

Workgroup: DRIP
Internet-Draft:
draft-wiethuechter-drip-auth-00
Published: 23 March 2020
Intended Status: Standards Track
Expires: 24 September 2020
Authors: A. Wiethuechter S. Card R. Moskowitz
 AX Enterprize AX Enterprize HTT Consulting
DRIP Authentication Formats

Abstract

This document describes how to include trust into the ASTM Remote ID specification defined in ASTM3411 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 24 September 2020.

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. Terms and Definitions](#)
 - [2.1. Requirements Terminology](#)
 - [2.2. Definitions](#)
- [3. Background](#)
 - [3.1. Problem Space And Document Focus](#)
 - [3.2. ASTM Authentication Message](#)
 - [3.3. Thoughts on ASTM Authentication Message](#)
- [4. HIP Based Extensions to the ASTM Authentication Message](#)
 - [4.1. Message Wrapper](#)
 - [4.1.1. Inner Header](#)
 - [4.1.2. Trust Timestamp](#)
 - [4.1.3. Payload](#)
 - [4.1.4. Forward Error Correction](#)
 - [4.2. Signed Hash Lists](#)
 - [4.2.1. Hash Operation](#)
 - [4.2.2. Pseudo-blockchain Hashes](#)
 - [4.2.3. Limitations](#)
 - [4.3. Offline Claim](#)
 - [4.3.1. Claim: Registry on Aircraft](#)
 - [4.3.2. Forward Error Correction](#)
- [5. Example Use Cases](#)
 - [5.1. Trusted Messages](#)

[5.2. Wrapped Signed Hashes](#)

[6. IANA Considerations](#)

[7. Security Considerations](#)

[8. Acknowledgments](#)

[9. References](#)

[9.1. Normative References](#)

[9.2. Informative References](#)

[Authors' Addresses](#)

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 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 authentication message format, using the Host Identity Protocol Version 2 (HIPv2) [[RFC7401](#)] Hierarchical HIT (HHIT) [[I-D.moskowitz-hip-hierarchical-hit](#)].

2. Terms and Definitions

2.1. Requirements Terminology

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

2.2. Definitions

HI

Host Identity. The public key portion of an assymmetric keypair from HIP. In this document it is assumed that the HI is based on a EdDSA25519 keypair. This is supported by new crypto defined in [[I-D.moskowitz-hip-new-crypto](#)].

HIT

Host Identity Tag. A 128 bit handle on the HI. Defined in HIPv2 [[RFC7401](#)].

HHIT

Hierarchical Host Identity Tag. A 128 bit handle on the HI contain extra information not found in a standard HIT. Defined in [[I-D.moskowitz-hip-hierarchical-hit](#)].

UA

Unmanned Aircraft. In this document UA's are typically though of as drones of commerical or military variety. This is a very strict definition which can be relaxed to include any and all aircraft that are unmanned.

UAS

Unmanned Aircraft System. Composed of Unmanned Aircraft and all required on-board subsystems, payload, control station, other required off-board subsystems, any required launch and recovery equipment, all required crew members, and C2 links between UA and the control station.

RID

Remote ID. A unique identifier found on all UA to be used in communication and in regulation of UA operation.

Observer

Referred to in other UAS documents as a "user", but there are also other classes of RID users, so we prefer "observer" to denote an individual who has observed an UA and wishes to know something about it, starting with its RID.

3. Background

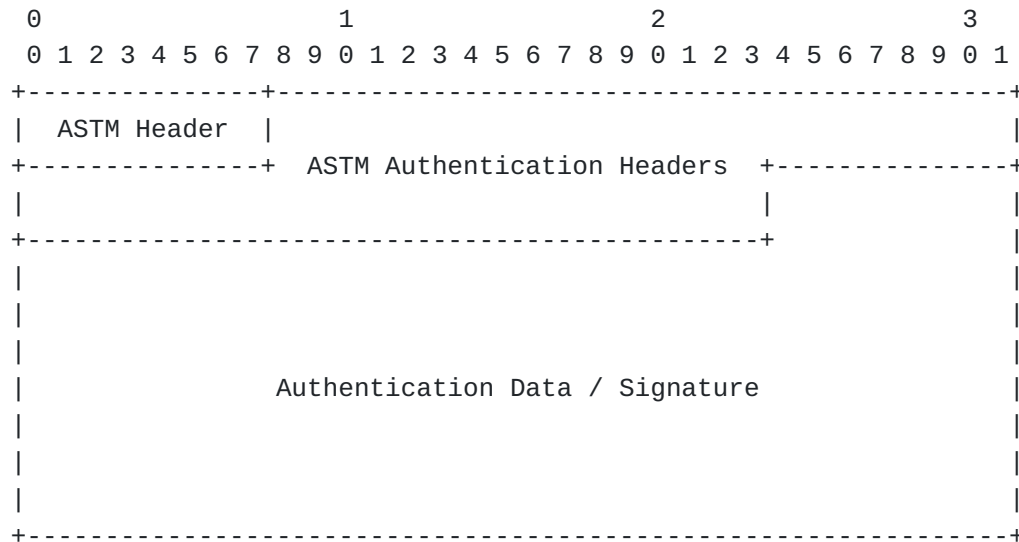
3.1. Problem Space And Document Focus

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.

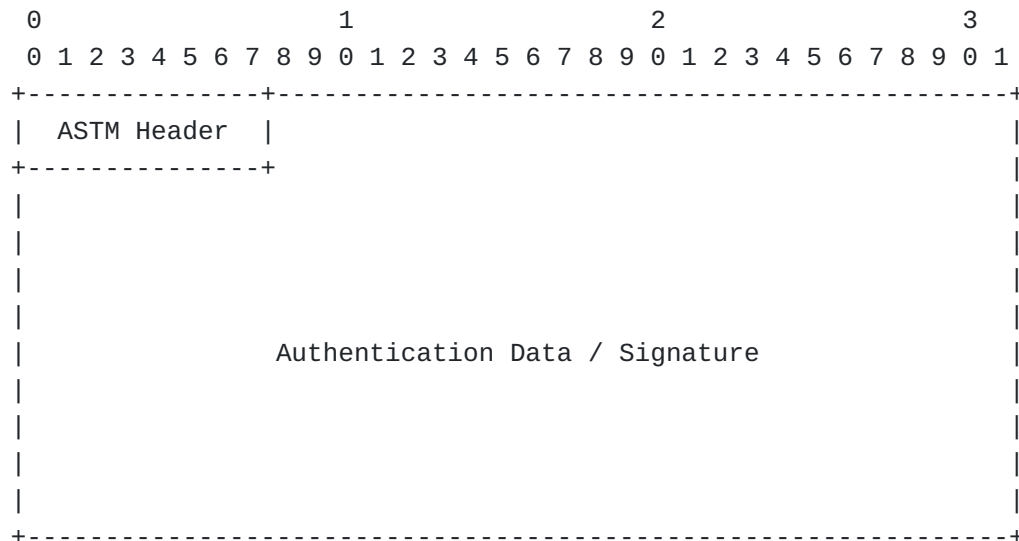
The following subsections will provide a high level reference to the ASTM standard for authentication messages and how their current limitations effect trust in the Broadcast RID enviornment.

