

DRIP Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 20 June 2021

A. Wiethuechter  
S. Card  
AX Enterprize, LLC  
R. Moskowitz  
HTT Consulting  
17 December 2020

DRIP Authentication Formats  
draft-wiethuechter-drip-auth-06

Abstract

This document describes how to include trust into the ASTM Remote ID specification defined in ASTM F3411-19 under a Broadcast Remote ID (RID) scenario. It defines a few different message schemes (based on the Authentication Message) that can be used to assure past messages sent by a UA and also act as an assurance for UA trustworthiness 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 20 June 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text

as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. DRIP Requirements Addressed . . . . .	3
2. Terminology . . . . .	3
2.1. Required Terminology . . . . .	3
2.2. Definitions . . . . .	3
3. Background . . . . .	3
3.1. Problem Space and Focus . . . . .	4
3.2. ASTM Authentication Message . . . . .	4
4. DRIP Authentication Framing Formats . . . . .	6
4.1. DRIP General Frame . . . . .	6
4.1.1. DRIP Header . . . . .	8
4.1.2. DRIP Authentication Data . . . . .	8
4.1.3. Forward Error Correction . . . . .	9
4.2. DRIP Wrapper Frame . . . . .	10
4.2.1. UA Hierarchical Host Identity Tag . . . . .	11
4.2.2. Trust Timestamp . . . . .	11
4.2.3. Wrapped Authentication Data . . . . .	12
4.2.4. Wrapper Signature . . . . .	16
4.3. DRIP Attestation Frame . . . . .	16
4.3.1. Attestation Data . . . . .	17
4.3.2. Expiration Timestamp . . . . .	18
4.3.3. Attestation Signature . . . . .	18
5. Transport Methods & Recommendations . . . . .	18
5.1. Legacy Advertisements (Bluetooth 4.X) . . . . .	18
5.2. Extended Advertisements (Bluetooth 5.X and Wifi NaN) . . . . .	19
6. ASTM Considerations . . . . .	19
7. IANA Considerations . . . . .	20
8. Security Considerations . . . . .	20
9. Acknowledgments . . . . .	20
10. Appendix A: Thoughts on ASTM Authentication Message . . . . .	20
11. References . . . . .	21
11.1. Normative References . . . . .	21
11.2. Informative References . . . . .	21
Authors' Addresses . . . . .	22

## 1. Introduction

UA 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 standard [F3411-19] focuses on two ways of communicating to a UAS for RID: Broadcast and Network.

This document will focus on adding trust to Broadcast RID in the current (and an expanded) Authentication Message format.

### 1.1. DRIP Requirements Addressed

The following [drip-requirements] will be addressed:

GEN 1: Provable Ownership This will be addressed using the Certificate Message type (Section 4.3.1.1).

GEN 2: Provable Binding This requirement is addressed using the Wrapped ASTM Message (Section 4.2.3.1.2), Manifest Message (Section 4.2.3.2) and Message Pack Signature (Section 4.2.3.1.1) types.

GEN 3: Provable Registration This requirement is addressed using the Certificate Message type (Section 4.3.1.1).

## 2. Terminology

### 2.1. Required Terminology

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

### 2.2. Definitions

See [drip-requirements] for common DRIP terms.

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

## 3. Background

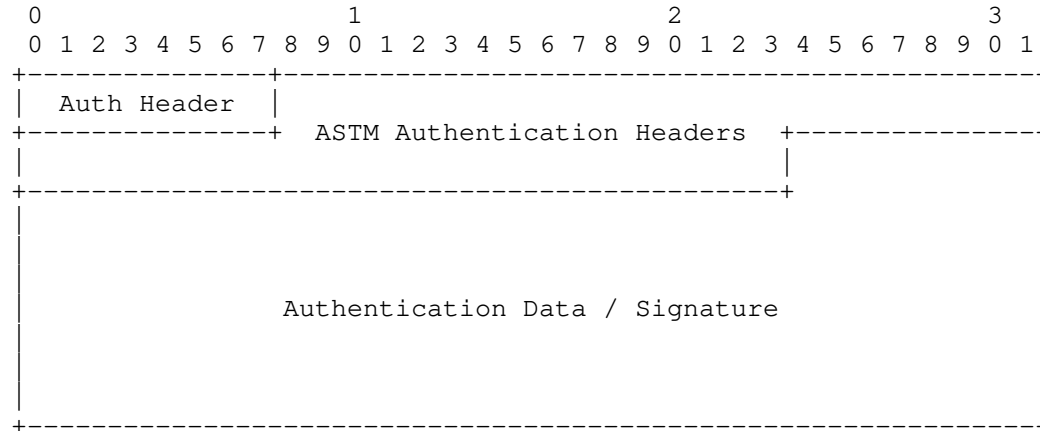
### 3.1. Problem Space and Focus

The current standard for Remote ID (RID) 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.

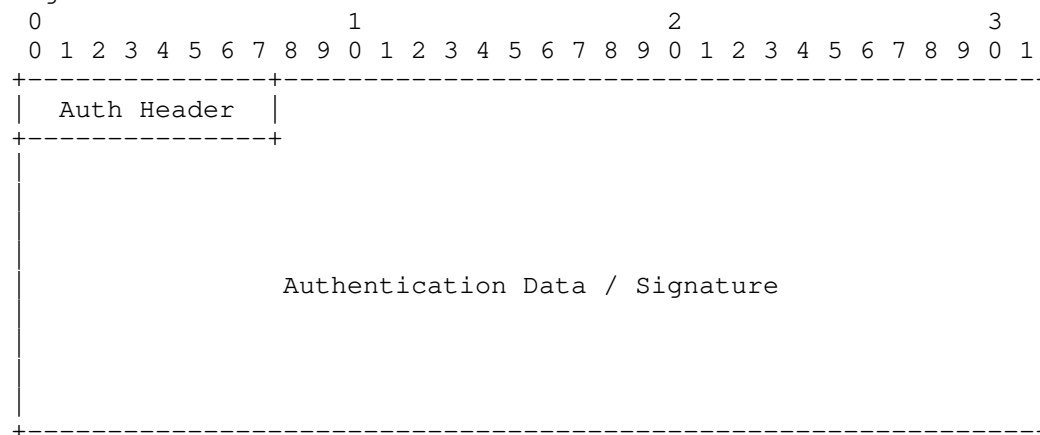
The following subsections will provide a high level reference to the ASTM standard for Authentication Messages and how their current limitations effect trust in the Broadcast RID environment.

### 3.2. ASTM Authentication Message

Page 0:



Page 1 - 4:



Auth Header (1 byte):

Contains Authentication Type (AuthType) and Page Number. For DRIP Authentication AuthType is a value of 0x5.

ASTM Authentication Headers: (6 bytes)

Contains other header information for the Authentication Message from ASTM UAS RID Standard.

Authentication Data / Signature: (109 bytes: 17+23\*4)

Opaque authentication data.

Figure 1: Standard ASTM Authentication Message format

The above diagram is the format defined by ASTM [F3411-19] that is the frame which everything this document fits into. The specific details of the ASTM headers are abstracted away as they are not necessarily required for this document.

There is a 25th byte exclude in the diagrams that comes before the Auth Header. This is the ASTM Header and consists of the Protocol Version and Message Type of the given message frame/page.

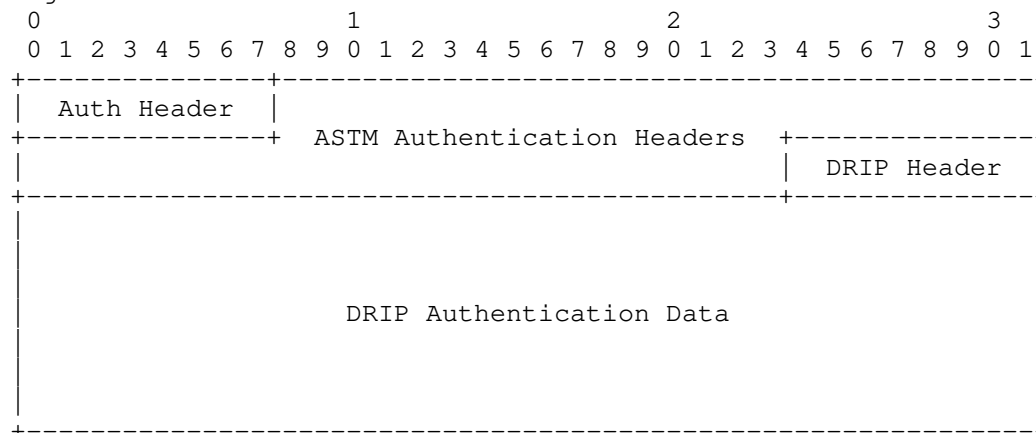
#### 4. DRIP Authentication Framing Formats

Currently the ASTM AuthType of 0x5 should be used to denote DRIP based Authentication. The max page count of the Authentication Message is increased to 10, instead of being capped at 5.

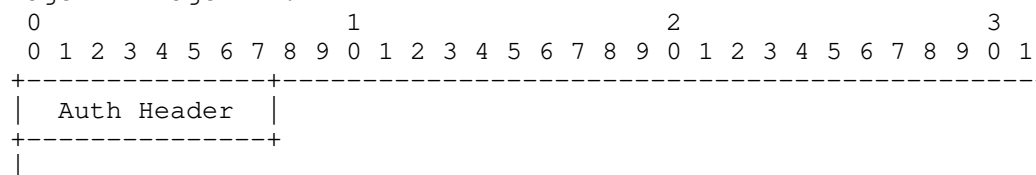
To keep consistent formatting across the different mediums (Bluetooth 4, Bluetooth 5 and Wifi NaN) and their independent restrictions the authentication data being sent is REQUIRED to fit within the first 9 pages of the Authentication Message. The final (10th) page of the message is reserved exclusively for Forward Error Correction bytes and is only present on Bluetooth 4.

