Workgroup: DRIP Working Group
Internet-Draft: draft-ietf-drip-auth-02
Published: 21 October 2021
Intended Status: Standards Track
Expires: 24 April 2022
Authors: A. Wiethuechter
AX Enterprize, LLC
S. Card
AX Enterprize, LLC
R. Moskowitz
HTT Consulting

DRIP Authentication Formats for Broadcast RID

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 24 April 2022.

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

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 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 5.3 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.

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.

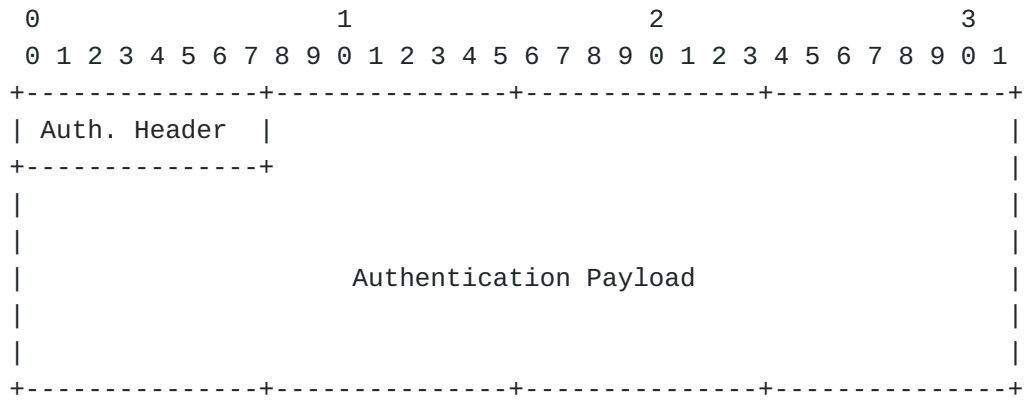
The following subsections will provide a high level reference to the ASTM standard for Authentication Messages and how their current limitations effect trust in the Broadcast RID environment.

3.2. Reasoning for IETF DRIP Authentication

The ASTM Authentication Message has provisions in [F3411-19] to allow for other organizations to define (and standardize) Authentication formats. The standardization of special 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 by ASTM or others) was flexible enough to satisfy this goal resulting in the work reflected in this document.

3.3. ASTM Authentication Message

The ASTM Authentication Message is a unique message in the Broadcast F3411 standard as it is the only one that is paged.

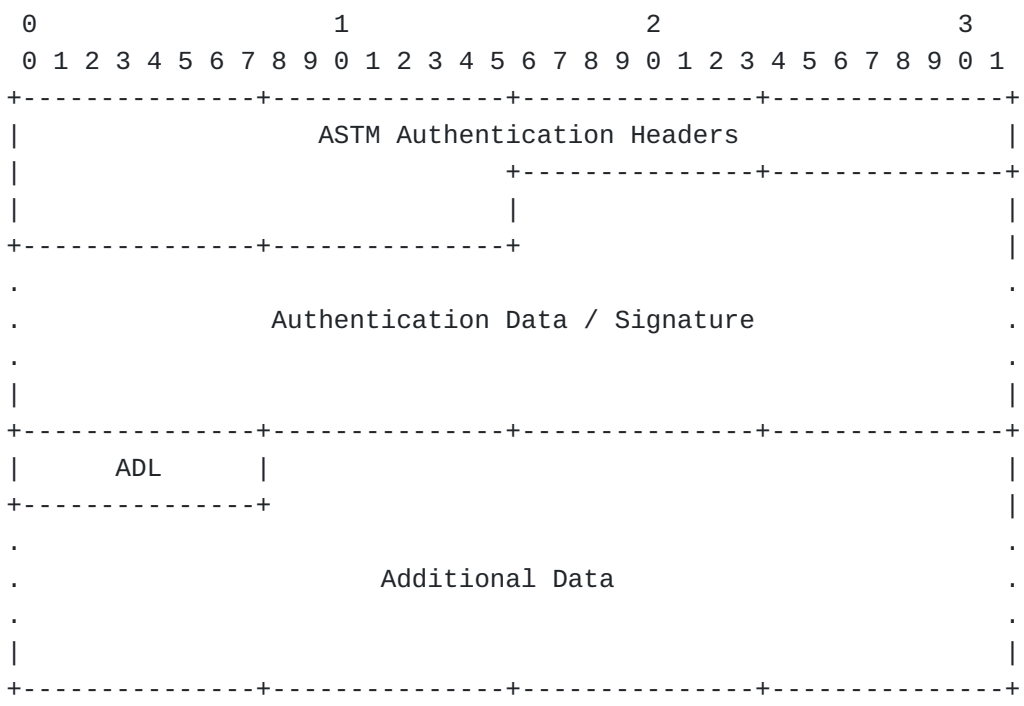


Auth. Header: (1 byte)
Contains Authentication Type (AuthType) and Page Number.

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

Figure 1: Standard ASTM Authentication Message Page

Figure 1 is an abstracted view of a single Authentication Page used in an Authentication Message. There is a max of 16 pages (indexed 0 to 15 in the first byte; Auth. Header) that can be sent for a single Authentication Message, with each page carrying a max 23-byte Authentication Payload.



ASTM Authentication Headers: (6 bytes)
    Contains other header information for the Authentication
    Message from ASTM UAS RID Standard.

Authentication Data / Signature: (0 to 255 bytes)
    Opaque authentication data.

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

Additional Data: (0 to 255 bytes):
    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 abstract view of the data fields found in the Authentication Message as defined by [F3411-19]. This data is placed into Figure 1's Authentication Payload, potentially 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.

Full examples of Authentication Messages (fully paginated; both with and without Additional Data) can be found in Appendix E.