3.2. ASTM Authentication Message

Page 0:



Page 1 - 4:



ASTM Header (1 byte):

Contains basic ASTM information such as message type and protocol version.

ASTM Authentication Headers: (6 bytes)

Contains header information for the authentication message from ASTM UAS RID Standard.

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

Opaque authentication data.

The above diagram is the format defined by ASTM 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.

3.3. Thoughts on ASTM Authentication Message

The format standardized by the ASTM is designed with a few major considerations in mind, which the authors 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 loss 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.

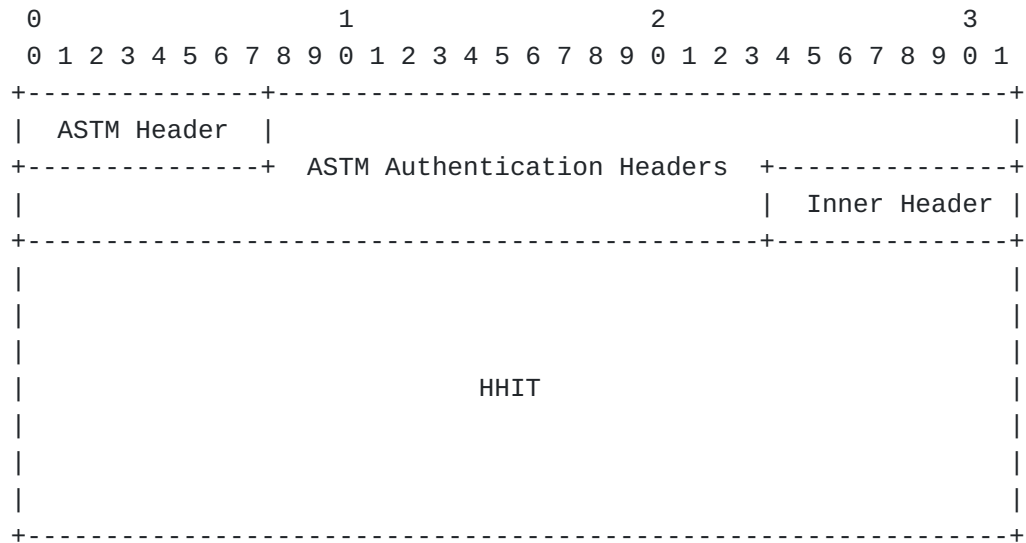
4. HIP Based Extensions to the ASTM Authentication Message

The following section describes various methods that HIP can help enable more trustworthy communication using the Authentication Message as the base. Each diagram will show all pages of the format filled out.

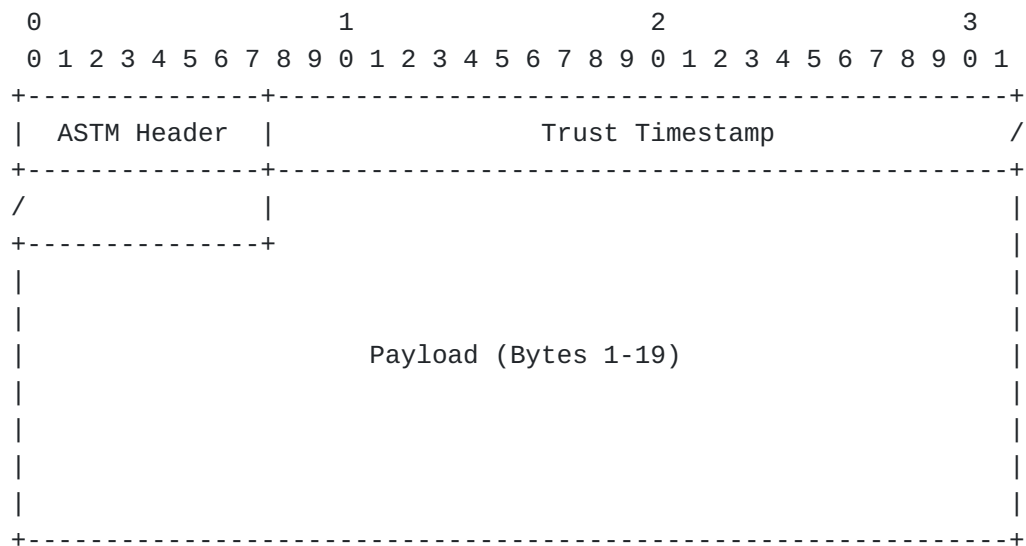
In the future it is probably necessary to have a AuthType assigned by ASTM for DRIP related authentication message formats. How this is accomplished and how it extends if more formats are proposed is still TBD.