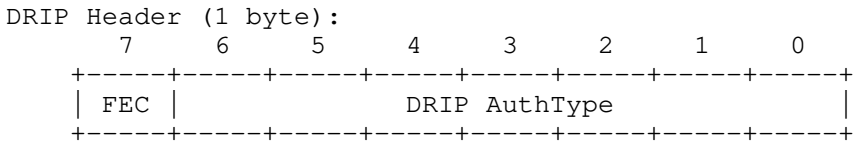
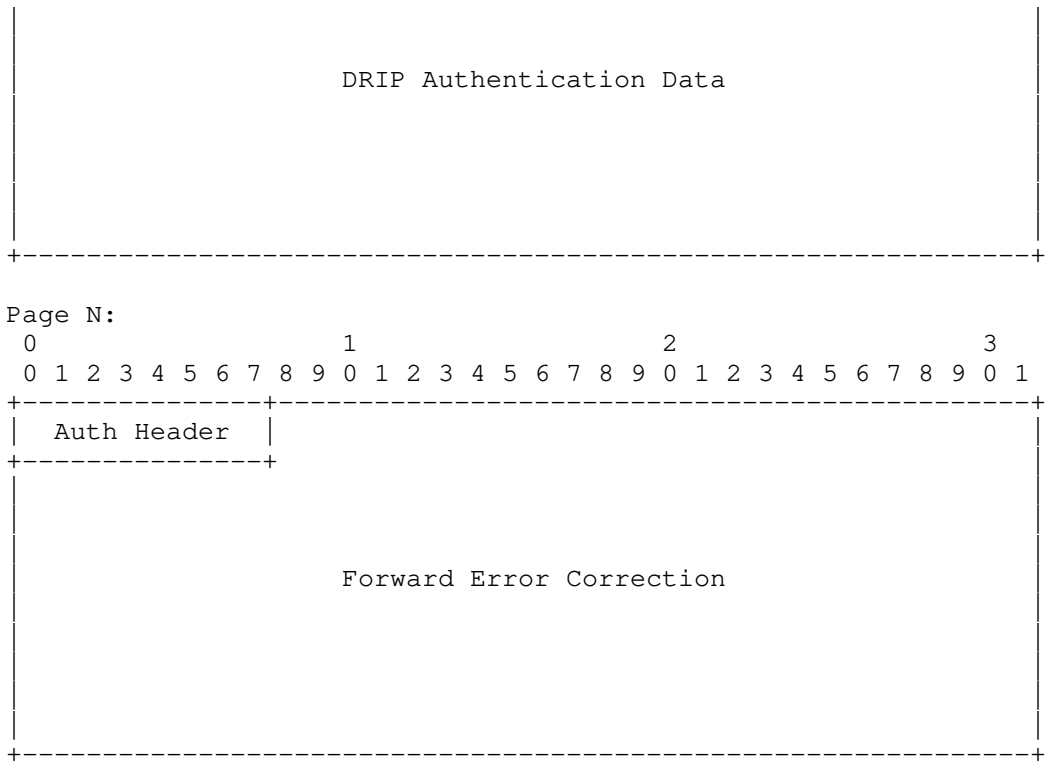
##### 4.1. DRIP General Frame

Page 0:



Page 1 - Page N-1:





FEC (1 bit):  
Enabled [1] or Disabled [0]. Signals if Page N is filled with XOR FEC.

DRIP AuthType (7 bits):	
DRIP AuthType	Values
-----	
0 Wrapped ASTM Message(s)	0
1 Wrapped ASTM Message(s)	1
2 Wrapped ASTM Message(s)	2
3 Wrapped ASTM Message(s)	3
4 Wrapped ASTM Message(s)	4
5 Wrapped ASTM Message(s)	5
8 Byte Manifest	6
4 Byte Manifest	7

Reserved (Wrapped Messages)	8-15
Certificate: Registry on Aircraft	16
Reserved (Certificates)	17-31
Private Use	32-63
Reserved	64-111
Experimental Use	112-127

DRIP Authentication Data (200 bytes):

DRIP Authentication data. 0 to 200 bytes.

Forward Error Correction (23 bytes):

Optional and signaled using DRIP Header. Always last Authentication page.

Figure 2: DRIP General Frame Format

#### 4.1.1. DRIP Header

The DRIP Header is used to signal what kind of Authentication under DRIP that the message is using and consists of two fields.

##### 4.1.1.1. Forward Error Correction (Bit 8)

The Most Significant Bit is used to signal if FEC is present in the final page of the Authentication Message. It MUST be set to 1 if FEC is being used. This is only enabled under Bluetooth 4 and MUST be set to 0 on Bluetooth 5 or Wifi NaN.

##### 4.1.1.2. DRIP AuthType (Bits 1-7)

The lower 7 bits are used as the DRIP AuthType field denoting what Authentication type is being used. There are 5 major areas carved out of the DRIP AuthType defined by the following bitmaps:

```

000 xxxx (0x00-0x0F): Wrapped Messages (16)
001 xxxx (0x10-0x1F): Certificates (16)
01x xxxx (0x20-0x3F): Private Use (32)
1xx xxxx (0x40-0x6F): Reserved (48)
111 xxxx (0x70-0x7F): Experimental Use (16)

```

Figure 3: DRIP Header Bitmasks

#### 4.1.2. DRIP Authentication Data

This field has a maximum size of 200 bytes. If the data is less than the max and a page is only partially filled then the rest of the partially filled page must be null padded.



This section is generally filled with either the Wrapper Frame (Section 4.2) or the Attestation Frame (Section 4.3).

#### 4.1.3. Forward Error Correction

To help Bluetooth (specifically Bluetooth 4) achieve the goal of reliable receipt of paged messages a Forward Error Correction (FEC) scheme is introduced and MUST be used for Legacy Advertising (Bluetooth 4) and MUST NOT be used for Extended Advertising (Bluetooth 5, Wifi NaN) under DRIP.

##### 4.1.3.1. Encoding

A compliant implementation of this standard MUST use XOR for the FEC. When generating the parity the first byte of every Authentication Page MUST be excluded from the XOR operation. For pages 1 through N this leaves the data portion of the page while page 0 will include a number of headers along with 17 bytes of data.

To generate the parity a simple XOR operation using the previous and current page is used. For page 0, a 23 byte null pad is used for the previous page. The resulting 23 bytes of parity is appended in one full page (always the last) allowing for recovery when any single page is lost in transmission.

##### 4.1.3.2. 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 Authentication Data field (and ASTM Authentication Headers fields if page 0) with nulls.

Using the same methods as encoding, an XOR operation is used between the previous and current page (a 23 byte null pad is used when page 0 is the current page). The resulting 23 bytes is the data of the missing page.

If page 0 is being reconstructed an additional check of the Page Count, to check against how many pages are actually present, MUST be performed for sanity. An additional check on the Data Length field can also be performed, but is not required.

#### 4.1.3.3. Limitations & Recommendations

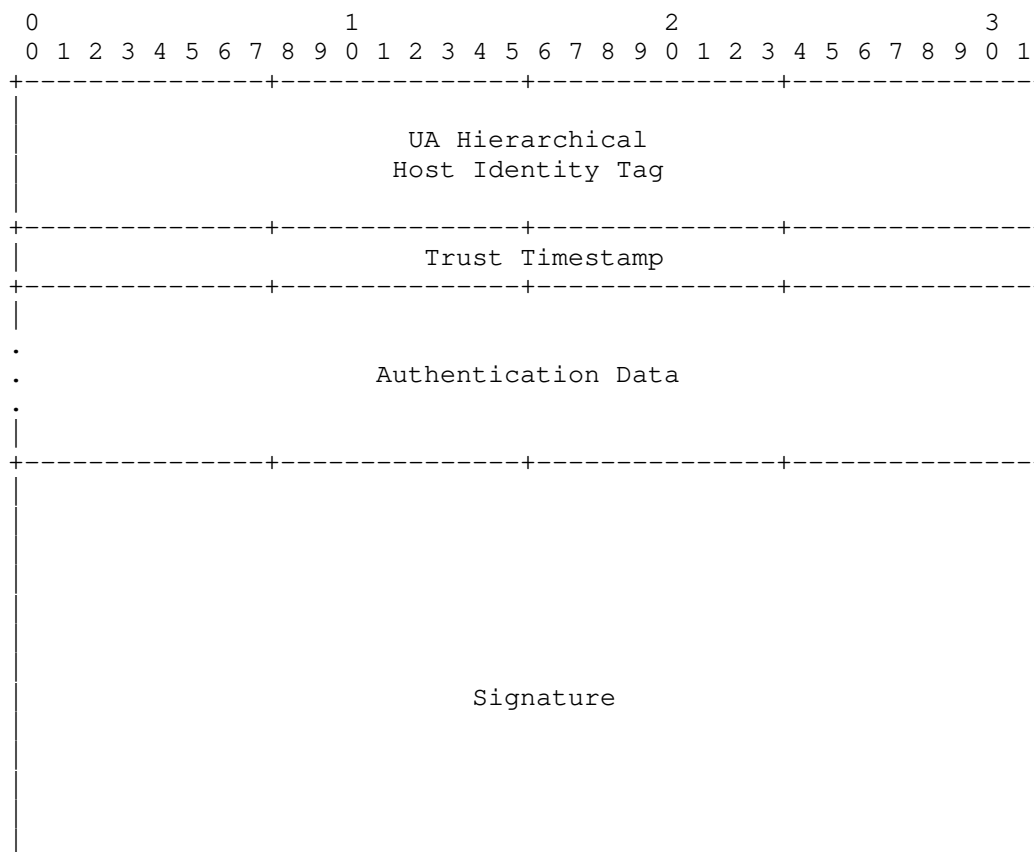
If more than one page is lost ( $>1/5$  for 5 page messages,  $>1/10$  for 10 page messages) than the error rate of the link is already beyond saving and the application has more issues to deal with.

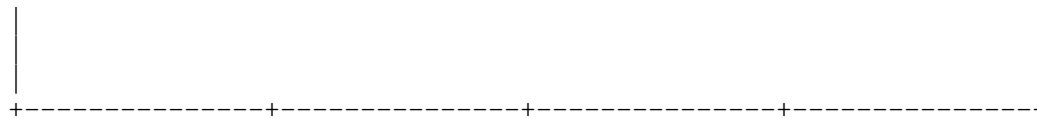
In theory under Bluetooth 4 up to 15 pages Authentication could be sent (9 pages reserved to Authentication and 6 pages reserved for Forward Error Correction). It is currently recommended however for a max of 10 pages total.