4. DRIP Authentication Formats

To keep consistent formatting across the different mediums (Bluetooth 4, Bluetooth 5 and Wi-Fi NaN) and their independent restrictions the authentication data being sent is REQUIRED to fit within the first 9 pages (Page 0 through Page 8) of the Authentication Message. Under DRIP, the Length field in the ASTM Authentication Headers (which denotes the length in bytes of Authentication Data only) MUST NOT exceed the value of 201.

All formats defined in this section fill the Authentication Data / Signature field in Figure 2.

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

4.1. UAS ID Signature

The existing ASTM [F3411-19] Authentication Type 0x1 can be used to send a fresh Self-Attestation of the UA over 7 pages.

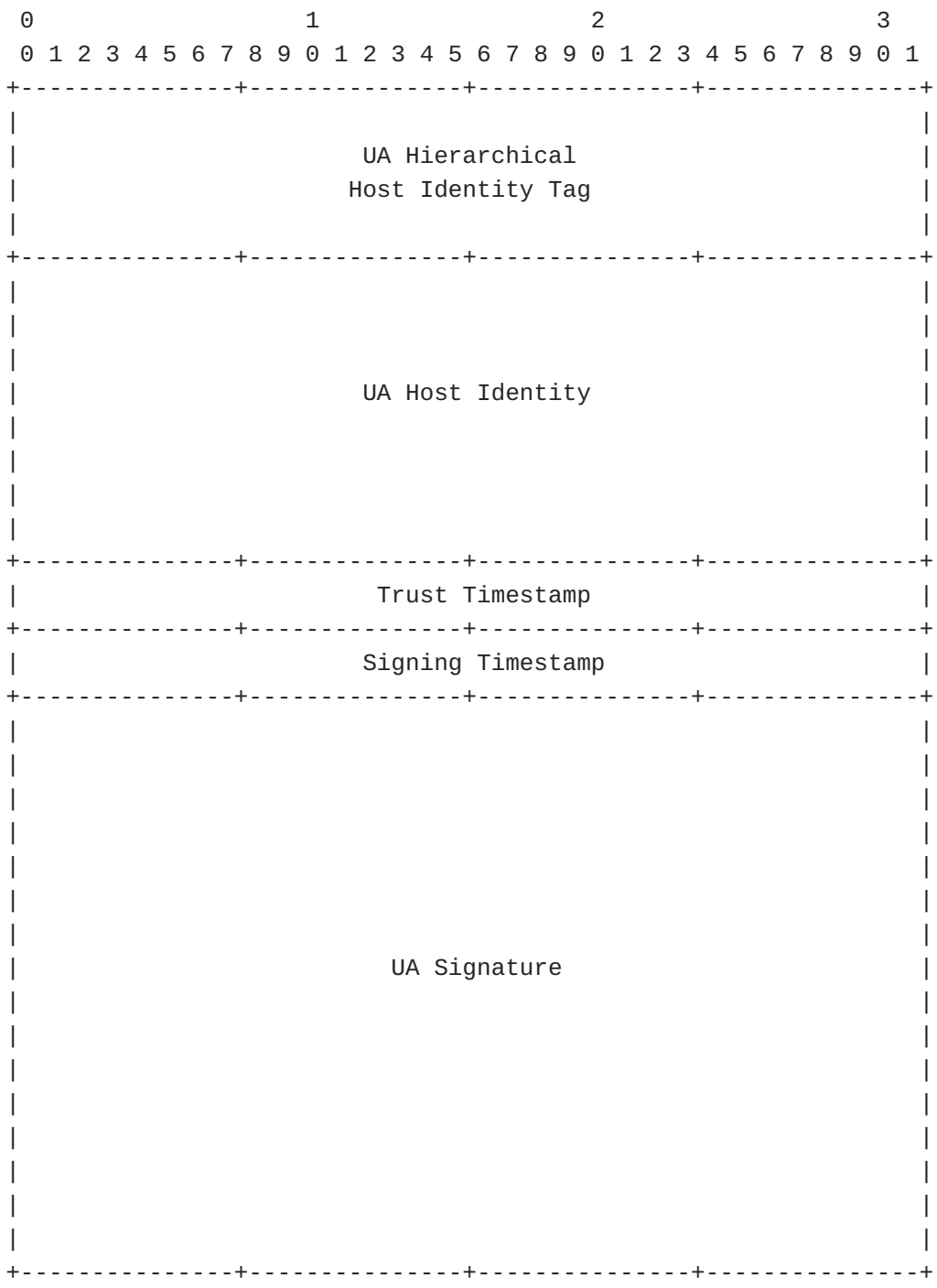


Figure 3: DRIP UAS ID Signature

4.2. Operator ID Signature

The existing ASTM [F3411-19] Authentication Type 0x2 can be used to send a static Self-Attestation of the Operator over 7 pages.

