registry) and have their own set of considerations as seen in [drip-registries].

The offset of the Trust Timestamp (defined as a very short Expiration Timestamp) is one that needs careful consideration for any implementation. The offset should be shorter than any given flight duration (typically less than an hour) but be long enough to be received and processed by Observers (larger than a few seconds). It 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) - this drove the requirement for max page length of Authentication Data itself.

11. References

11.1. Normative References

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

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

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

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

11.2. Informative References

[drip-registries] Wiethuechter, A., Card, S., and R. Moskowitz, "DRIP Registries", Work in Progress, Internet-Draft, draft-wiethuechter-drip-registries-01, 22 October 2021, <https://www.ietf.org/archive/id/draft-wiethuechter-drip-registries-01.txt>.

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

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

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

Appendix A. Authentication State Diagrams & Color Scheme

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

There are two keys of interest: the PK of the UA and the PK of the HDA (or Registry). This document gives a clear way to send the PK of the UA over the Broadcast RID messages - however the PK of the Registry is not. It can be using the same mechanism but is not required to do so due to potential operational constraints and implementation of a given UA transmitter. As such there are scenarios where you may have part of the key-chain but not all of it.

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

A.1. State Table

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

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

Table 5

A.2. State Diagrams

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

A.2.1. Notations

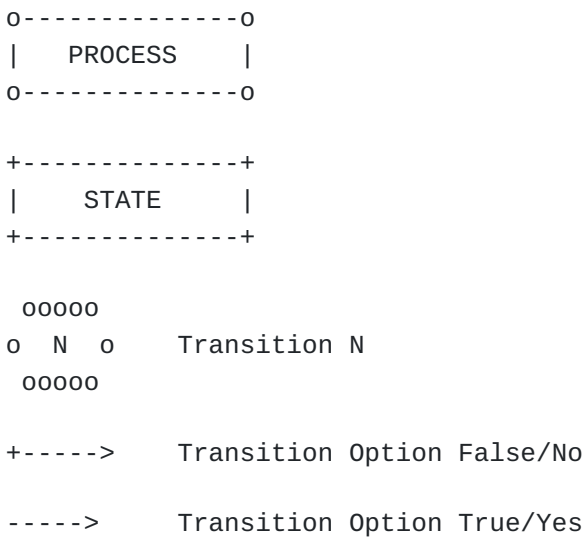


Figure 11: Diagram Notations

A.2.2. General

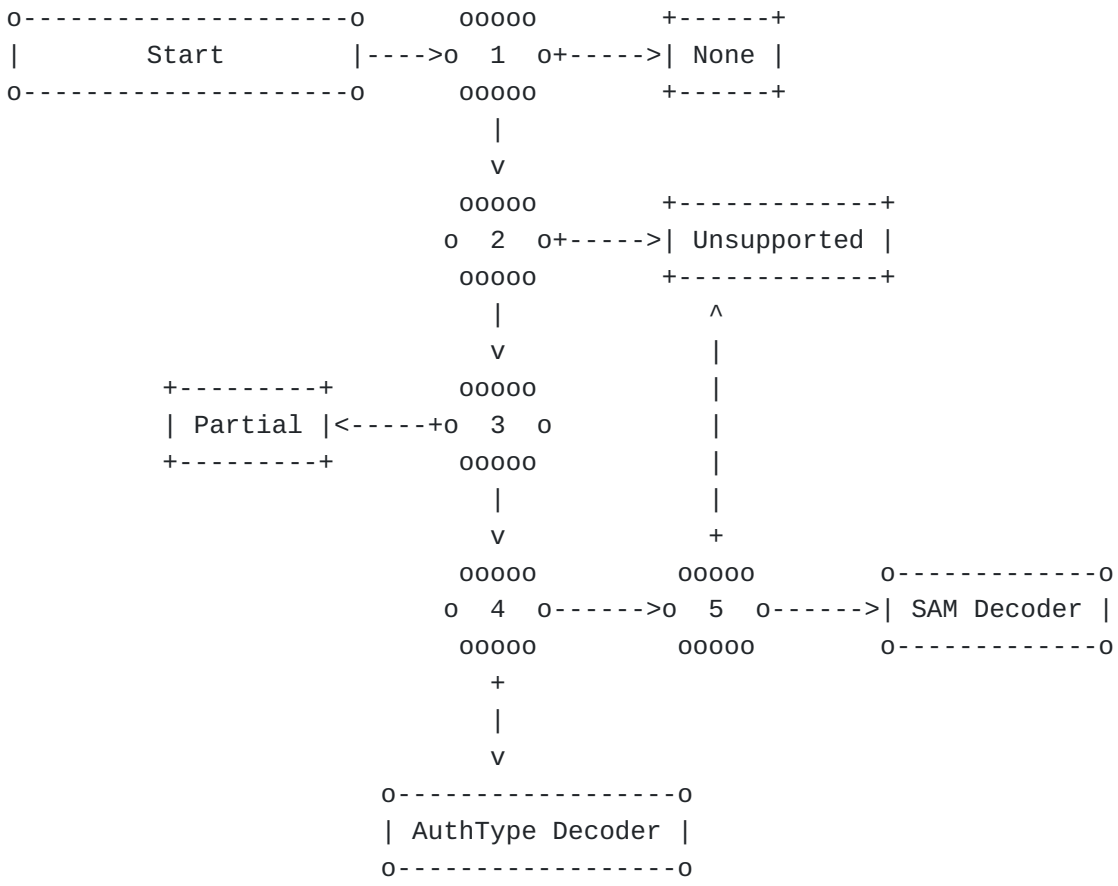


Figure 12: Standard Authentication Colors/State

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

Table 6

A.2.3. DRIP SAM

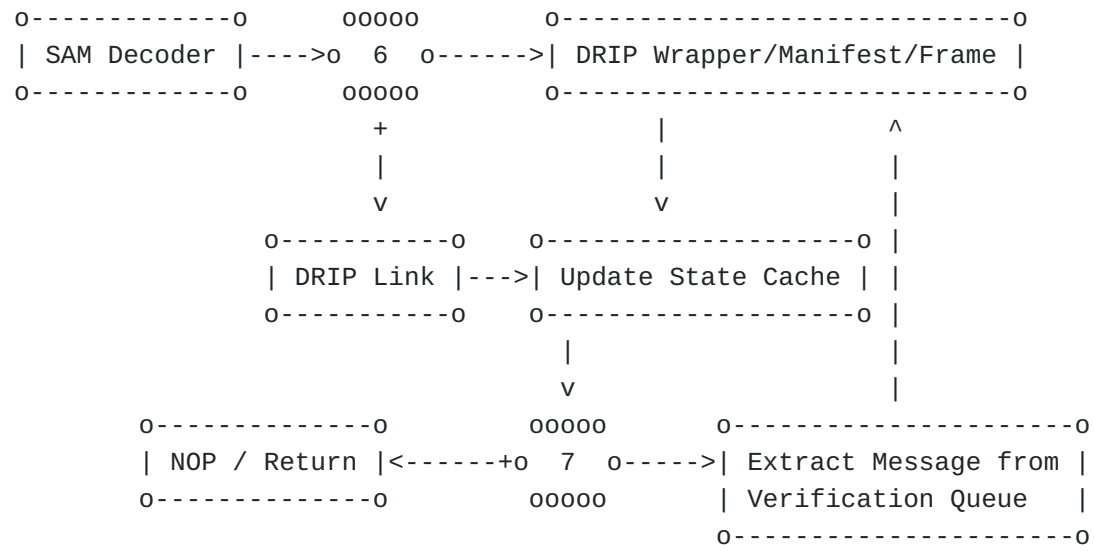


Figure 13: DRIP SAM Decoder

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

Table 7

A.2.4. DRIP Link

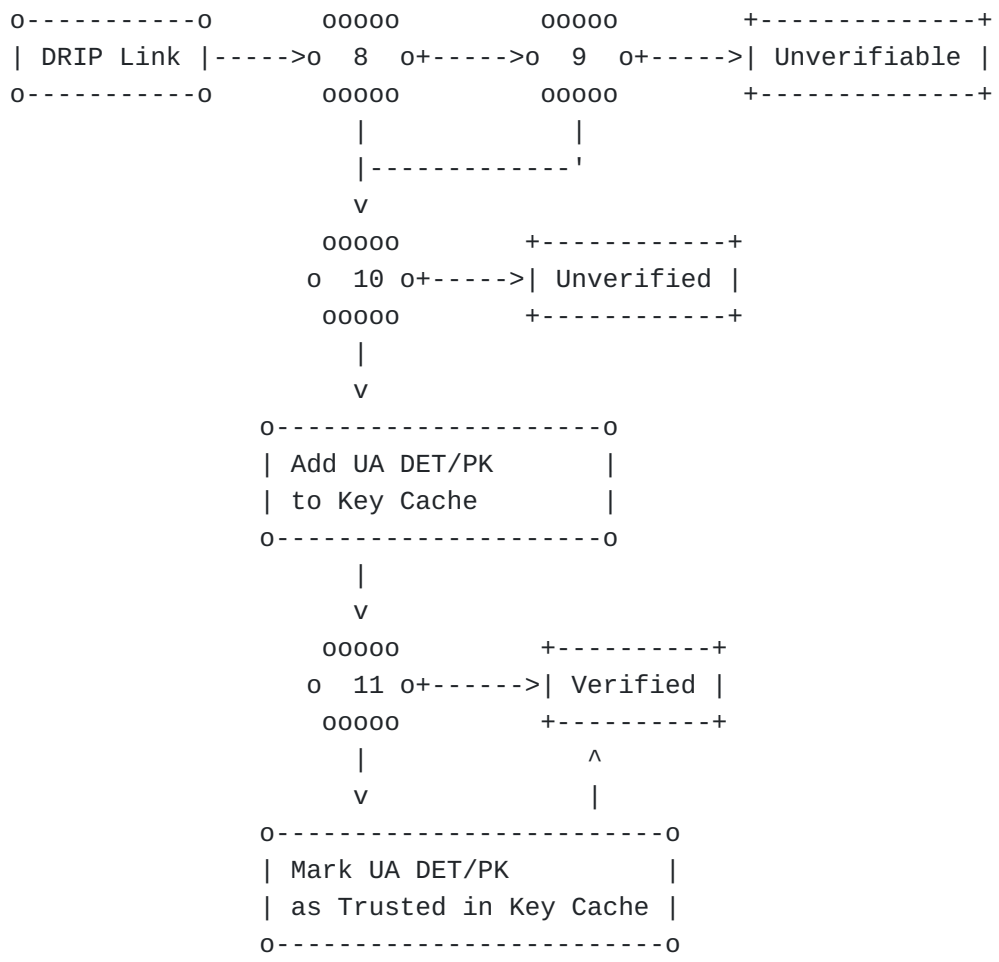