#### 4.2. DRIP Wrapper Frame

This format MUST be encapsulated by the General Frame (Section 4.1) and reside in its data field (Section 4.1.2).

Typically the DRIP Header is set in the range of 0x00 through 0x0F (FEC disabled) or 0x80 through 0x8F (FEC enabled).





UA Hierarchical Host Identity Tag (16 bytes):

The UAs HHIT in byte form. Hashed from the EdDSA25519 public key.

Trust Timestamp (4 bytes):

Timestamp denoting current time plus an offset to trust message to.

Authentication Data (116 bytes):

Opaque authentication data using DRIP format specified in the DRIP Header. 0 to 116 bytes.

Signature (64 bytes):

Signature over preceding fields using the EdDSA25519 keypair of the UA.

Figure 4: DRIP Wrapper Frame Format

#### 4.2.1. UA Hierarchical Host Identity Tag

To avoid needing the UAs HHIT via the ASTM Basic ID in a detached fashion the 16 byte HHIT of the UA is included in the wrapper frame.

The HHIT for the UA (and other entities in the RID and greater UTM system under DRIP) is an enhancement of the Host Identity Tag (HIT) [RFC7401] introducing hierarchy (and how they are used in UAS RID) as defined in [drip-rid].

#### 4.2.2. Trust Timestamp

The Trust Timestamp is of the format defined in [F3411-19]. That is a UNIX timestamp offset by 01/01/2019 00:00:00. An additional offset is then added to push the timestamp a short time into the future to avoid replay attacks.

When wrapping a Vector (Position/Location) Message the payload WILL contain (by ASTM rules) constantly changing data, this includes its own timestamp. This timestamp is only 2 bytes, which is easily attacked and only expresses the 1/10th of seconds since the last hour.

Other ASTM message types, such as Basic ID and Self-ID are static messages with no changing data. To protect a replay of these signed

messages the Trust Timestamp is the field during signing to be guaranteed to change.

The offset used against the UNIX timestamp is not defined in this document. Best practices to identify a acceptable offset should be used taking into consideration the UA environment, and propagation characteristics of the messages being sent.

#### 4.2.3. Wrapped Authentication Data

This field has a maximum of 116 bytes in length.

##### 4.2.3.1. Wrapped ASTM Message Formats

When wrapping any ASTM Messages and filling the Wrapped Authentication Data field under DRIP the messages MUST be in Message Type order as defined by ASTM. All message types except Authentication (0x2) and Message Pack (0xF) are allowed.

###### 4.2.3.1.1. 0 Wrapped ASTM Message(s)

This payload type MUST only be used under Extended Advertisement (Bluetooth 5.X and Wifi NaN).

The Wrapped Authentication Data is the concatenation of all messages in the Message Pack (excluding Authentication) in Message Type order. No actual data payload is present in this format as the data is found outside the Authentication Message in the same Message Pack.

The DRIP Header is set to 0x00 (0).

###### 4.2.3.1.2. 1 to 4 Wrapped ASTM Message(s)

This payload type can be used on either Legacy or Extended Advertisements.

The DRIP Header is set to 0x81-0x84 (129-134) when using Legacy Advertisements (FEC is enabled) and 0x01-0x04 (1-4) when using Extended Advertisements (FEC is disabled).

###### 4.2.3.1.3. 5 Wrapped ASTM Message(s)

Editors Note: This payload type does not currently fit in the 116 byte limit of the Wrapper Frame. If the ASTM relaxes the Max Page Count limit for Legacy Advertisements to use all 15 pages then this is possible.

This payload type MUST only be used on Legacy Advertisements (Bluetooth 4.X). It requires 11 pages to complete.

The DRIP Header is set to 0x85 (133).

This payload type allows in Legacy Advertisements to have a pseudo-Message Pack like what is found in Extended Advertisements.

#### 4.2.3.1.4. Limitations

When wrapping a single ASTM Message the 25 byte payload actually causes an inefficiency in the framing format, create a whole page unused except for a single byte. This can be optimized by removing a single byte out of the wrapped message but creates an issue on the receiver of knowing which byte was removed.

When sending a Location Message (Message Type 0x1) a single byte can be removed at the end of the message as it is currently unused. Many other messages in the ASTM Message set however do not have this ability. The first byte can not be removed as it is the key to know how to decode the message.

#### 4.2.3.2. Manifests

Manifests fill the Wrapped Authentication Data field with hashes of previously send messages.

By hashing previously sent messages and signing them we gain trust in UAs previous reports. An observer who has been listening for any considerable length of time can hash received messages and cross check against listed hashes.

##### 4.2.3.2.1. Hash Algorithm and Operation

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

A standard HHIT would be using cSHAKE128 from [NIST.SP.800-185]. With cSHAKE128, the hash is computed as follows:

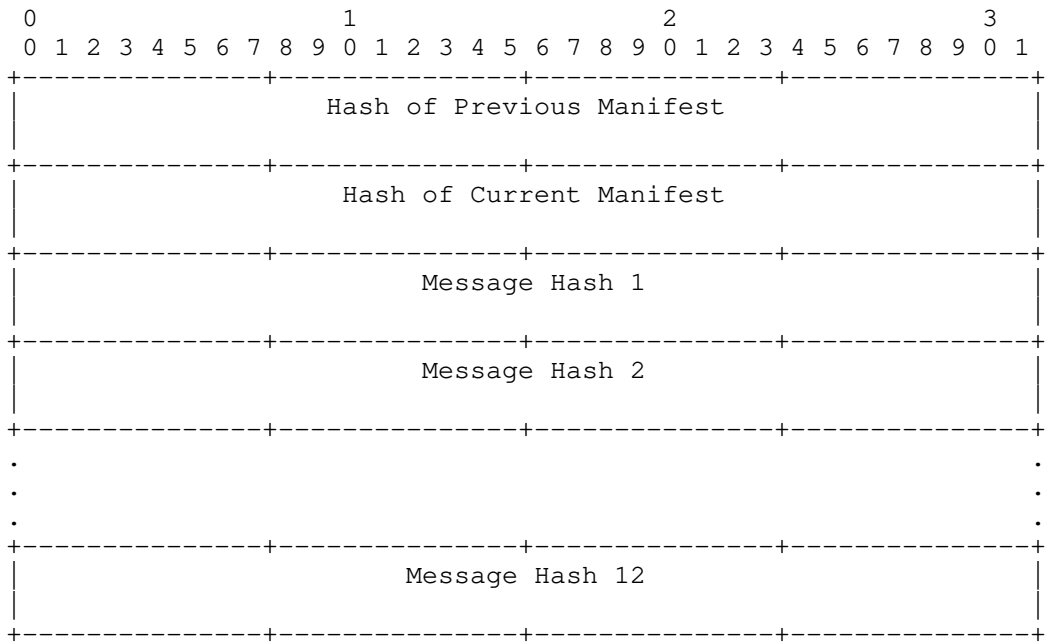
```
cSHAKE128(MAC Address|Message, 8*H-Len, "", "RemoteID Auth Hash")
```

The message MAC Address of the transmitter is prepended to the message, as the MAC Address is the only information that links UA messages from a specific UA.

Editors Note: It should be noted that for Bluetooth mediums this is valid - however Wifi NaN does not give the receiver device the

transmitters MAC Address - making this impossible. Either MAC Address should be removed entirely or something different be used in its place to link to a given UA. Thanks Soren Friis for pointing this out.

4.2.3.2.2. 8 Byte



DRIP Header:  
    With FEC: 0x87 [135] (RECOMMENDED)  
    Without FEC: 0x07 [7]

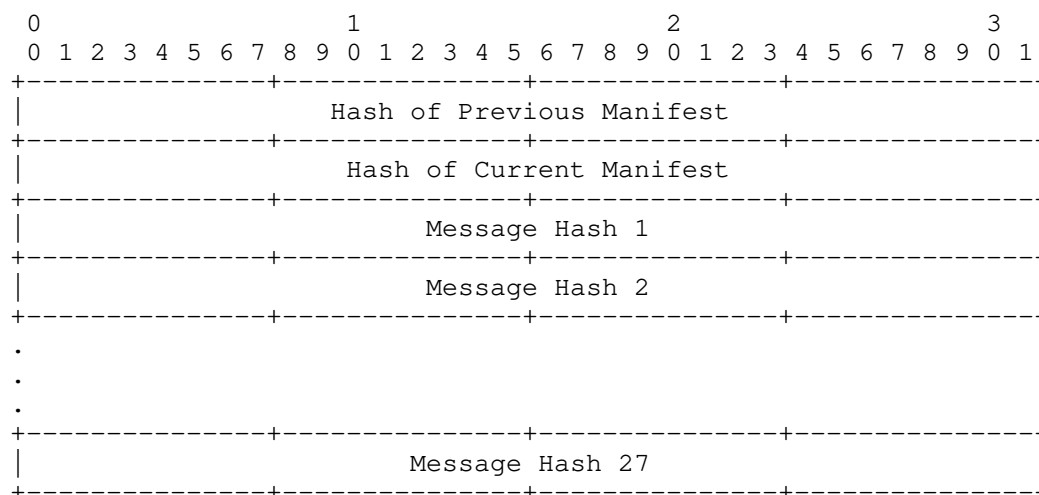
Hash of Previous Manifest: (8 bytes)  
    A hash of the previously sent Authentication message.

Hash of Current Manifest: (8 bytes)  
    A hash of the current Authentication message.

Message Hash: (8 bytes)  
    A hash of a previously sent message. 12 max.

Figure 5: 4 Byte Manifest

4.2.3.2.3. 4 Byte



DRIP Header:

With FEC: 0x86 [132] (RECOMMENDED)

Without FEC: 0x06 [6]

Hash of Previous Manifest: (4 bytes)

A hash of the previously sent Authentication message.

Hash of Current Manifest: (4 bytes)

A hash of the current Authentication message.

Message Hash: (4 bytes)

A hash of a previously sent message. 27 max.

Figure 6: 4 Byte Manifest

#### 4.2.3.2.4. 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.

#### 4.2.3.2.5. Manifest Limitation

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. Without the original messages received no verification can be done. Examples of such scenarios include delivery or survey UA.