4.1. Message Wrapper

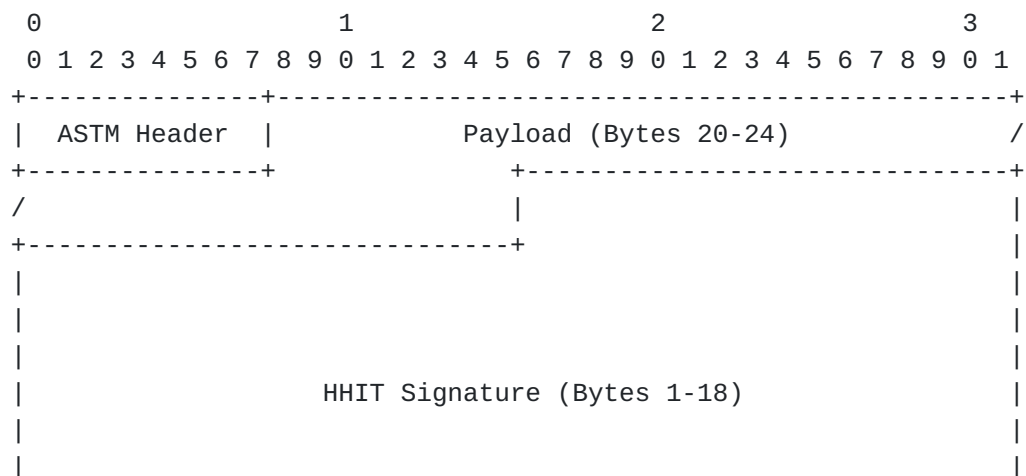
Page 0:

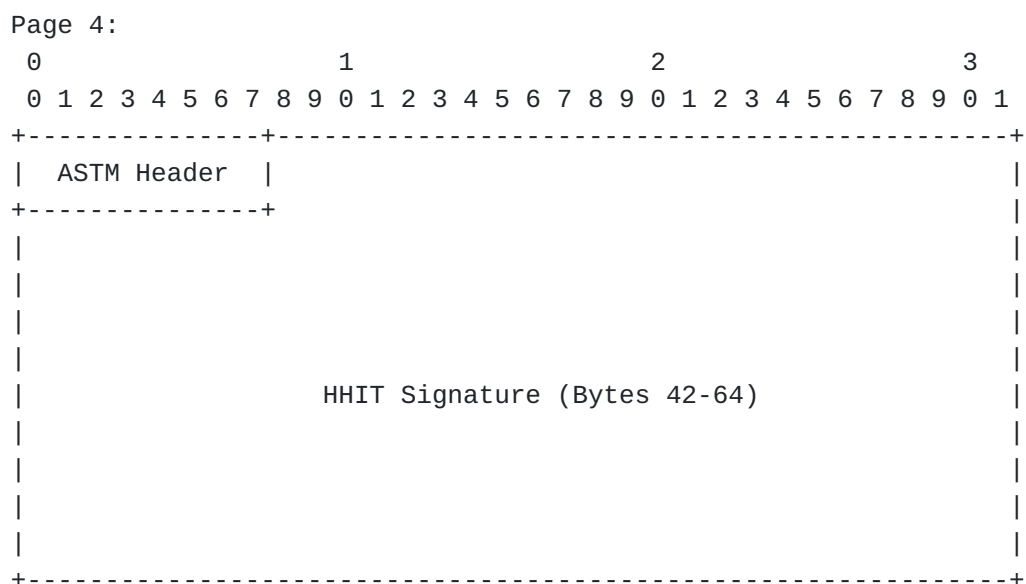
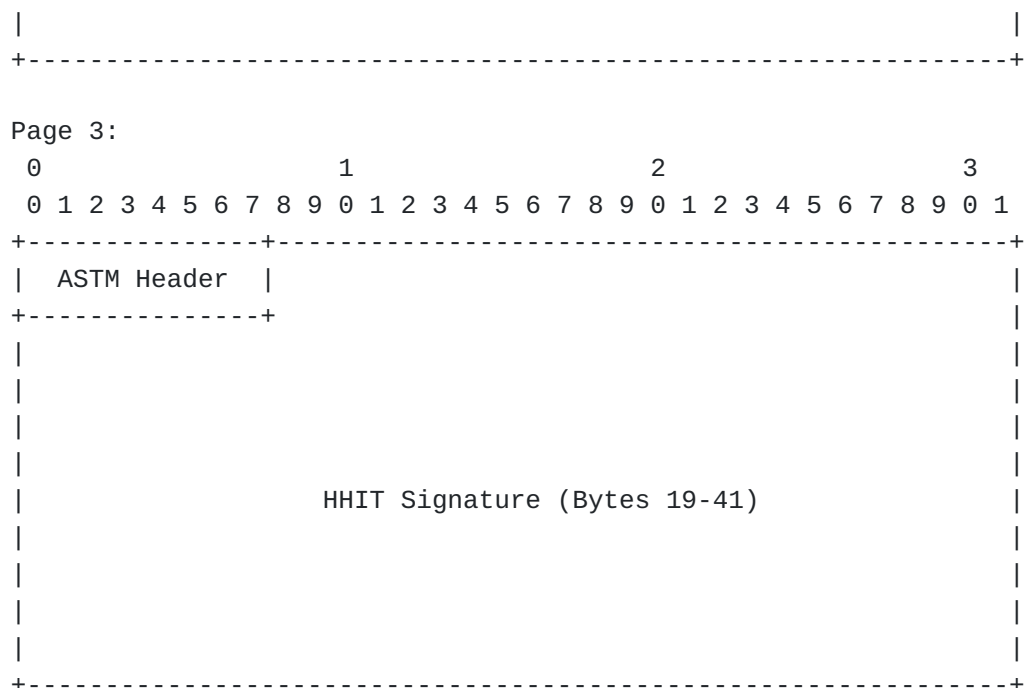


Page 1:



Page 2:





Inner Header (1 byte):
 See Inner Header section.

HHIT: (16 bytes)
 HHIT using the EdDSA25519 HI.

Trust Timestamp: (4 bytes)
 Timestamp denoting current time plus an offset to trust message to.

Payload: (24 bytes)
 Opaque payload data.

HHIT Signature: (64 bytes)

Signature over preceding fields using the EdDSA25519 keypair.

When this authentication format is recieved the HHIT is first looked up in DNS by standard mechanisms to retrieve the HIP RR. From this the HI can be used to perform signature verification. The data signed is all the data preceding the 64 byte signature (excluding the ASTM Headers and ASTM Authentication Headers).

When there is no Internet connectivity on the Observers device the HI of the UA can be obtained by referencing the claim sent in the Offline Based Authentication format, if sent by the UA.

4.1.1. Inner Header

When the payload is another ASTM message then the first byte of the full message should be used to fill in the Inner Header field. This byte can be signed but is primarily for identitying the inner message type (for unpacking purposes if desired by the Observer device).