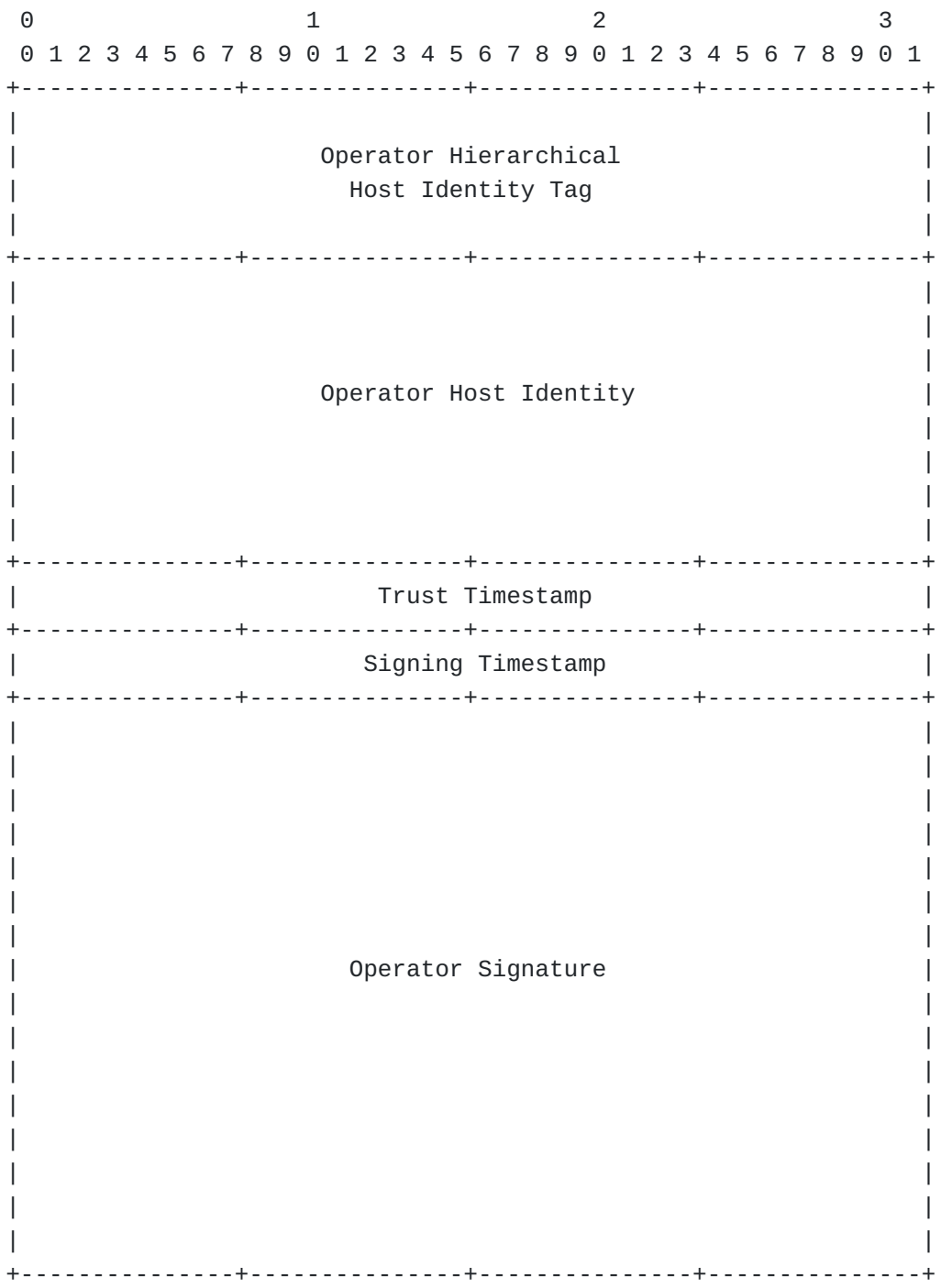


Figure 4: DRIP Operator ID Signature

4.3. Message Set Signature

When running under Extended Advertisements, the existing ASTM [F3411-19] Authentication Type 0x3 can be used to sign over the adjacent ASTM Messages in the Message Pack (0xF).

The concatenation of all messages in the Message Pack (excluding Authentication) before signing MUST be in Message Type order and be placed between the UA HHIT and Trust Timestamp field.

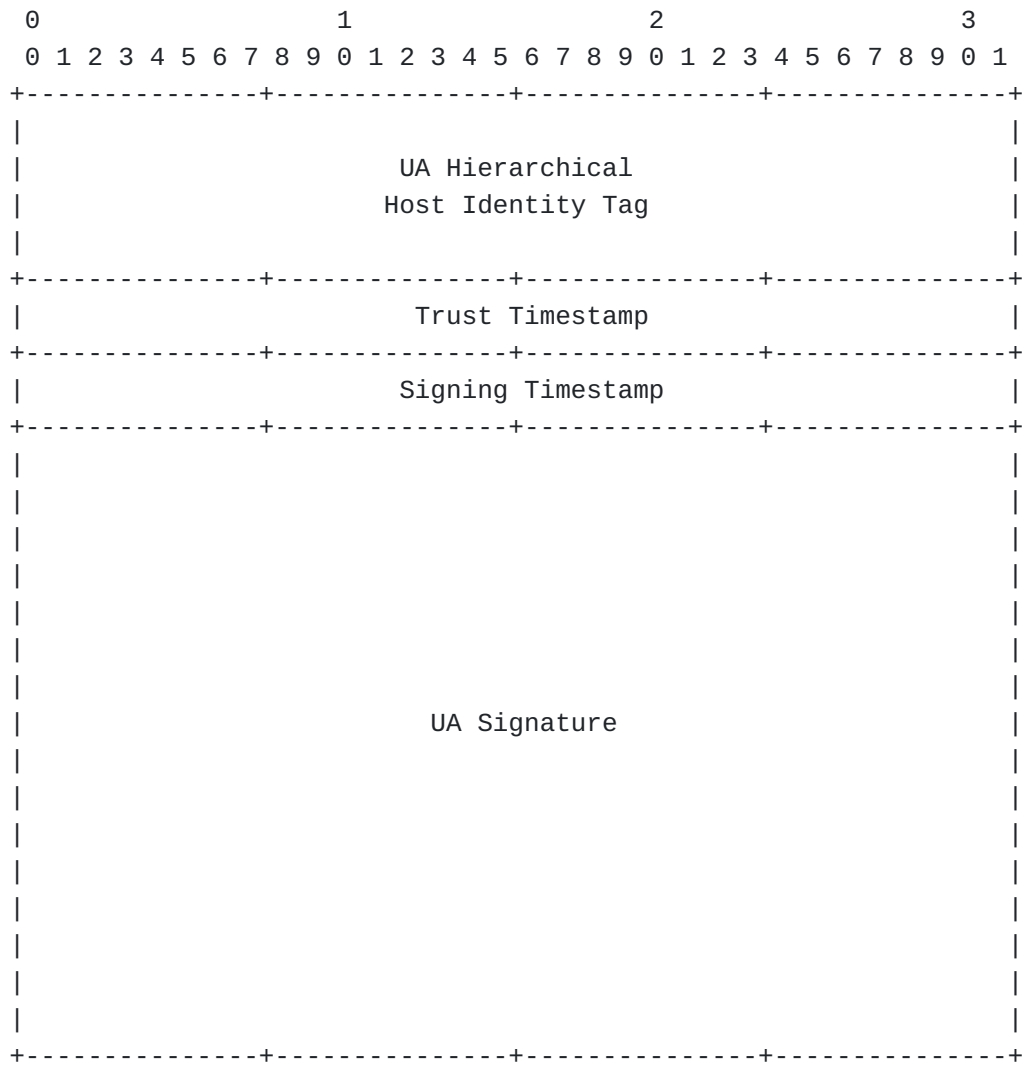


Figure 5: DRIP Message Set Signature

4.4. Specific Method

For ASTM Specific Method (Authentication Type 0x5) a special Specific Authentication Method (SAM) Type is used to multiplex between various formats.