#### 4.2.4. Wrapper Signature

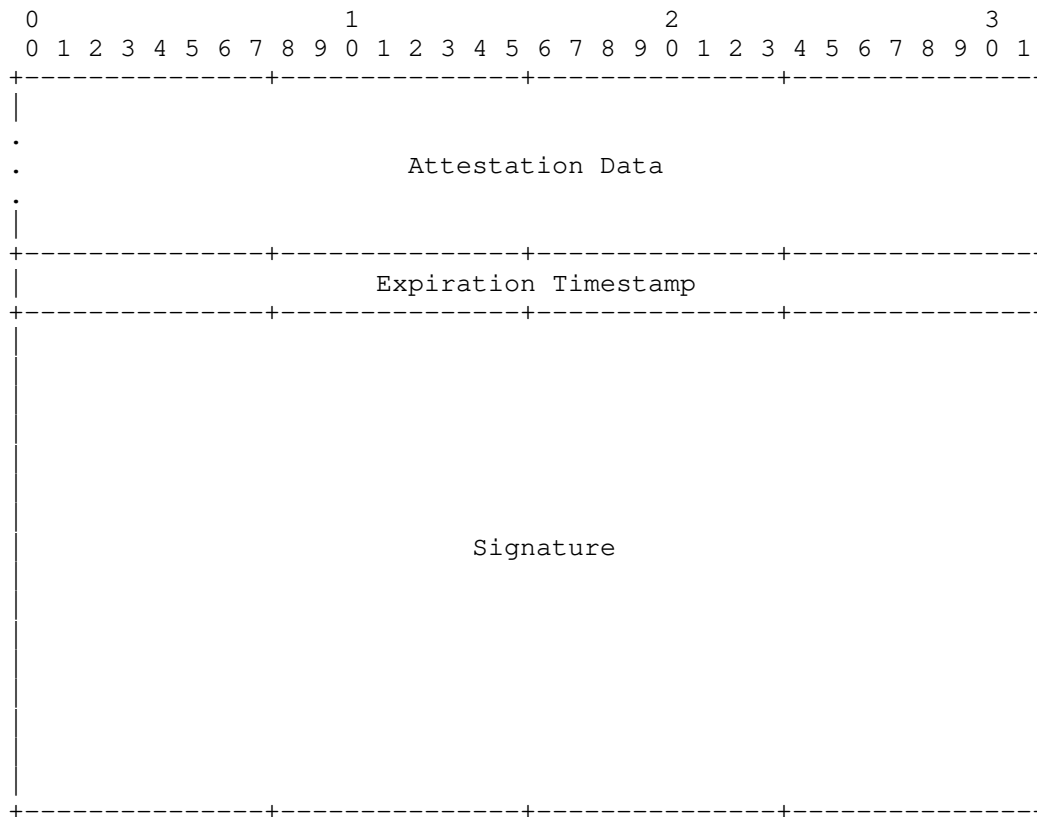
The wrapper signature is generated using the private key half of the the UAs Host Identity (HI) and is done over all preceding data. ASTM/DRIP Headers are exclude from this operation only information within the Wrapper Fame (Section 4.2) is signed.

#### 4.3. DRIP Attestation Frame

This format MUST be encapsulated by the General Frame (Section 4.1) and reside in its data field (Section 4.1.2).

This format is typically used to form a complete certificate using attestation data from a Registry defined in [identity-claims]. The DRIP Header is normally in the range of 0x10 through 0x1F (FEC disable) or 0x90 through 0x9F (FEC enabled).





Attestation Data: (up to 132 bytes):

Data the UA asserts claim to.

Up to 132 bytes in length.

Expiration Timestamp (4 bytes):

Generated by the UA to protect against replay attacks.

Signature (64 bytes):

Signature over preceding fields using the EdDSA25519  
keypair of the UA.

Figure 7: DRIP Attestation Format

#### 4.3.1. Attestation Data

Any data up to 132 bytes in length that the UA wishes to assert truth to.

#### 4.3.1.1. DRIP Certificate

This payload type can be used in either Legacy or Extended Advertising. It is used to grant the ability to authenticate UA Remote ID when the receiving device of the observer (e.g. a smartphone with a dedicated RID application) has no Internet service (e.g. LTE signal).

The DRIP Header is set to 0x90 (144) when used for Legacy Advertisements and 0x10 (16) for Extended Advertisements.

The Attestation Data field is filled with the Attestation: Registry on Aircraft (Section 3.2.2 Attestation: X on Y (Offline Form) from [identity-claims]). This is binding claim between the Registry and the Aircraft, asserting the relationship between the two entities. It also provides the UA Host Identity to allow signature verification of messages signed by the UA. Also included in its structure is the HHIT of the Registry to check the local shortlist of Registries that the Observer device trusts (mapping HHITs to HIs).

More details about this Attestation and other certificates and the provisioning process can be found in [identity-claims].

#### 4.3.2. Expiration Timestamp

Generated by the UA during the creation of the Authentication message. It is set a short time into the future to protect against replay attacks of this DRIP format.

It shares the same format as the Trust Timestamp (Section 4.2.2).

#### 4.3.3. Attestation Signature

Performed by the UA using the onboard keypair which matches the HHIT in the Basic ID Message (0x0).

### 5. Transport Methods & Recommendations

#### 5.1. Legacy Advertisements (Bluetooth 4.X)

With Legacy Advertisements the goal is to attempt to bring reliable receipt of the paged Authentication Message. Forward Error Correction (Section 4.1.3) MUST be enabled when using Legacy Advertising methods (such as Bluetooth 4.X).

Under ASTM Bluetooth 4.X rules, transmission of dynamic messages are at least every 1 second while static messages (which is what

Authentication is classified under) are sent at least every 3 seconds.

Under DRIP the Certificate Message MUST be transmitted to properly meet the GEN 1 and GEN 3 requirement.

The ASTM Message Wrapper and Manifest both satisfy the GEN 2 requirement. At least one MUST be implemented to comply with the GEN 2 requirement.

A single Manifest can carry at most (using the full 10 page limit and 8 byte hashes) 12 unique hashes of previously sent messages (of any type). This results in a total of 22 (12 + 10) frames of Bluetooth data being transmitted over Bluetooth.

In comparison the Message Wrapper sends 6 pages (each a single frame) for each wrapped message. For backwards compatibility the implementation should also send the standard ASTM message that was wrapped for non-DRIP compliant receivers to obtain. This method results in 84 total Bluetooth frames (12 + (12 \* 6)) sent.

The question of which is better suited is up to the implementation.

## 5.2. Extended Advertisements (Bluetooth 5.X and Wifi NaN)

Under the ASTM specification, Bluetooth 5 or Wifi NaN transport of Remote ID is to use the Message Pack (Type 0xF) format for all transmissions. Under Message Pack all messages are sent together (in Message Type order) in a single Bluetooth frame (up to 9 single frame equivalent messages). 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 4.1.3) MUST NOT be used as it is impractical.

## 6. ASTM Considerations

- \* Increase Authentication Max Page Count from 5 to 10. Legacy Advertising can use all 10 while Extended Advertising has a maximum of 9 due to Bluetooth 5 limitations.
- \* Allocate Authentication Type 0x5 for DRIP from ASTM AuthType field.

## 7. IANA Considerations

This document does not require any actions by IANA.

## 8. Security Considerations

TODO

(Ed. Note: Hash lengths (length vs strength/collision rate); replay attacks with timestamps; static Cra (issue but nulled if UA signing other stuff dynamically meaning signatures will fail as HI won't match - this is probably a deeper discussion topic for provisioning security considerations when we get to there))

## 9. Acknowledgments

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

## 10. Appendix A: Thoughts on ASTM Authentication Message

The format standardized by the ASTM is designed with a few major considerations in mind, which the authors of this document feel put significant limitations on the expansion of the standard.

The primary consideration (in this context) is the use of the Bluetooth 5.X Extended Frame format. This method allows for a 255 byte payload to be sent in what the ASTM refers to as a "Message Pack".

The idea is to include up to five standard ASTM Broadcast RID messages (each of which are 25 bytes) plus a single authentication message (5 pages of 25 bytes each) in the Message Pack. The reasoning is then the Authentication Message is for the entire Message Pack.

The authors have no issues with this proposed approach; this is a valid format to use for the Authentication Message provided by the ASTM. However, by limiting the Authentication Message to ONLY five pages in the standard it ignores the possibility of other formatting options to be created and used.

Another issue with this format, not fully addressed in this document is fragmentation. Under Bluetooth 4.X, each page is sent separately which can result in lose of pages on the receiver. This is disastrous as the loss of even a single page means any signature is incomplete.

With the current limitation of 5 pages, Forward Error Correction (FEC) is nearly impossible without sacrificing the amount of data sent. More pages would allow FEC to be performed on the Authentication Message pages so loss of pages can be mitigated.

All these problems are further amplified by the speed at which UA fly and the Observer's position to receive transmissions. There is no guarantee that the Observer will receive all the pages of even a 5 page Authentication Message in the time it takes a UA to traverse across their line of sight. Worse still is that is not including other UA in the area, which congests the spectrum and could cause further confusion attempting to collate messages from various UA. This specific problem is out of scope for this document and our solutions in general, but should be noted as a design consideration.

## 11. References

### 11.1. Normative References

- [F3411-19] "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", DOI 10.6028/nist.sp.800-185, NIST Special Publication 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-requirements]  
Card, S., Wiethuechter, A., Moskowitz, R., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Requirements", Work in Progress, Internet-Draft, draft-ietf-drip-reqs-06, 1 November 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-drip-reqs-06.txt>>.

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

[identity-claims] Wiethuechter, A., Card, S., and R. Moskowitz, "DRIP Identity Claims", Work in Progress, Internet-Draft, draft-wiethuechter-drip-identity-claims-03, 2 November 2020, <<http://www.ietf.org/internet-drafts/draft-wiethuechter-drip-identity-claims-03.txt>>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <<https://www.rfc-editor.org/info/rfc7401>>.

#### Authors' Addresses

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

Email: [adam.wiethuechter@axenterprize.com](mailto:adam.wiethuechter@axenterprize.com)

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

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

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

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

drip Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 6 May 2021

A. Wiethuechter  
S. Card  
AX Enterprize, LLC  
R. Moskowitz  
HTT Consulting  
2 November 2020

DRIP Identity Claims  
draft-wiethuechter-drip-identity-claims-03

Abstract

This document describes the Identity Proofs (in the form of Claims, Certificates and Attestations) for use in various Drone Remote ID Protocols (DRIP) and the wider Unmanned Traffic Management (UTM) system.