Another use of the Inner Header byte is holding the H-Alg and H-Len fields if this format is used for Hashed Messages. See Section 5.2 for a detailed example of this. When this format is used all 24 bytes of the payload MUST be filled with hash values. This requirement means that only even numbered hash lengths can be used.

Any other wrapped messages that are not 24 bytes long and require padding on the payload MUST use the Inner Header to inform the reciever the length of payload in octets.

To accomadate payload differences the use of AuthType in the ASTM Authentication Headers section SHOULD be set in the range of 0xA-F (which is availible for Private Use). Only recommended values are defined here, as the allocation in the reserved space of this header is controlled by ASTM.

AuthType	Values
-----	-----
Wrapped ASTM Message	A
Wrapped Hashes	B
Wrapped Message Length (octets)	C

4.1.2. Trust Timestamp

Trust Timestamp MUST be current UNIX time plus an offset into the future.

To avoid replay attacks the Trust Timestamp field must be well founded. When wrapping a vector (position) message the payload WILL contain (by ASTM rules) constantly changing data, this includes its

own timestamp. In this case the Trust Timestamp could be argued as superfluous.

Other 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.

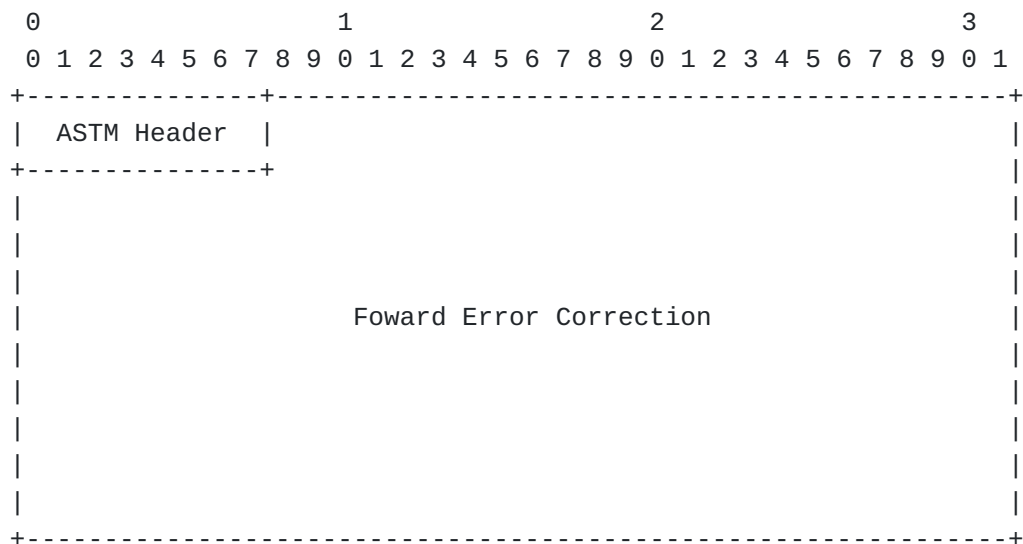
4.1.3. Payload

The payload can be anything that fits within the 24 byte limit. An example of what could be done with this format is found in Section 5.1. If the payload is less than 24 bytes, null padding MUST be used to fill up to 24 bytes and the AuthType of 0xC MUST be used to identify message size.

4.1.4. Forward Error Correction

Due to the nature of the wireless link a FEC method is suggested. Under DRIP the Message Wrapper format has two forms of FEC that can be used.

Page 5:



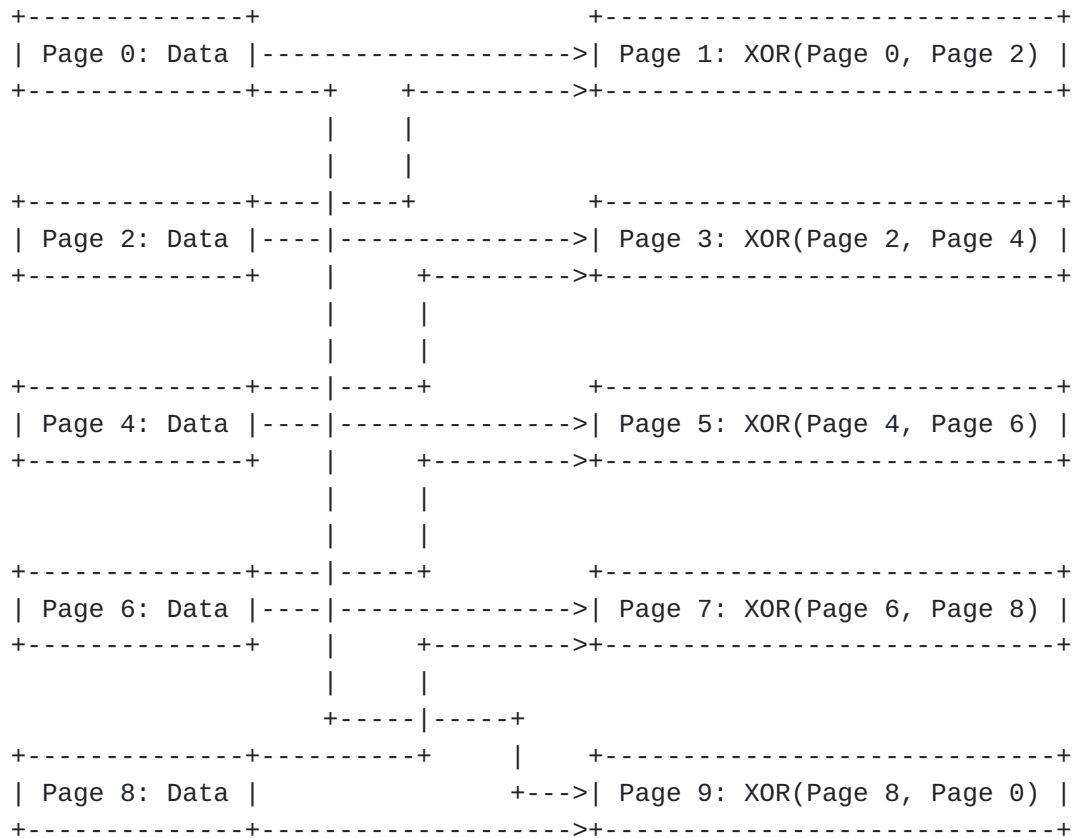
4.1.4.1. Reed Solomon

The preferred form of FEC is using Reed Solomon. With this the entire authentication message (all pages, including headers) are used to

generate 23 bytes of parity. This parity is appended in one page allowing for recovery when any single page is lost in transmission.