4.4.1. DRIP Frame

This is the general frame to hold DRIP Authentication for Specific Method; containing the SAM Type and laying out the major data fields.

SAM Type (1 byte):
For DRIP there are four SAM Type's allocated:

SAM Type	Value
DRIP Frame	1
DRIP Wrapper	2
DRIP Manifest	3m
DRIP Link	4

DRIP Authentication Data (0 to 200 bytes):
DRIP Authentication data.

Figure 6: DRIP Frame Format

4.4.1.1. Specific Authentication Method (SAM) Type

The SAM Type field is maintained by the International Civil Aviation Organization (ICAO) and for DRIP four are allocated; DRIP Frame (0x01) (Section 4.4.1), DRIP Wrapper (0x02), DRIP Manifest (0x03) and DRIP Link (0x04).

4.4.1.2. DRIP Authentication Data

This field has a maximum size of 200 bytes. When possible the DRIP Broadcast Attestation Structure (Appendix C) should be used in this space.

4.4.2. DRIP Wrapper Format

This is specified when the SAM Type is DRIP Wrapper. It is encapsulated by the DRIP Frame (Section 4.4.1) and Broadcast Attestation Structure (Appendix C); filling the Attestation Data (Appendix C.2) field with full (25-byte) ASTM Messages. The minimum number of ASTM Messages being 1 and the max being 4. The encapsulated ASTM Messages MUST be in Message Type order as defined by ASTM. All message types except Authentication (0x2) and Message Pack (0xF) are allowed.

To determine the number of messages wrapped the receiver can check that the length of the Attestation Data (Appendix C.2) field of the DRIP Broadcast Attestation (Appendix C) is a multiple of 25-bytes.

4.4.2.1. Wrapper Limitations

See Section 5.3.1 for a discussion on a comparison between DRIP Wrapper and DRIP Manifest.

4.4.3. DRIP Manifest Format

This format is specified when SAM Type is set to DRIP Manifest. It is encapsulated by the DRIP Frame (Section 4.4.1) and Broadcast Attestation Structure (Appendix C); filling the Attestation Data (Appendix C.2) field with 8-byte hashes of previous ASTM 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. This is a way to evade the limitation of a maximum of 4 messages in the Wrapper Format and reduce overhead.

(Editors Note: Manifests MUST NOT be of a length multiple of 25-bytes or 48-bytes.)