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 6 May 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Claims, Assertions, Attestations, and Certificates . . . .	2
1.1.1. Claims . . . . .	3
1.1.2. Assertions . . . . .	3
1.1.3. Attestations . . . . .	3
1.1.4. Certificates . . . . .	3
2. Terminology . . . . .	3
2.1. Required Terminology . . . . .	3
2.2. Definitions . . . . .	4
3. DRIP Proofs . . . . .	4
3.1. Certificate: X on X (Cxx Form) . . . . .	4
3.1.1. Certificate: X on X (Short Form) . . . . .	6
3.2. Attestation: X on Y (Axy Form) . . . . .	6
3.2.1. Attestation: X on Y (Short Form) . . . . .	8
3.2.2. Attestation: X on Y (Offline Form) . . . . .	9
3.3. Timestamps . . . . .	11
3.4. Signatures . . . . .	11
4. Provisioning . . . . .	11
4.1. HHIT Delegation . . . . .	11
4.2. Manufacturer . . . . .	12
4.3. Registry . . . . .	12
4.4. Operator . . . . .	13
4.5. Aircraft . . . . .	14
4.5.1. Standard Provisioning . . . . .	14
4.5.2. Operator Assisted Provisioning . . . . .	16
4.5.3. Initial Provisioning . . . . .	18
5. Security Considerations . . . . .	18
6. References . . . . .	18
6.1. Normative References . . . . .	18
6.2. Informative References . . . . .	18
Authors' Addresses . . . . .	19

## 1. Introduction

DRIP Proofs are the backbone of trust in DRIP UAS RID, consisting of a chain of special certificates/attestations that results in a something useful in Broadcast RID. Some of the certificates are stored in and are generated by the Registries (defined in [hhit-registries]) and allow a user to confirm the trustworthiness of an Unmanned Aircraft (herein referred to as Aircraft) even in the scenario that the Observer is disconnected from the Internet.

## 1.1. Claims, Assertions, Attestations, and Certificates

The authors wish to make a clear distinction on exactly what these terms mean in the context of DRIP.



This is due to the term "certificate" having significant technologic and legal baggage associated with it, specifically around X.509 certificates. These type of certificates and Public Key Infrastructure invokes more legal and public policy considerations than probably any other electronic communication sector. It emerged as a governmental platform for trusted identity management and was pursued in intergovernmental bodies with links into treaty instruments.

As such much discussion has been made around the terms being used.

#### 1.1.1. Claims

For DRIP claims are used in the form of a predicate (X is Y, X has property Y, and most importantly X owns Y). The basic form of a claim is an entity using a HHIT as an identifier in the DRIP UAS system.

#### 1.1.2. Assertions

Assertions, under DRIP, are defined as being a set of one or more claims. This definition is borrowed from JWT/CWT. An HHIT in of itself is a set of assertions. First that the identifier is unique and is a handle to an asymmetric keypair owned by the entity and that it also is part of the given registry (specified by the HID).

#### 1.1.3. Attestations

An attestation is a signed claim. The signee may be the claimant themselves or a third party. Under DRIP this is normally used when a set of entities asserts a relationship between them along with other information.

#### 1.1.4. Certificates

Certificates in DRIP have a narrow definition of being signed exclusively by a third party and are only over identities.

### 2. Terminology

#### 2.1. Required Terminology

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

## 2.2. Definitions

See [drip-requirements] for common DRIP terms.

HDA: Hierarchial HIT Domain Authority. The 16 bit field identifying the HIT Domain Authority under a RAA.

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

RAA: Registered Assigning Authority. The 16 bit field identifying the Hierarchical HIT Assigning Authority.

## 3. DRIP Proofs

The DRIP Proofs is a set of custom structures to be used in the USS/UTM system. They are created during the provision of an Aircraft and are tied to the UAS ID (expected to be a HHIT, see [drip-rid] for details).

These structures when chained together can create a root of trust all the way back to the manufacturer itself during the initial production of a given Aircraft. The chain can also be used by authorized entities to trace an Aircraft through all owners and flights in the Aircraft's lifetime (something of interest to ICAO).

The rest of this section will define the formats of proofs in DRIP as forms of certificates and attestations and their common uses.

### 3.1. Certificate: X on X (Cxx Form)

The Cxx Form of DRIP Proofs is a self-signed certificate (by an entity known as 'X') staking an unverified claim on a HHIT/HI pairing until an expiration date/time.

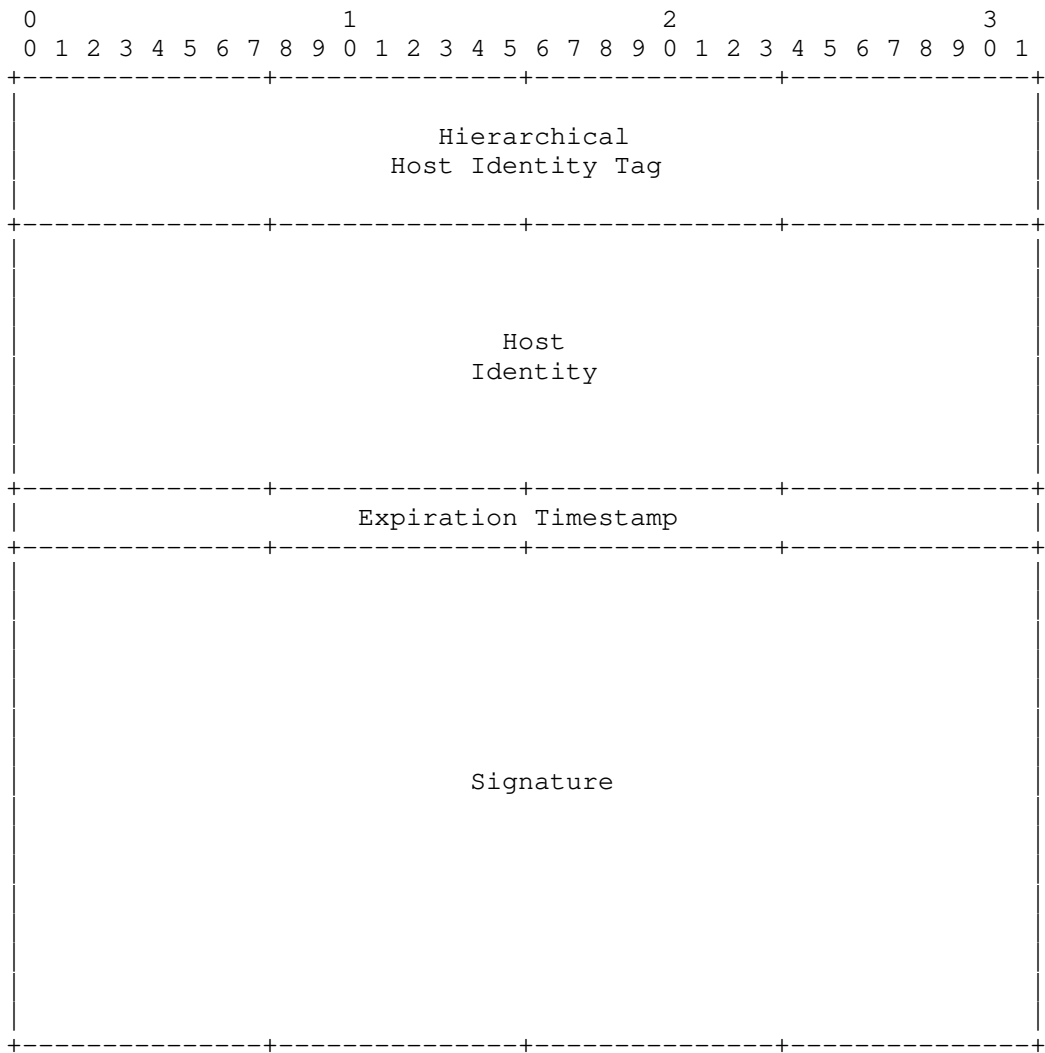


Figure 1: Certificate: X on X

Certificates of the Cxx Form are 116 bytes. The offset of the Expiration Timestamp SHOULD be of significant length (possibly years).

These are 5 (five) Cxx Certificates that can be created in a standard DRIP UAS RID system: Manufacturer on Manufacturer, RAA on RAA, HDA on HDA (Registry on Registry), Operator on Operator, and Aircraft on Aircraft. This is not an exhaustive list as any entity with the DRIP UAS system SHOULD have a Cxx for itself.

3.1.1. Certificate: X on X (Short Form)

A smaller version of Certificate: X on X exists where the Host Identity is removed allowing a claim to be made in 84 bytes.