4.1.4.2. Exclusive Or

This method of FEC is experimental and created by one of the authors during initial prototyping. Note that this format of FEC requires a 10-page authentication message, currently unsupported by ASTM.

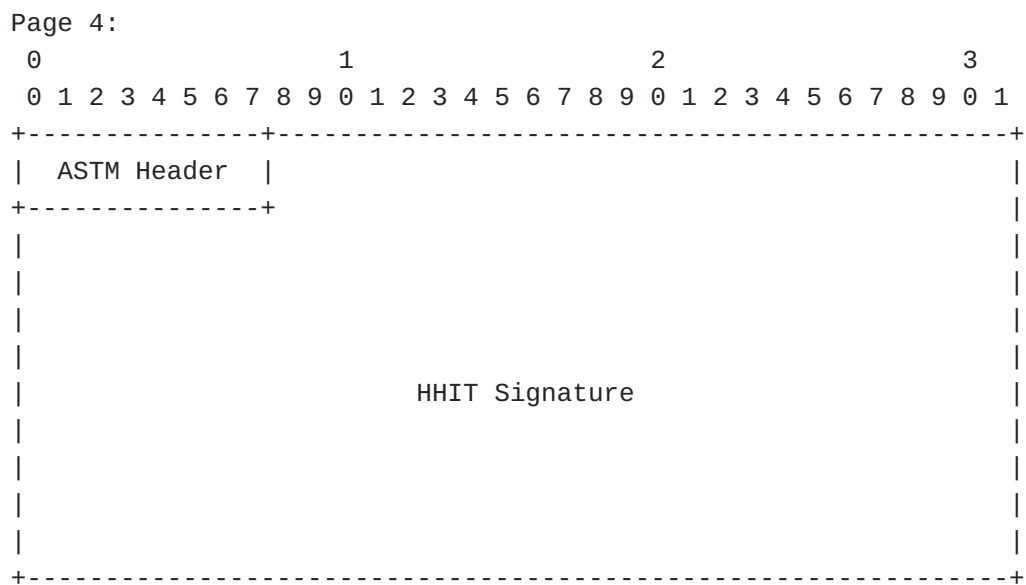


The above illustration shows the relationship between the 10 page authentication message and the XOR operation performed.

The advantage of this format is that a large percentage of the pages can be lost in transmission and recovered using other pages received.

The authors are well aware that this has a high overhead (100%) that might be avoided using other methods. In some cases where wireless links are highly congested or weak this could potentially prove useful.

4.2. Signed Hash Lists



H-Alg, H-Len: (4 bits), (4 bits)

These are fields for relaying information of the Hash algorithm used for the messages and the Hash length (in octets). For this example of the format a length of 4 bytes is used.

H-Alg	Values
-----	-----
RESERVED	0
cSHAKE128	1 [SP800-185] (RECOMMENDED)

Hash of Previous Auth. Message: (4 bytes)

A hash of the previously sent Authentication message.

Hash of Current Auth. Message: (4 bytes)

A hash of the current Authentication message.

Message Hash: (4 bytes)

A hash of a previously sent message.

HHIT Signature: (64 bytes)

EdDSA25519 signature using an EdDSA25519-based HI from HIP.

This format is designed to provide provenance to Broadcast RID messages sent by a given UAS. It should be noted that the HHIT is not provided in the format like others specified here - instead it must be obtained via the Basic ID Message in a detached fashion.

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. The signature is signed across the list of hashes.

4.2.1. Hash Operation

With cSHAKE128 [NIST SP 800-185](#) [[NIST.SP.800-185](#)], the hash is computed as follows:

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

The message MAC is prepended to the message, as the MAC is the only information that links a UA's messages from a specific UA.

4.2.2. Pseudo-blockchain Hashes

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

In regards to the creation and use of the current authentication hash field:

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.

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

4.2.3. Limitations

With the current format proposed by ASTM only 7 messages can be hashed reasonably in the above format. PADDING and redundant H-Alg, H-Len fields could be removed. This would increase the total list of hashes to 9 while losing word alignment of the hashes in each page.

To address this problem properly the authors feel that the Authentication Messages needs to have a max bound of 10 pages, instead of 5.

4.3. Offline Claim

```

0      1      2      3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  ASTM Header  |                                     |
+-----+-----+ ASTM Authentication Headers +-----+-----+
|                                     |               |
+-----+-----+-----+-----+-----+-----+
|
|
|
|
|      Claim: Registry on Aircraft (Bytes 1-17)
|
|
|
+-----+-----+-----+-----+-----+-----+

```

```

0                                     1                                     2                                     3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|  ASTM Header  |
+-----+-----+
|
|
|
|
|      Claim: Registry on Aircraft (Bytes 18-200)
|
|
|
|
|
+-----+-----+-----+-----+