Figure 14: DRIP Link State Decoder

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

Table 8

A.2.5. DRIP Wrapper/Manifest/Frame

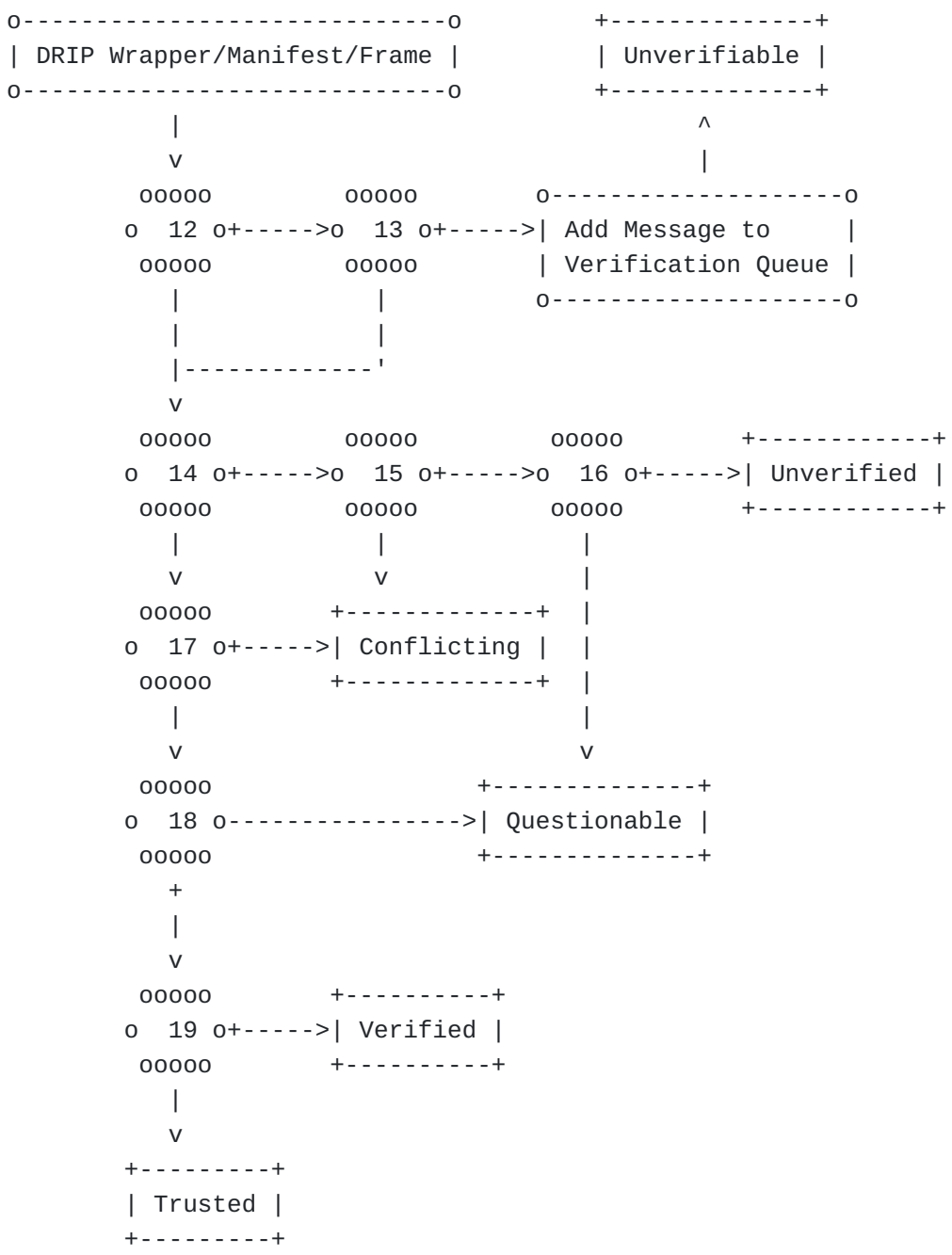


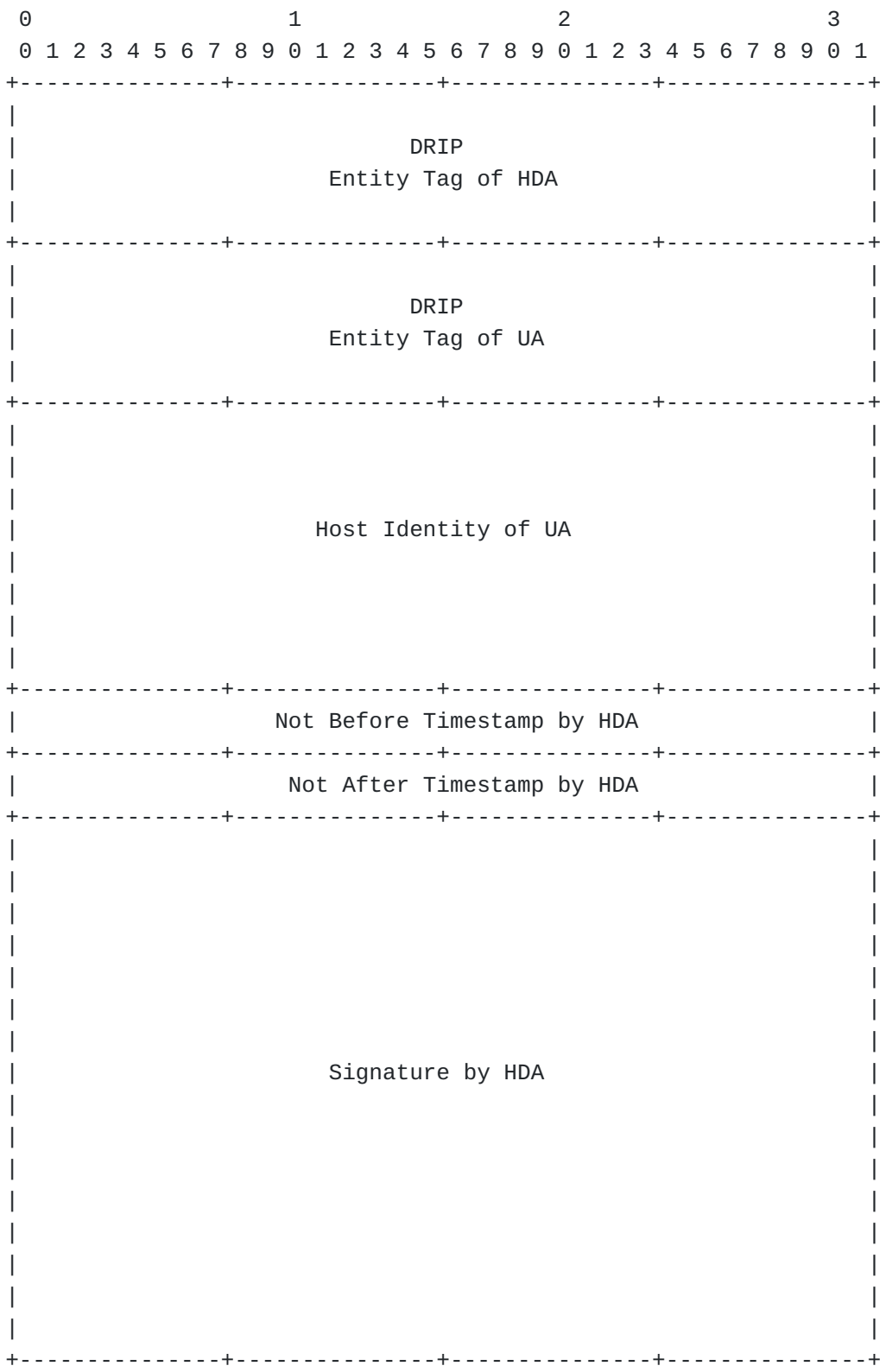
Figure 15: DRIP Wrapper/Manifest/Frame State Decoder

Transition	Transition Query	Next State/Process/ Transition (Yes, No)
12	UA DET/PK in Key Cache?	14, 13
13	UA PK found Online?	14, Add Message to Verification Queue
14	UA Signature Verified?	17, 15
15	Has past Messages of this type been marked as Trusted?	Conflicting, 16
16		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 9

Appendix B. HDA-UA Broadcast Attestation



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

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

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

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

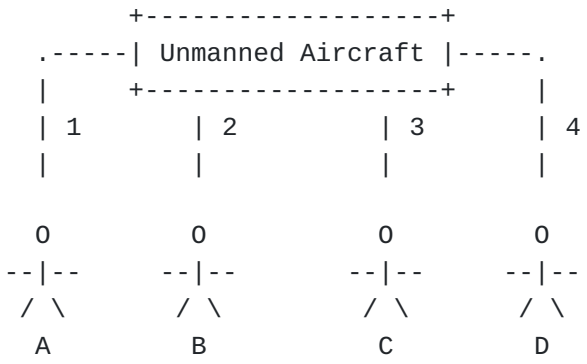
Signing Timestamp by HDA (4 bytes):
    Current time at signing.

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

Figure 16: Example DRIP HDA-UA Broadcast Attestation

Appendix C. Example TX/RX Flow

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



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

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

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

Authors' Addresses

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

Email: adam.wiethuechter@axenterprize.com

Stuart Card
AX Enterprize, LLC

4947 Commercial Drive
Yorkville, NY 13495
United States of America

Email: stu.card@axenterprize.com

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

Email: rgm@labs.htt-consult.com