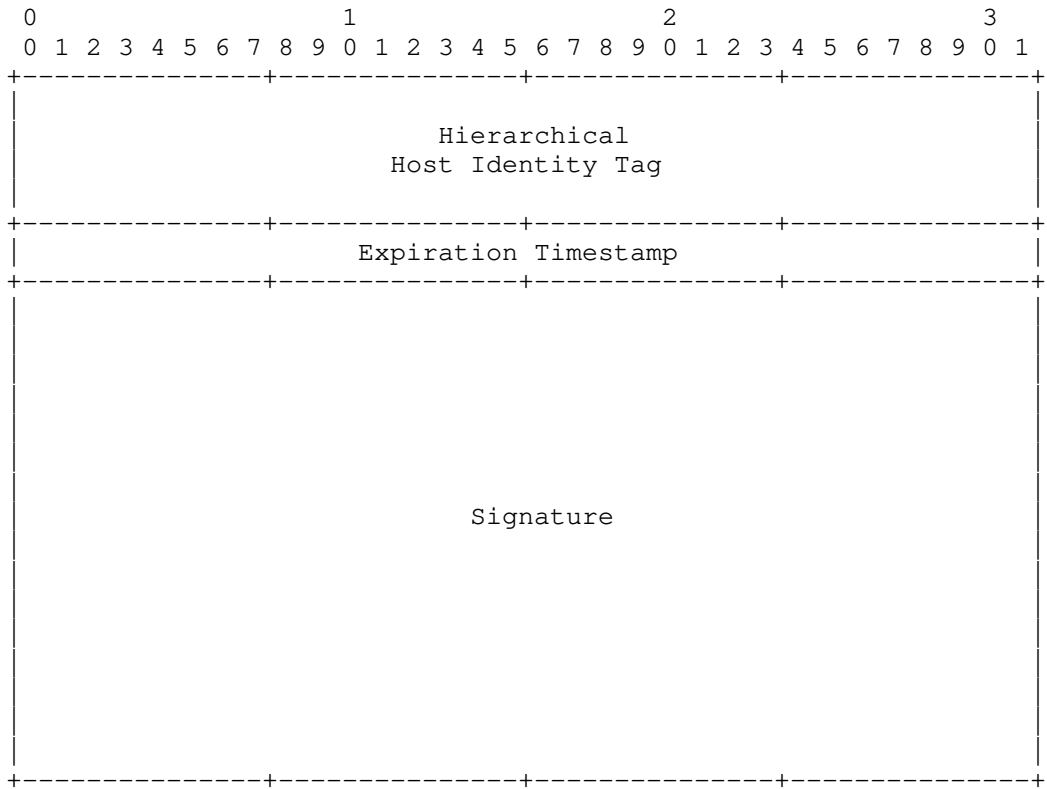


Figure 2: Certificate: X on X (Short Form)

3.2. Attestation: X on Y (Axy Form)

This DRIP Proof is an attestation where Entity X asserts trust in the binding claimed by Entity Y (in Cyy) and signs this asserting with a timestamp and an expiration of when the binding is no longer asserted by Entity X.

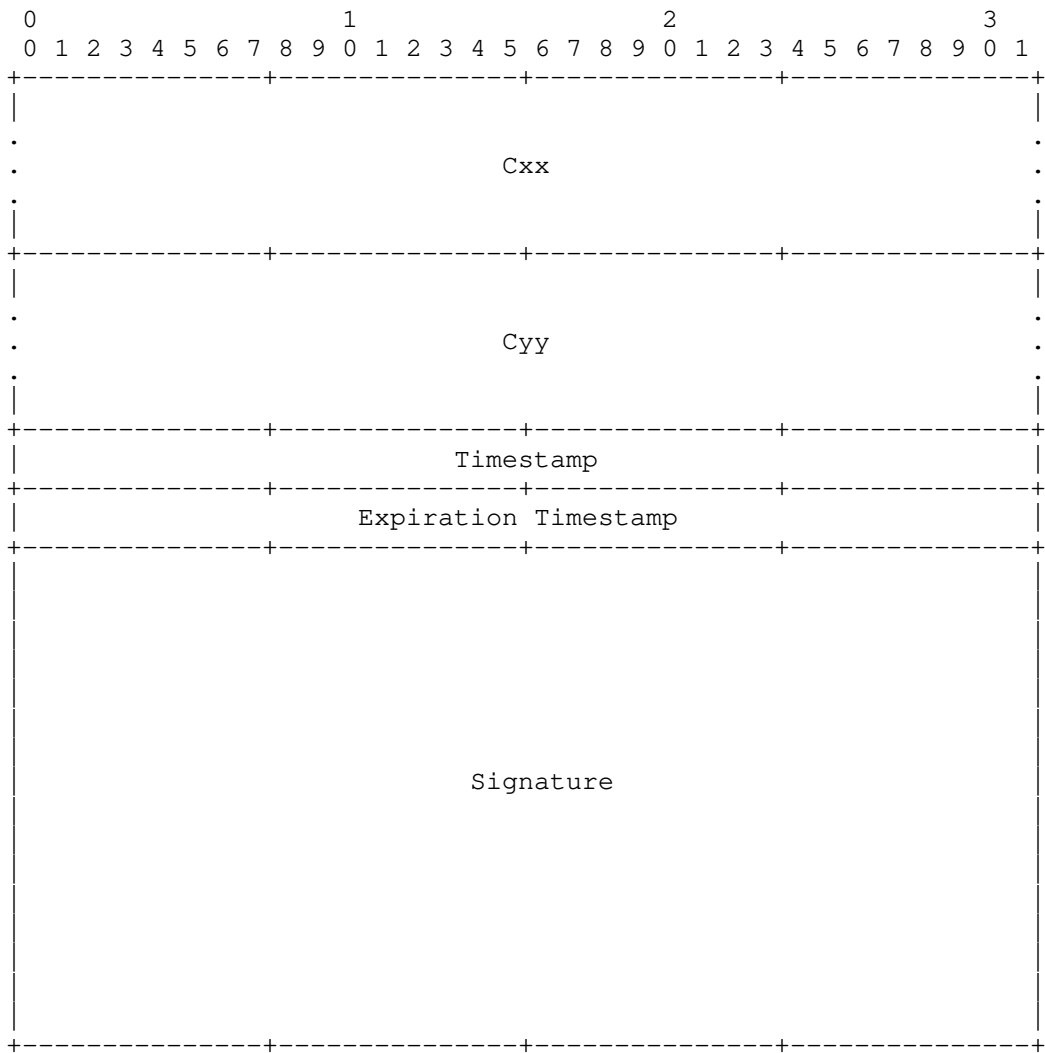


Figure 3: Attestation: X on Y

Axy Form wraps both self-signed certificates of the entities and is signed by Entity X. Two timestamps, one taken at the time of signing and one as an expiration time are used to set boundaries to the assertion. Care should be given to how far into the future the Expiration Timestamp is set, but is left up to system policy.

Most attestations of this form have a length of 304 bytes. Attestation: Registry on Operator on Aircraft is unique in that is 680 bytes long, binding of two Axy forms (in this specific case

Attestation: Registry on Operator with Attestation: Operator on Aircraft).

3.2.1. Attestation: X on Y (Short Form)

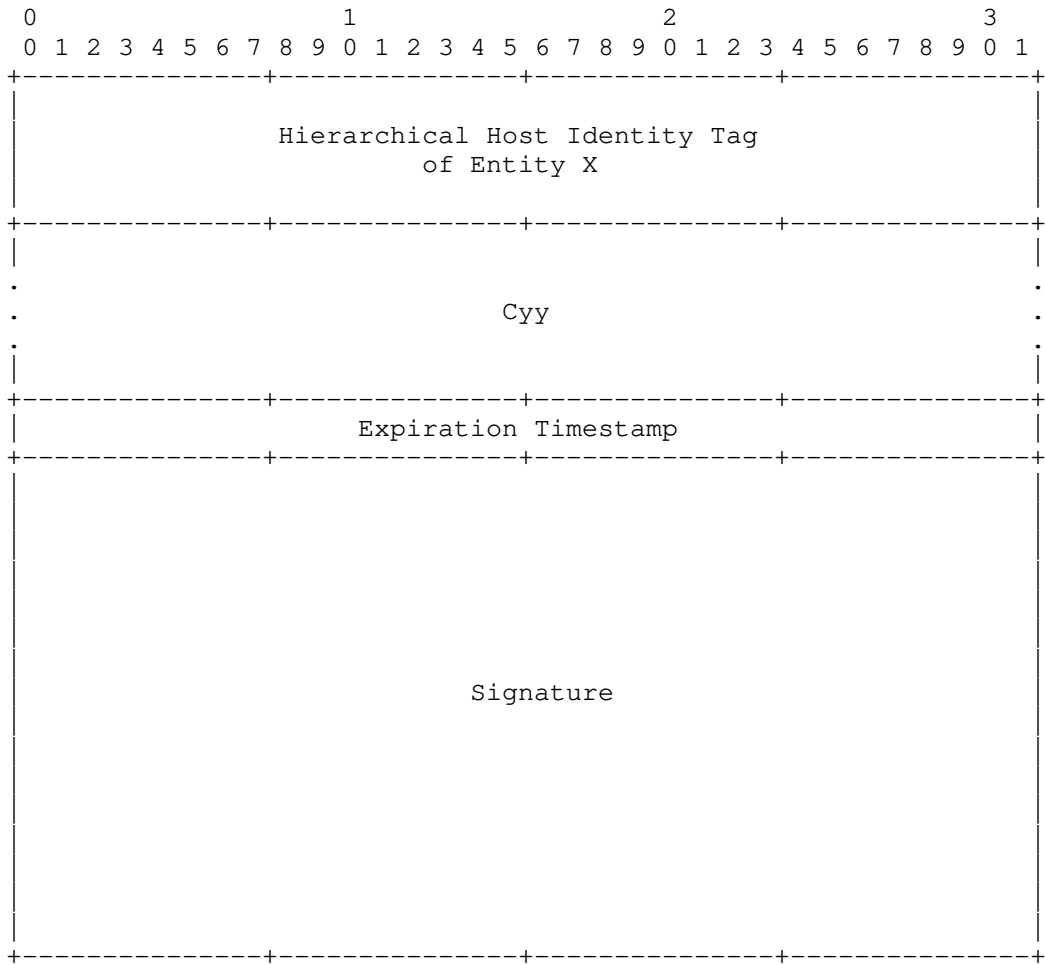


Figure 4: Attestation: X on Y (Short Form)

The short form of the Axy this attestation is 200 bytes long and is designed to fit inside the framing of the ASTM F3411 Authentication Message. The HHIT of Entity X is used in place of the full Cxx (see Section 5 for comments). The timestamp is removed and only an expiration timestamp is present.

During creation the Expiration Timestamp MUST be no later than the Expiration Timestamp found in Cyy.

### 3.2.2. Attestation: X on Y (Offline Form)

A special attestation that is the basis for a certificate finalized onboard the aircraft during flight. It is used in Broadcast RID to provide the trustworthiness of the Aircraft without the need of the Observer to be connected to the Internet.