```

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|  ASTM Header  |
+-----+-----+-----+
|
|
|
|          Foward Error Correction
|
|
|
|

```

| |
+-----+

Certificate: Registry on Aircraft (200 bytes):

A certificate granted by the Registry that asserts the binding of UA to the given Registry. Note that Page 8's final byte is actually padding.

Foward Error Correction (FEC) (23 bytes):

FEC on the previous 9 pages.

This specific format does not currently fit within the ASTM specification. Requiring a minimum of 200 bytes, this would require the Authentication Message to have 10 pages, instead of the current 5 page limit. The rest of this section assumes 10 pages to demonstrate how this message is laid out and works.

What this grants is the ability to authenticate UA information when the receiving device of the observer (e.g. a smartphone with a dedicated RID application) has no Internet service (e.g. LTE signal).

By including the device HI along with a signature from the registry the UA is under, we can assert trust of a given UA without requiring the need for immediate reverse lookups online.

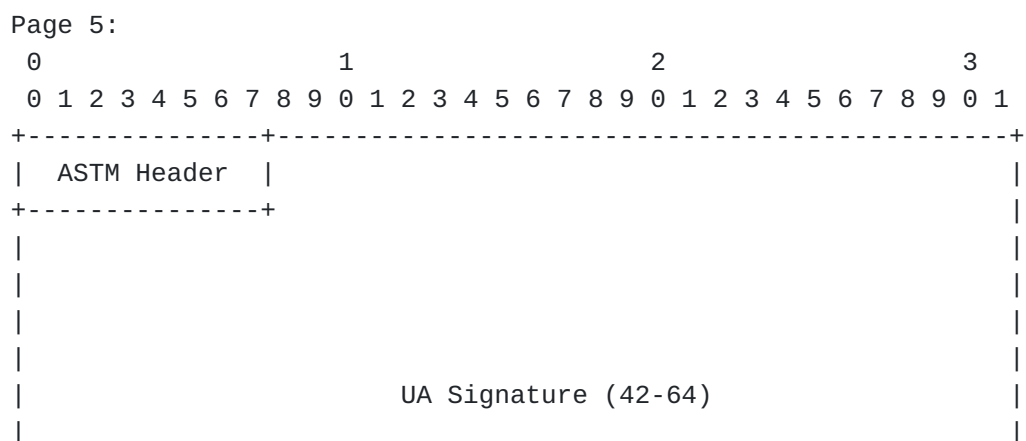
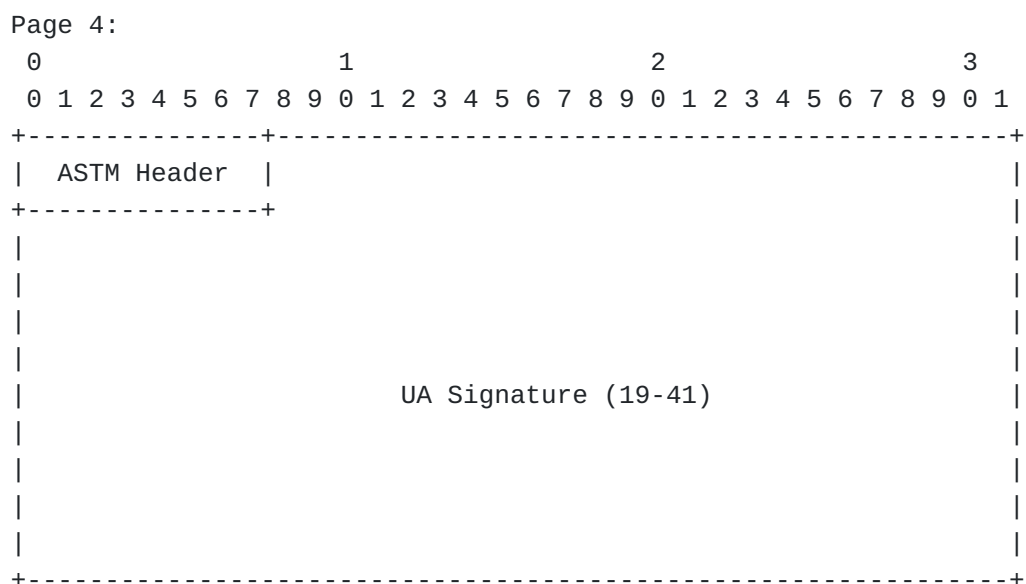
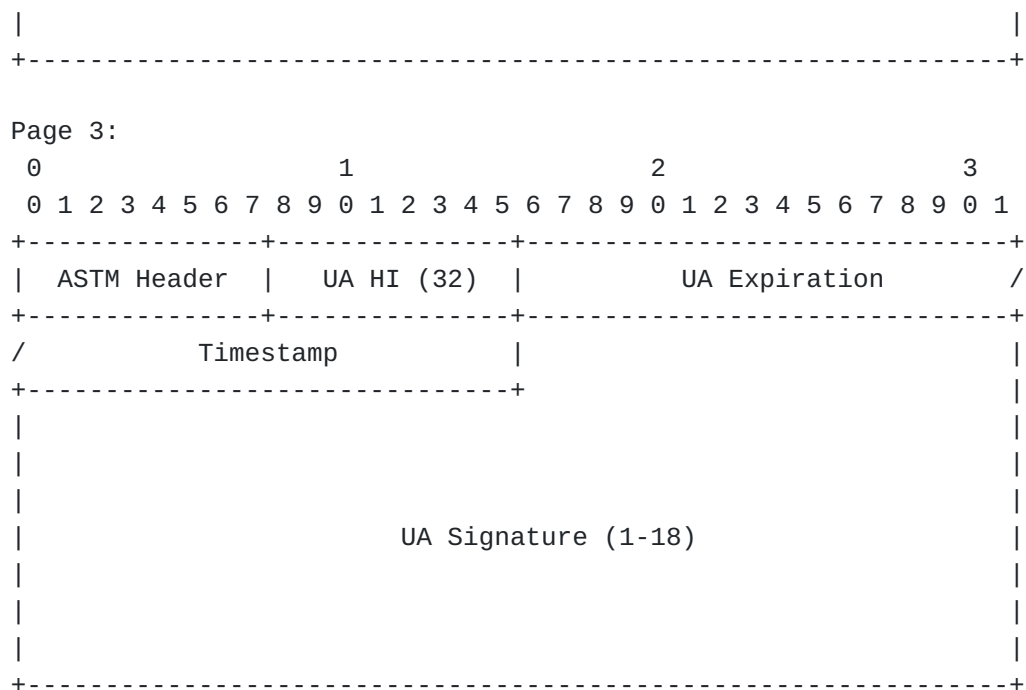
4.3.1. Claim: Registry on Aircraft

The bulk of this message is made up with the 200 byte claim titled "Registry on Aircraft" (henceforth referred to as "Cra"). Below is Cra fully marked out inside the ASTM Authentication Message.

[illegible]

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
+-----+-----+-----+-----+-----+-----+-----+-----+																															
ASTM Header																															
+-----+-----+-----+-----+-----+-----+-----+-----+																															
																UA HHIT (2-16)															
+-----+-----+-----+-----+-----+-----+-----+-----+																															
																UA HI (1-8)															
+-----+-----+-----+-----+-----+-----+-----+-----+																															

[illegible]



```

|
|
|
+-----+

```

Page 6:

```

      0              1              2              3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|  ASTM Header  | Registry Expiration Timestamp | /
+-----+-----+-----+-----+
/              |
+-----+-----+-----+-----+
|
|
|
| Registry Signature (1-19)
|
|
|
+-----+-----+-----+-----+

```

Page 7:

```

      0              1              2              3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|  ASTM Header  |
+-----+-----+-----+-----+
|
|
|
| Registry Signature (20-42)
|
|
|
+-----+-----+-----+-----+

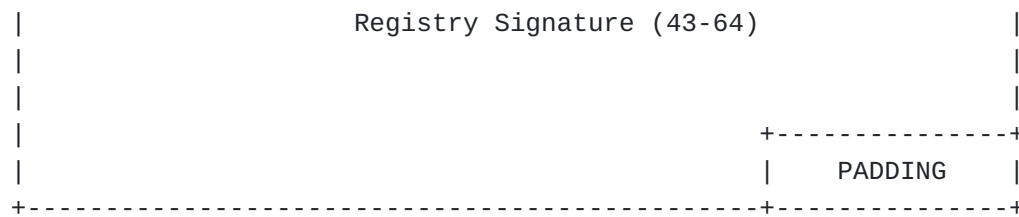
```

Page 8:

```

      0              1              2              3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|  ASTM Header  |
+-----+-----+-----+-----+
|
|
|
|
+-----+-----+-----+-----+

```



Registry HHIT (16 bytes):

Registry HHIT that UA is registered under and handle for the HI used in the signing of full certificate.

Claim: Aircraft on Aircraft [Cra] (116 bytes):

UA HHIT (16 bytes):

Nominally the Aircraft HHIT (specified here as UA)

UA HI (32 bytes):

Public half of EdDSA-25519 asymmetric keypair used to sign Caa.

UA Expiration Timestamp (4 bytes):

UNIX current time (at time of signing) + offset

UA Signature (64 bytes):

Signature with UA keypair using the data of UA HHIT, UA HI and UA Expiration Timestamp.

Registry Expiration Timestamp (4 bytes):

UNIX current time (at time of signing) + offset

Registry Signature (64 bytes):

Signature with Registry keypair using the data of
Registry HHIT, Caa and Registry Expiration Timestamp.

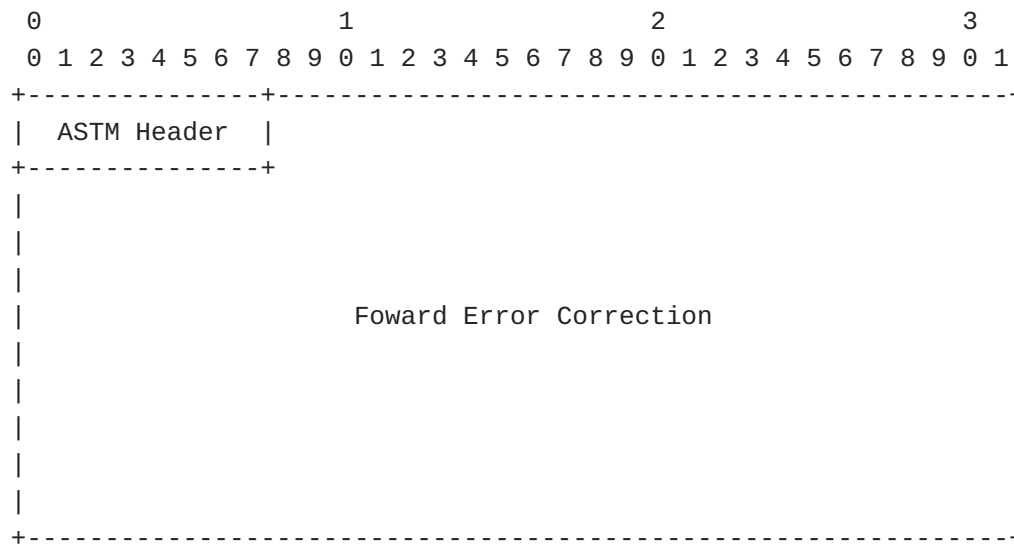
Cra is in practice a binding claim between the Registry and the Aircraft, asserting the relationship between the two entities. Cra signs another claim, Caa (Claim: Aircraft on Aircraft), that is created during UA provisioning.

Importantly this claim allows offline signature verification from the UA. This is as the UA HI is included in the claim. Also included 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 Caa, Cra, other claims and the provisioning process can be found in [[I-D.wiethuechter-tmrid-identity-claims](#)].

4.3.2. Foward Error Correction

Page 9:



The prefered form of FEC is using Reed Solomon. With this the entire authentication message (all pages, including headers) are used to generate 23 bytes of parity. This parity is inserted into the final page of the Claim format for recovery when any single page is lost in transmission.

5. Example Use Cases

This section introduces potential use cases of the HIP based extensions to the proposed ASTM standard authentication message.

5.1. Trusted Messages

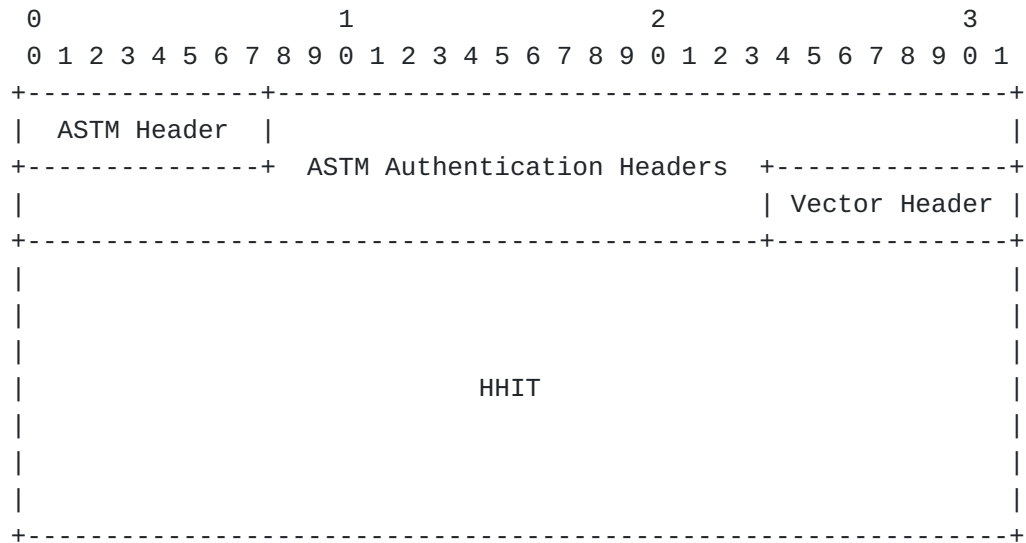
Using the HIP Based Authentication Wrapper any single Broadcast RID message defined by ASTM can become what the authors refer to as a "Trusted Message".

One specific use case that is useful in the UAS RID space is the creation of a "Trusted Vector Message". By placing a previous [or new] vector message into the payload section of this format a verifiable broadcast can be created.

Due to being signed this creates an authentic vector that is hard to spoof, which can confirm flight paths in near real time.

The figure below is a example of a "Trusted Vector Message".

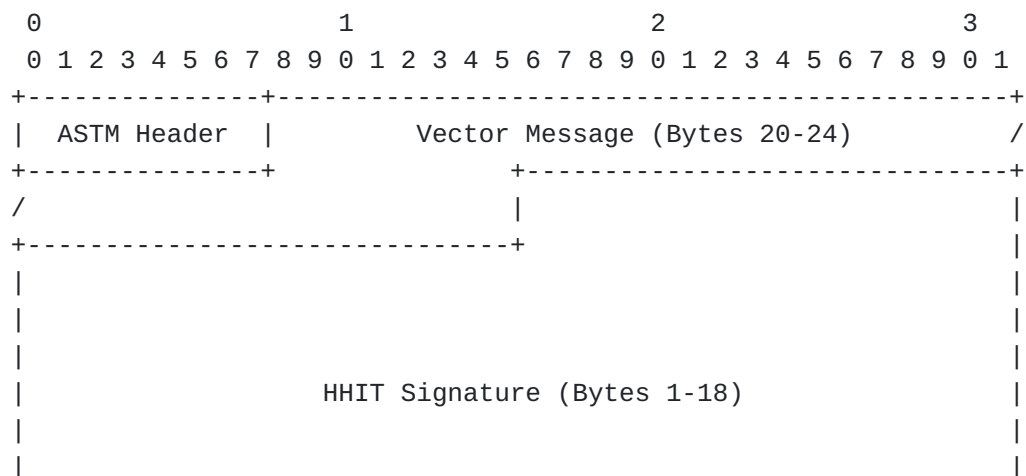
Page 0:



Page 1:



Page 2:



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

Page 3:

```
      0              1              2              3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|  ASTM Header  |
+-----+-----+-----+-----+
|
|
|
|
|      HHIT Signature (Bytes 19-41)
|
|
|
|
+-----+-----+-----+-----+
```

Page 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
+-----+-----+-----+-----+
|  ASTM Header  |
+-----+-----+-----+-----+
|
|
|
|
|      HHIT Signature (Bytes 42-64)
|
|
|
|
+-----+-----+-----+-----+
```


5.2. Wrapped Signed Hashes

Using the Message Wrapper a [short] list of hashes can be signed. These hashes are of previous individual RID messages.

This follows the format of the Signed Hash List, excluding the psuedo-blockchain hashes found in the Signed Hash List.

To the authors, this format has limited use due to numerous concerns of replay attacks. It is suggested to instead use the full Signed Hash List format.

[illegible]

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	2	3	4	5	6	7	8	9
+-----+										+-----+										+-----+										+-----+									
ASTM Header																				Trust Timestamp										/									
+-----+										+-----+										+-----+										+-----+									
/																				Message Hash										/									
+-----+										+-----+										+-----+										+-----+									
/																				Message Hash										/									
+-----+										+-----+										+-----+										+-----+									
/																				Message Hash										/									
+-----+										+-----+										+-----+										+-----+									
/																				Message Hash										/									
+-----+										+-----+										+-----+										+-----+									

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	2	3	4	5	6	7	8	9
+-----+										+-----+										+-----+										+-----+									
ASTM Header /																				Message Hash										/									
+-----+										+-----+										+-----+										+-----+									
/																																							
+-----+																																							
										HHIT Signature (Bytes 1-18)																													

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

Page 3:

```
      0              1              2              3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|  ASTM Header  |
+-----+-----+-----+-----+
|
|
|
|
|      HHIT Signature (Bytes 19-41)
|
|
|
|
+-----+-----+-----+-----+
```

Page 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
+-----+-----+-----+-----+
|  ASTM Header  |
+-----+-----+-----+-----+
|
|
|
|
|      HHIT Signature (Bytes 42-64)
|
|
|
|
+-----+-----+-----+-----+
```

6. IANA Considerations

TBD

7. Security Considerations

TBD

8. Acknowledgments

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

9. References

9.1. Normative References

[NIST.SP.800-185] Kelsey, J., Change, S., and R. Perlner, "SHA-3 derived functions: cSHAKE, KMAC, TupleHash and ParallelHash", National Institute of Standards and Technology report, DOI 10.6028/nist.sp.800-185, December 2016, <<https://doi.org/10.6028/nist.sp.800-185>>.

[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>>.

9.2. Informative References

[I-D.moskowitz-hip-hierarchical-hit]

Moskowitz, R., Card, S., and A. Wiethuechter, "Hierarchical HITs for HIPv2", Work in Progress, Internet-Draft, draft-moskowitz-hip-hierarchical-hit-04, 3 March 2020, <<https://tools.ietf.org/html/draft-moskowitz-hip-hierarchical-hit-04>>.

[I-D.moskowitz-hip-new-crypto]

Moskowitz, R., Card, S., and A. Wiethuechter, "New Cryptographic Algorithms for HIP", Work in Progress, Internet-Draft, draft-moskowitz-hip-new-crypto-04, 23

January 2020, <<https://tools.ietf.org/html/draft-moskowitz-hip-new-crypto-04>>.

[I-D.wiethuechter-tmrid-identity-claims]

Wiethuechter, A., Card, S., and R. Moskowitz, "TMRID Identity Claims", Work in Progress, Internet-Draft, draft-wiethuechter-tmrid-identity-claims-01, 9 March 2020, <<https://tools.ietf.org/html/draft-wiethuechter-tmrid-identity-claims-01>>.

[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
4947 Commercial Drive
Yorkville, NY 13495
United States of America

Email: adam.wiethuechter@axenterprize.com

Stuart W. Card
AX Enterprize
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