4.4.3.1. Hash Algorithms 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(Message, 128, "", "Remote ID Auth Hash")
```

4.4.3.2. 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.

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

See Section 5.3.1 for a discussion on a comparison between DRIP Wrapper and DRIP Manifest.

Another limitation is the length of hash, which is discussed in Section 8.1.

4.4.4. DRIP Link Format

This format is specified when SAM Type is set to DRIP Link. It is encapsulated by the DRIP Frame (Section 4.4.1) and Broadcast Attestation Structure (Appendix C) but the attestation has already taken place, thus the UA need not dynamically sign the structure.

For details on the Broadcast Attestation see [drip-rid] and Appendix B.6.

4.4.4.1. Link Limitations

See Section 8.2 for details on why this structure is not dynamically signed.

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 (Appendix D) MUST be used 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.

5.2. Extended Advertisements (Bluetooth 5.X, Wi-Fi NaN, Wi-Fi Beacon)

Under the ASTM specification, Bluetooth 5 or Wi-Fi NaN transport of Remote ID is to use the Message Pack (Type 0xF) format for all transmissions. Under Message Pack 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 (Appendix D) MUST NOT be used as it is impractical.

5.3. DRIP Recommendations

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

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

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

3. DRIP Link using the Broadcast Attestation of 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 (satisfies GEN-1 and GEN-2)

5.3.1. DRIP Wrapper vs. DRIP Manifest

(Editors Note: review this to confirm numbers are still accurate.)

A single Manifest can carry at most (using the full 9-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.

6. ICAO Considerations

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

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

7. IANA Considerations

This document does not require any actions by IANA.

8. Security Considerations

8.1. Manifest Hash Length

For DRIP Manifest an 8-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 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.

8.2. 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 aircraft 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.

8.3. Trust Timestamp Offsets