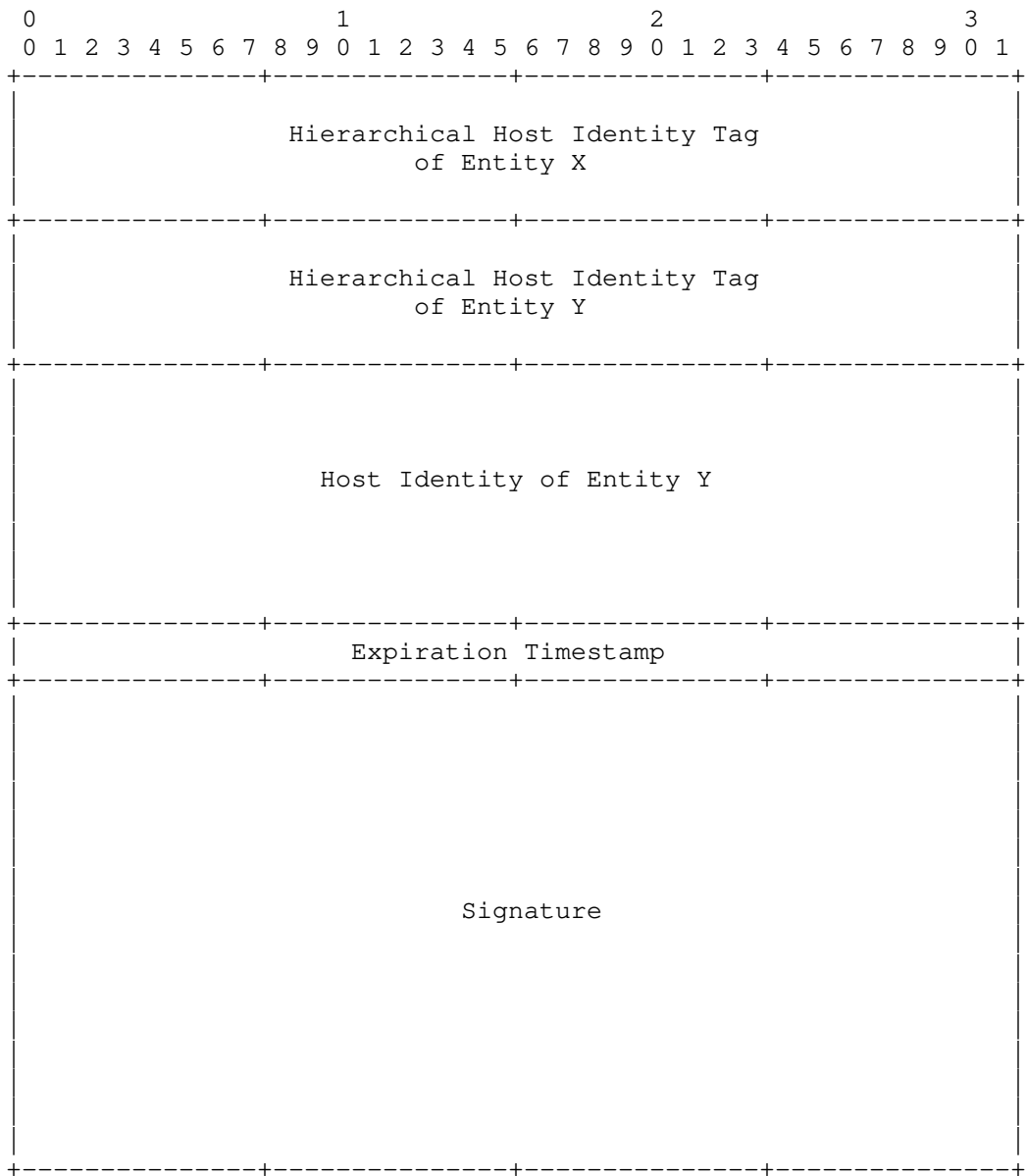


Figure 5: Attestation: X on Y (Offline Form)

The signature is generated using Entity X's keypair.



### 3.3. Timestamps

Timestamps MAY be the standard UNIX time or a protocol specific timestamp, to avoid programming complexities. For example [F3411-19] uses a 00:00:00 01/01/2019 offset. When a Expiration Timestamp is required a desired offset is added, setting the timestamp into the future. The amount of offset for specific timestamps is left to best practice.

### 3.4. Signatures

Signatures are ALWAYS taken over the preceding fields in the certificate/attestation. For DRIP the EdDSA25519 algorithm from [RFC8032] is used.

## 4. Provisioning

Under DRIP UAS RID a special provisioning procedure is required to properly generate and distribute the certificates and attestations to all parties in the USS/UTM ecosystem using DRIP RID.

Keypairs are expected to be generated on the device hardware it will be used on. Due to hardware limitations (see Section 5) and connectivity it is acceptable under DRIP RID to generate keypairs for the Aircraft on Operator devices and later securely inject them into the Aircraft (as defined in Section 4.5.2). The methods to securely inject and store keypair information in a "secure element" of the Aircraft is out of scope of this document.

### 4.1. HHIT Delegation

Under the FAA [NPRM], it is expecting that IDs for UAS are assigned by the UTM and are generally one-time use. The methods for this however are unspecified leaving two options.

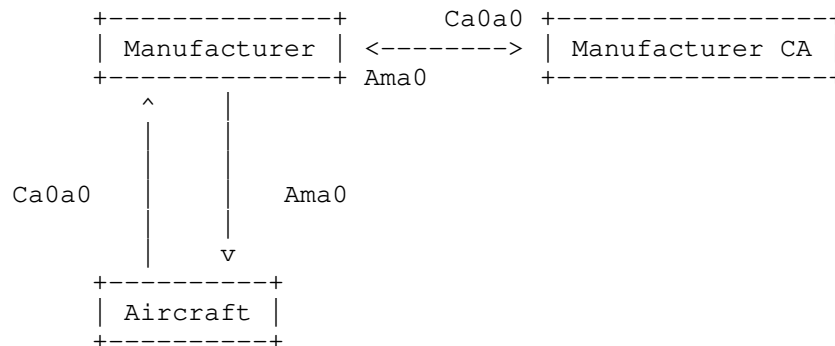
- 1 The entity generates its own HHIT, discovering and using thr RAA and HDA for the target Registry. The method for discovering a Registry's RAA and HDA is out of scope here. This allows for the device to generate an HHIT to send to the Registry to be accepted (thus generating the required Host Identity Claim) or denied.
- 2 The entity sends to the Registry its HI for it to be hashed and result in the HHIT. The Registry would then either accept (returning the HHIT to the device) or deny this pairing.

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

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

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

#### 4.2. Manufacturer



During the initial configuration and production at the factory the Aircraft MUST be configured to have a serial number. ASTM defines this to be an ANSI/CTA-2063A. Under DRIP a HHIT can be encoded as such to be able to convert back and forth between them. This is out of scope for this document.

Under DRIP the Manufacturer SHOULD be using HHITs and have their own keypair and Cxx (Certificate: Manufacturer on Manufacturer). (Ed. Note: some words on aircraft keypair and certs here?).

Certificate: Aircraft 0 on Aircraft 0 (Ca0a0) is extracted by the manufacturer and send to their Certificate Authority (CA) to be verified and added. A resulting certificate (Attestation: Manufacturer on Aircraft 0) SHOULD be a DRIP Attestation in the Axy Form - however this could be a X.509 certificate binding the serial number to the manufacturer.

#### 4.3. Registry

TODO

DRIP UAS RID defines two levels of hierarchy maintained by the Registration Assigning Authority (RAA) and HHIT Domain Authority (HDA). The authors anticipate that an RAA is owned and operated by a

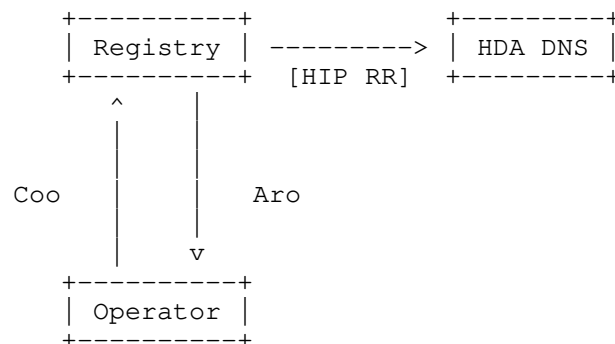
regional CAA (or a delegated party by an CAA in a specific airspace region) with HDAs being contracted out. As such a chain of trust for registries is required to ensure trustworthiness is not compromised. More information on the registries can be found in [hhit-registries].

Both the RAA and HDA generate their own keypairs and self-signed certificates (Certificate: RAA on RAA and Certificate: HDA on HDA respectively). The HDA sends to the RAA its self-signed certificate to be added into the RAA DNS.

The RAA confirms the certificate received is valid and that no HHIT collisions occur before added a HIP RR to its DNS for the new HDA. An Attestation: RAA on HDA is sent as a confirmation that provisioning was successful.

The HDA is now a valid "Registry" and uses its keypair and Certificate: HDA on HDA with all provisioning requests from downstream.

#### 4.4. Operator



The Operator generates a keypair and HHIT as specified in DRIP UAS RID. A self-signed certificate (Certificate: Operator on Operator) is generated and sent to the desired Registry (HDA). Other relevant information and possibly personally identifiable information needed may also be required to be sent to the Registry (all over a secure channel - the method of which is out of scope for this document).

The Registry cross checks any personally identifiable information as required. Certificate: Operator on Operator is verified (both using the expiration timestamp and signature). The HHIT is searched in the Registries database to confirm that no collision occurs. A new attestation is generated (Attestation: Registry on Operator) and sent securely back to the Operator. Optionally the HHIT/HI pairing can be added to the Registries DNS in to form of a HIP Resource Record (RR).

Other RRs, such as CERT and TXT, may also be used to hold public information.

With the receipt of Attestation: Registry on Operator the provisioning of an Operator is complete.

#### 4.5. Aircraft

##### 4.5.1. Standard Provisioning

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

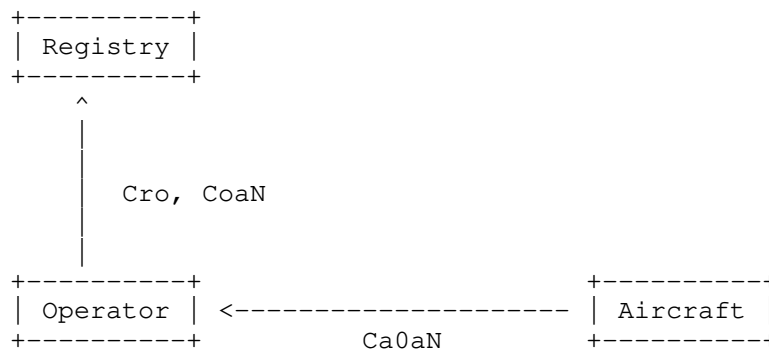


Figure 6: Standard Provision: Step 1

Through mechanisms not specified in this document the Aircraft should have methods to instruct the Aircrafts onboard systems to generate a keypair and certificate. This certificate is chained to the factory provisioned certificate (Certificate: Aircraft 0 on Aircraft 0). This new attestation (Attestation: Aircraft 0 on Aircraft N) is securely extracted by the Operator.

With Attestation: Aircraft 0 on Aircraft N the sub certificate (Certificate: Aircraft N on Aircraft N) is used by the Operator to generate Attestation: Operator on Aircraft N. This along with Attestation: Registry on Operator is sent to the Registry.



Figure 7: Standard Provision: Step 2

On the Registry, Attestation: Registry on Operator is verified and used as confirmation that the Operator is already registered. Attestation: Operator on Aircraft N also undergoes a validation check and used to generate a token to return to the Operator to continue provisioning.

Upon receipt of this token, the Operator injects it into the Aircraft and its used to form a secure connection to the Registry. The Aircraft then sends Attestation: Manufacturer on Aircraft 0 and Attestation: Aircraft 0 to Aircraft N.

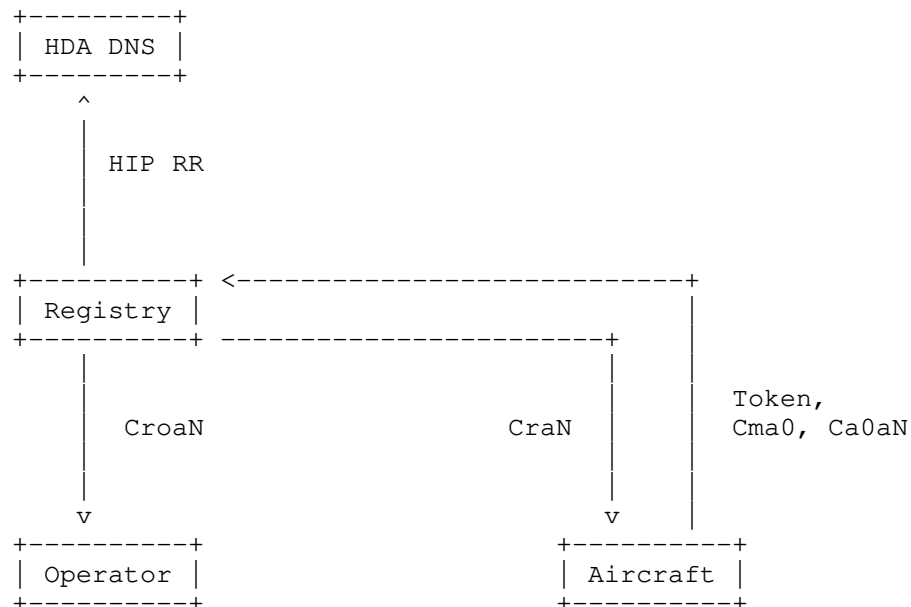


Figure 8: Standard Provision: Step 3

The Registry uses Attestation: Manufacturer on Aircraft 0 (with an external database if supported) to confirm the validity of the Aircraft. Attestation: Aircraft 0 on Aircraft N is correlated with Attestation: Operator on Aircraft N and Attestation: Manufacturer on Aircraft 0 to see the chain of ownership. The new HHIT tied to Aircraft N is then checked for collisions in the HDA. With the information the Registry generates two certificates: Attestation: Registry on Operator on Aircraft N and Attestation: Registry on Aircraft N (Offline Form). A HIP RR (and other RR types as needed) are generated and inserted into the HDA.

Attestation: Registry on Operator on Aircraft N is sent via a secure channel back to the Operator to be stored. Attestation: Registry on Aircraft N (Offline Form) is sent to the Aircraft to be used in Broadcast RID.

#### 4.5.2. Operator Assisted Provisioning

This provisioning scheme is for when the Aircraft is unable to connect to the Registry itself or does not have the hardware required to generate keypairs and certificates.

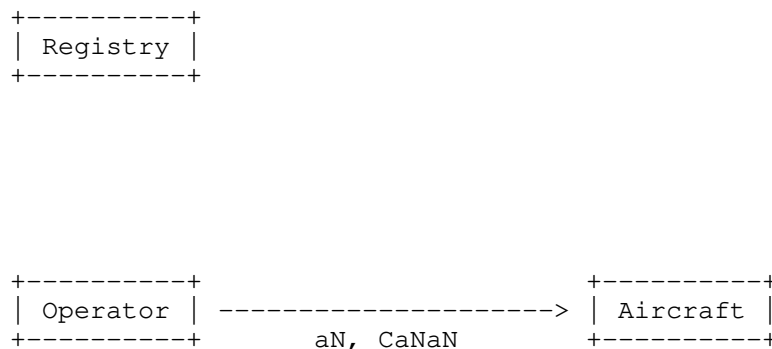


Figure 9: Operator Assisted Provision: Step 1

To start the Operator generates on behalf of the Aircraft a new keypair and Certificate: Aircraft N on Aircraft N. This keypair and certificate are injected into the Aircraft for it to generate Attestation: Aircraft 0 on Aircraft N. After injecting the keypair and certificate, the Operator MUST destroy all copies of the keypair.

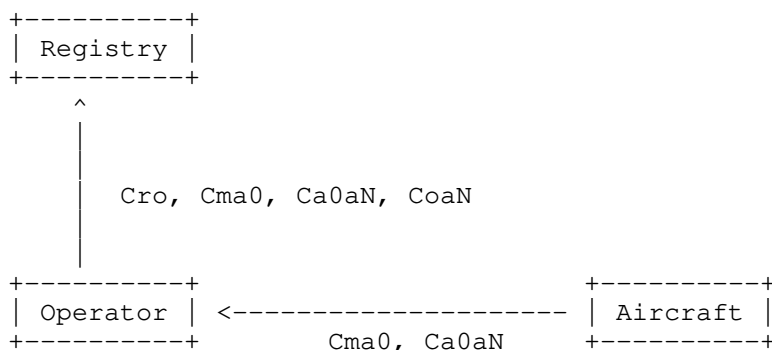


Figure 10: Operator Assisted Provision: Step 2

Attestation: Manufacturer on Aircraft 0 and Attestation: Aircraft 0 on Aircraft N is extracted by the Operator and the following data items are sent to the Registry; Attestation: Registry on Operator, Attestation: Manufacturer on Aircraft 0, Attestation: Aircraft 0 on Aircraft N, Attestation: Operator on Aircraft N.

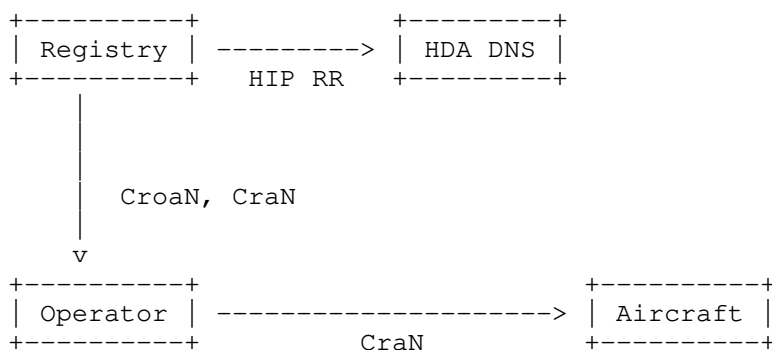


Figure 11: Operator Assisted Provision: Step 3

On the Registry validation checks are done on all attestations as per the previous sections. Once complete then the Registry checks for a HHIT collision, adding to the HDA if clear and generates Attestation: Registry on Operator on Aircraft N and Attestation: Registry on Aircraft N (Offline Form). Both are sent back to the Operator.

The Operator securely inject Attestation: Registry on Aircraft N (Offline Form) and securely stores Attestation: Registry on Operator on Aircraft N.

#### 4.5.3. Initial Provisioning

A special form of provisioning is used when the Aircraft is first sold to an Operator. Instead of generating a new keypair, the built in keypair and certificate done by the Manufacturer is used to provision and register the aircraft to the owner.

For this either Standard or Operator Assisted methods can be used.

### 5. Security Considerations

A major consideration is the optimization done in Attestation: X on Y (Short Form) to get its length down to 200 bytes. The truncation of Certificate: HDA on HDA down to just its HHIT is one that could be used against the system to act as a false Registry. For this to occur an attacker would need to find a hash collision on that Registry HHIT and then manage to spoof all of DNS being used in the system.

The authors believe that the probability of such an attack is low when Registry operators are using best practices in security. If such an attack can occur (especially in the time frame of "one-time use IDs") then there are more serious issues present in the system.

### 6. References

#### 6.1. Normative References

- [F3411-19] "Standard Specification for Remote ID and Tracking", February 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <<https://www.rfc-editor.org/info/rfc8032>>.
- [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>>.

#### 6.2. Informative References



## [drip-requirements]

Card, S., Wiethuechter, A., Moskowitz, R., and A. Gurtov,  
"Drone Remote Identification Protocol (DRIP)  
Requirements", Work in Progress, Internet-Draft, draft-  
ietf-drip-reqs-06, 1 November 2020, <[http://www.ietf.org/  
internet-drafts/draft-ietf-drip-reqs-06.txt](http://www.ietf.org/internet-drafts/draft-ietf-drip-reqs-06.txt)>.

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

## [hhit-registries]

Moskowitz, R., Card, S., and A. Wiethuechter,  
"Hierarchical HIT Registries", Work in Progress, Internet-  
Draft, draft-moskowitz-hip-hhit-registries-02, 9 March  
2020, <[http://www.ietf.org/internet-drafts/draft-  
moskowitz-hip-hhit-registries-02.txt](http://www.ietf.org/internet-drafts/draft-moskowitz-hip-hhit-registries-02.txt)>.

[NPRM] "Notice of Proposed Rule Making on Remote Identification  
of Unmanned Aircraft Systems", December 2019.

## Authors' Addresses

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

Email: [adam.wiethuechter@axenterprize.com](mailto:adam.wiethuechter@axenterprize.com)

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

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

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

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