Note the discussion of Trust Timestamp Offsets here is in context of the DRIP Wrapper (Section 4.4.2) and DRIP Manifest (Section 4.4.3) messages. For DRIP Link (Section 4.4.4) messages these offsets are set by the Attestor (typically a registry) and have their own set of considerations as seen in (Editors Note: link to registry draft security considerations here).

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.

9. 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 WiFi protocols would not 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.

10. References

10.1. 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", 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>.

10.2. Informative References

[drip-requirements] Card, S. W., Wiethuechter, A., Moskowitz, R., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Requirements", Work in Progress, Internet-Draft, draft-ietf-drip-reqs-18, 8 September 2021, <https://www.ietf.org/archive/id/draft-ietf-drip-reqs-18.txt>.

[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>.

[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, <https://www.ietf.org/archive/id/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>.

Appendix A. Thoughts on ASTM Authentication Message

(Editor Note: is this valid anymore to keep?)

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.

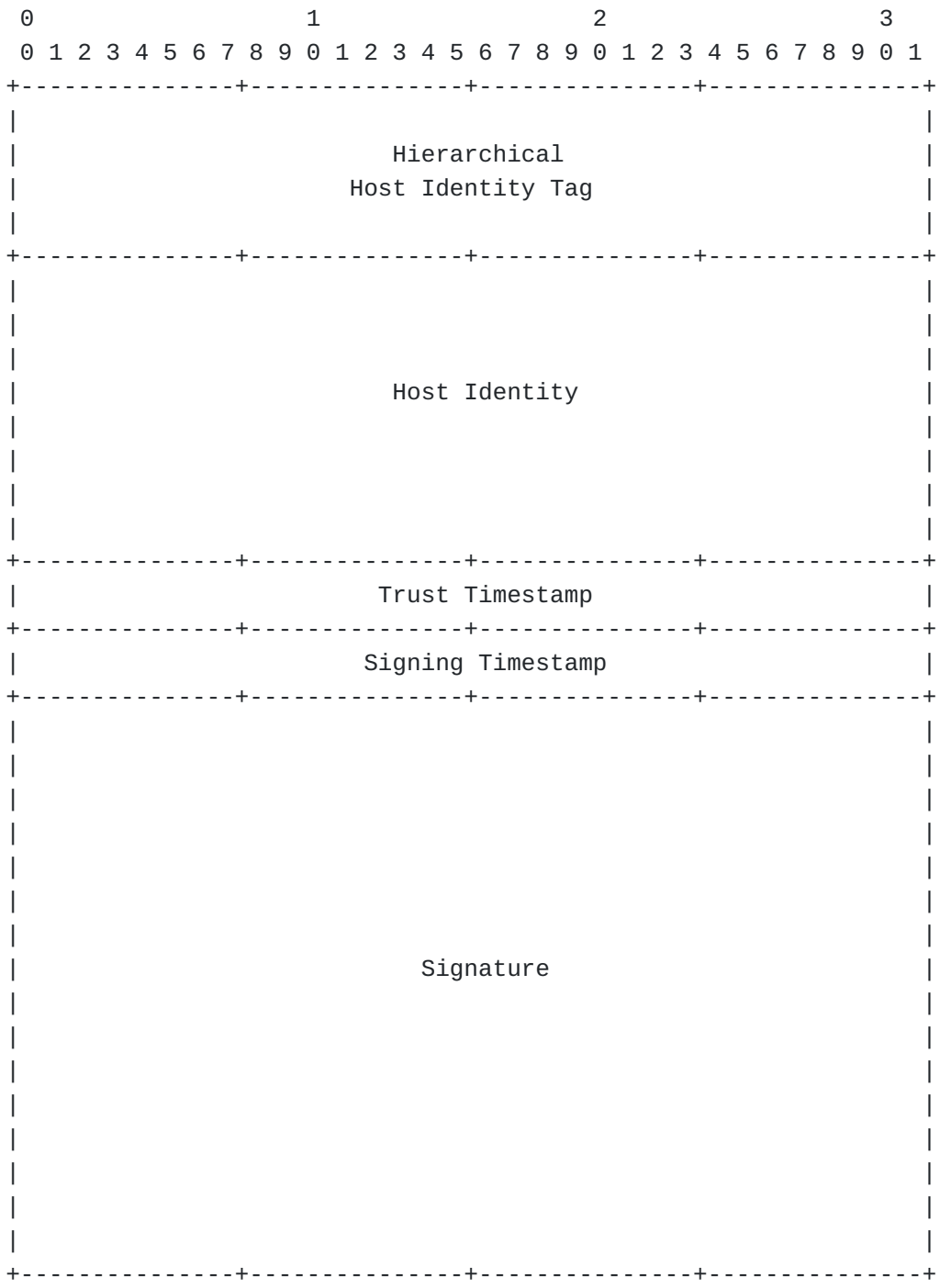
Another issue with this format, not fully addressed in this document is fragmentation. Under Bluetooth 4.X, each page is sent separately which can result in lose of pages on the receiver. This is disastrous as the loss of even a single page means any signature is incomplete.

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.

Appendix B. DRIP Attestations

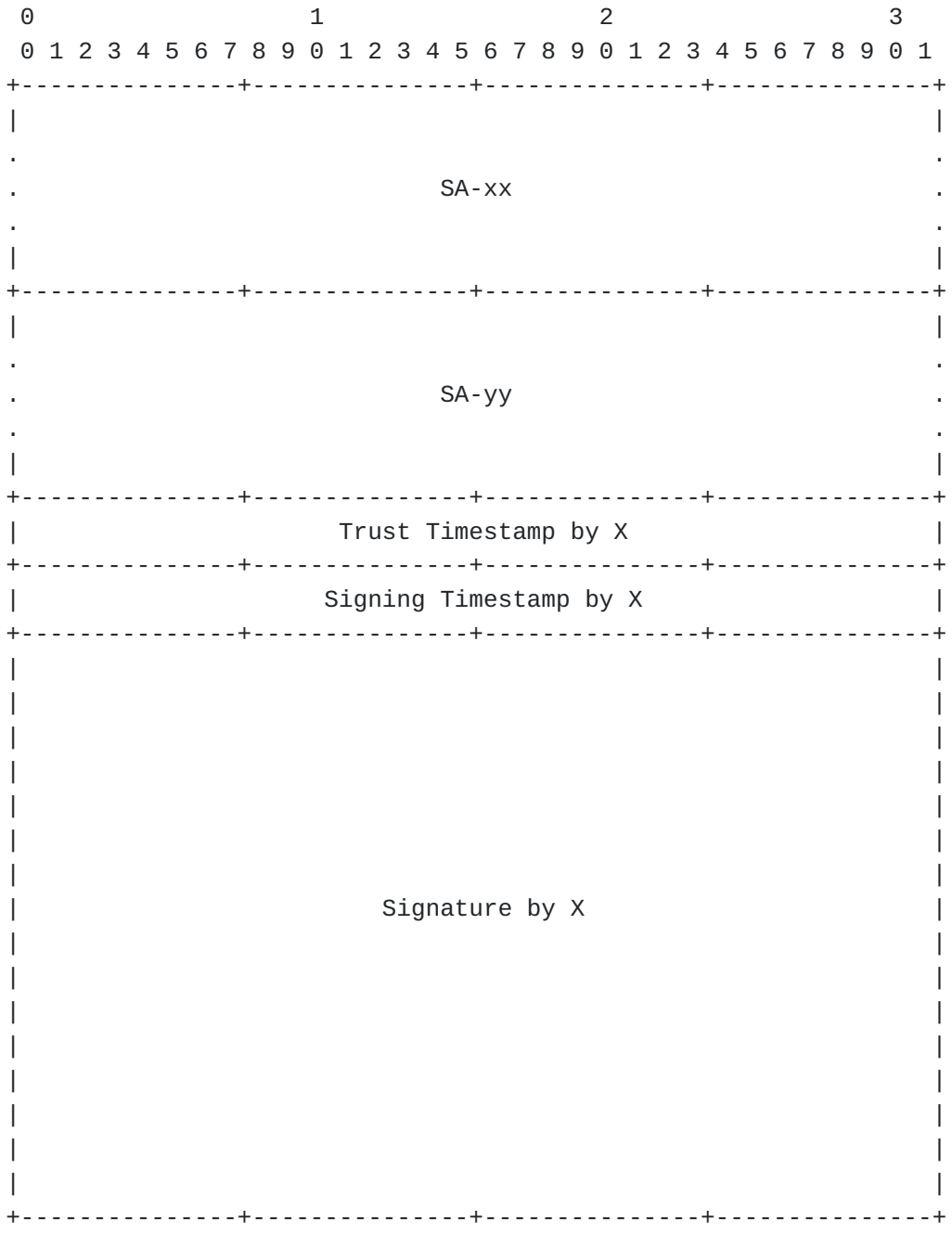
B.1. Self-Attestation (SA-xx)



Length = 120-bytes

Figure 7: DRIP Self-Attestation

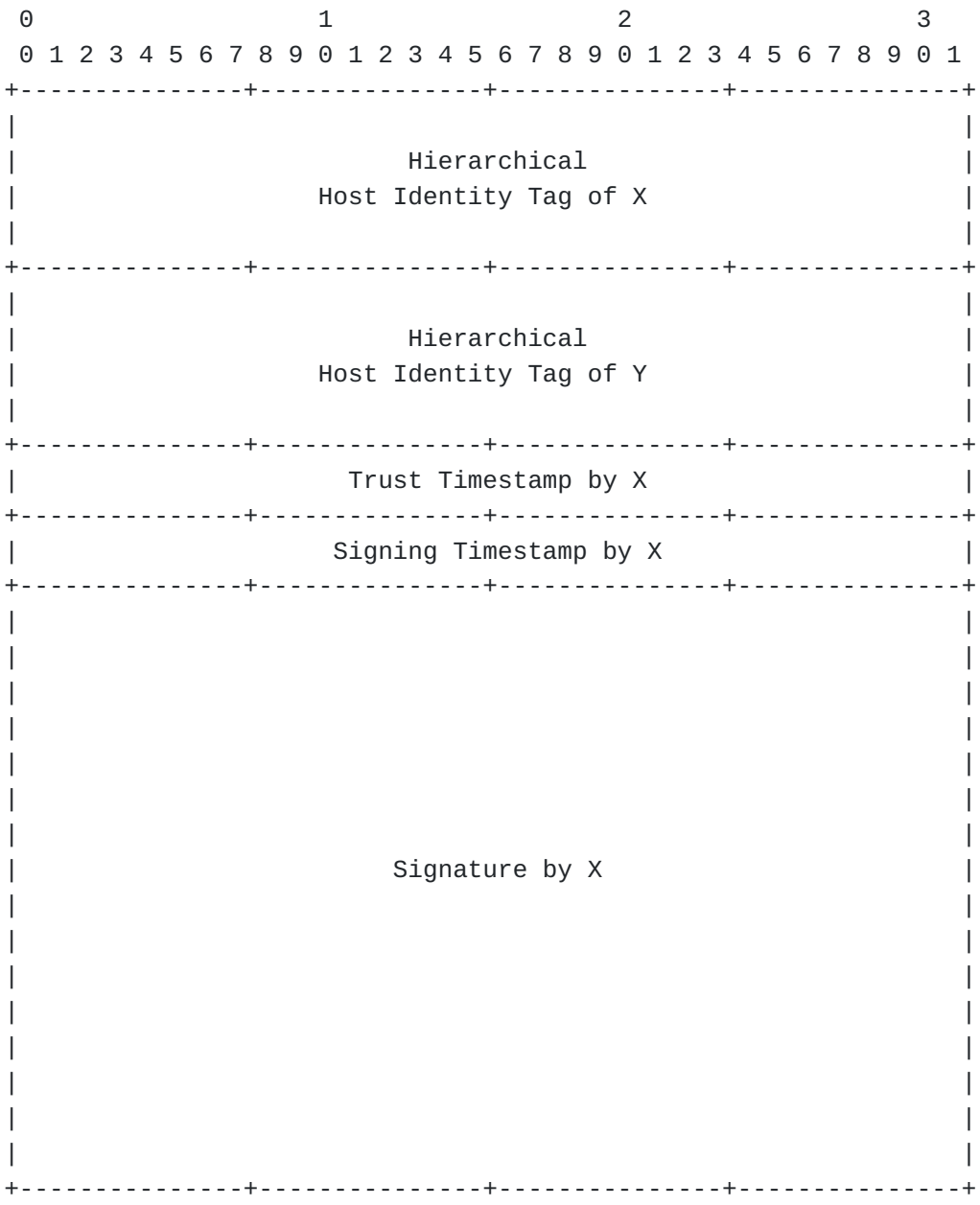
B.2. Attestation (A-xy)



Length = 312-bytes

Figure 8: DRIP Attestation

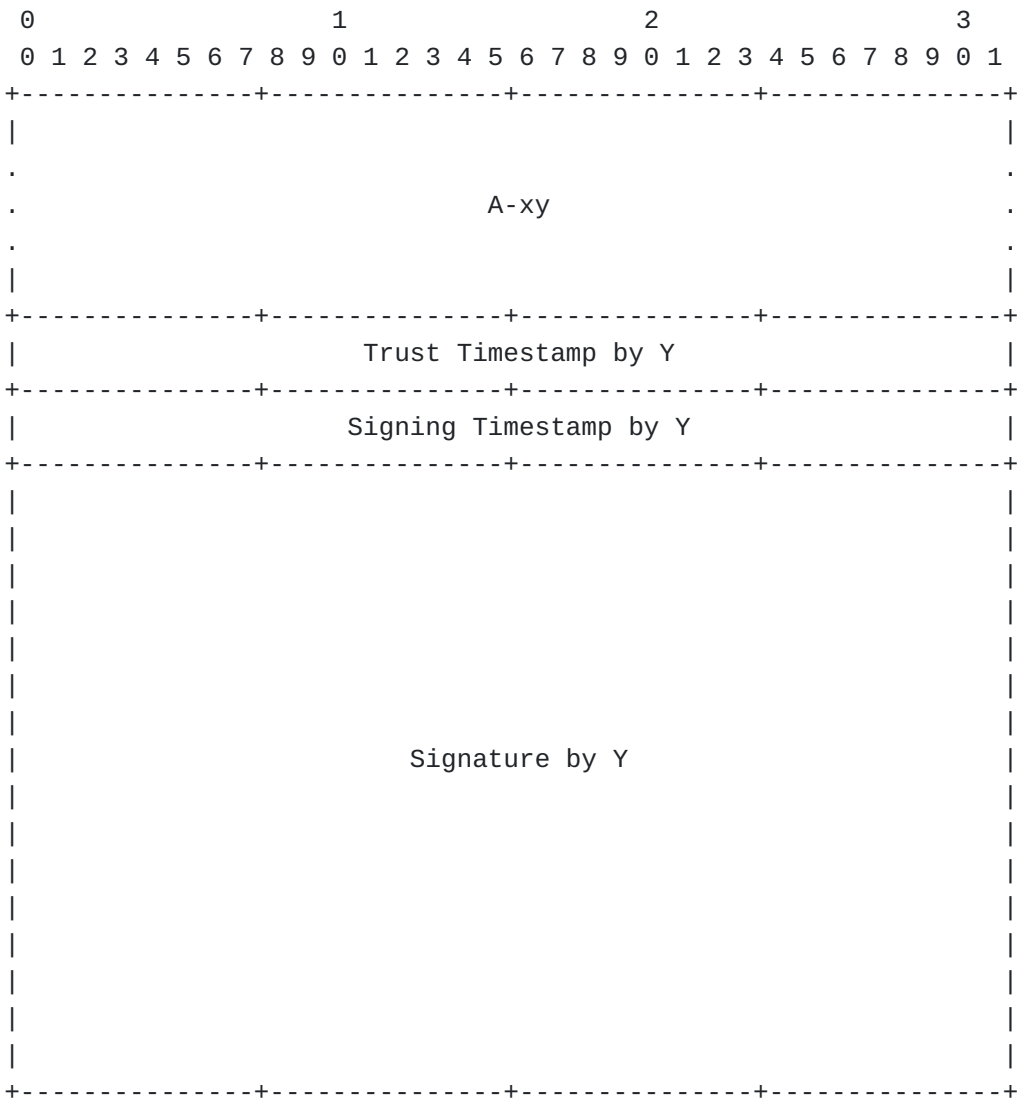
B.3. Concise Attestation (CA-xy)



Length = 104-bytes

Figure 9: DRIP Concise Attestation

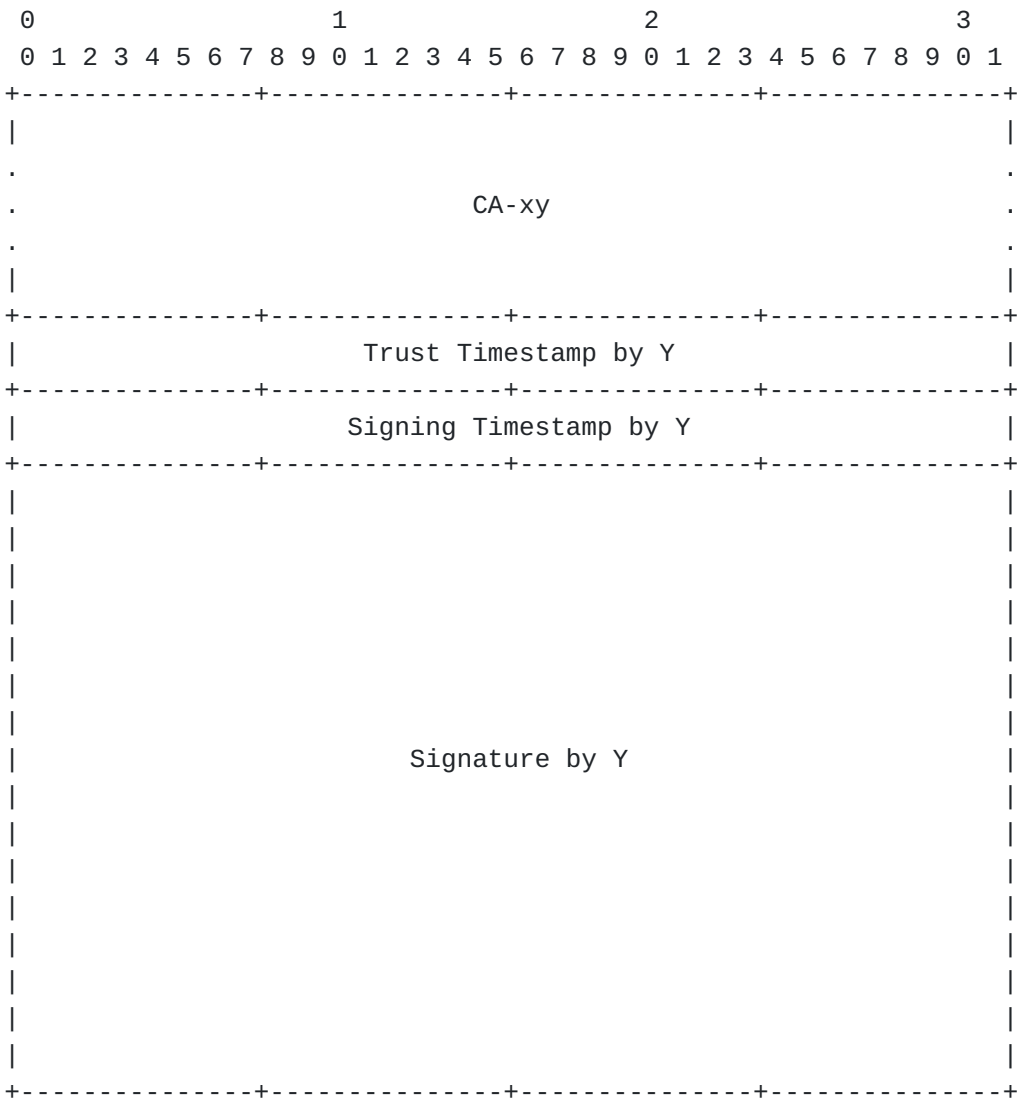
B.4. Mutual Attestation (MA-xy)



Length = 384-bytes

Figure 10: DRIP Mutual Attestation

B.5. Link Attestation (LA-xy)



Length = 176-bytes

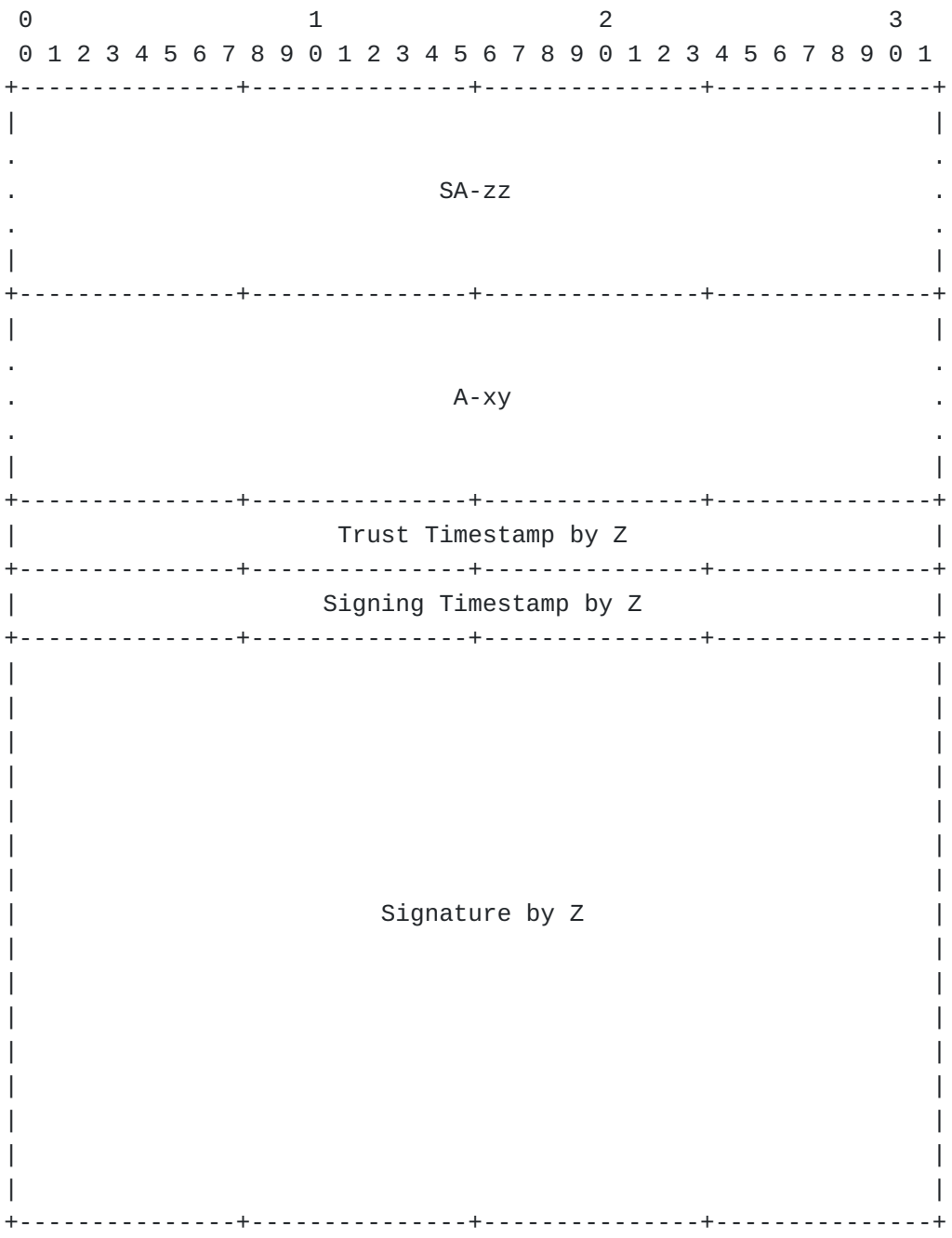
Figure 11: DRIP Link Attestation

B.6. Broadcast Attestation (BA-xy)

Length = 136-bytes

Figure 12: DRIP Broadcast Attestation

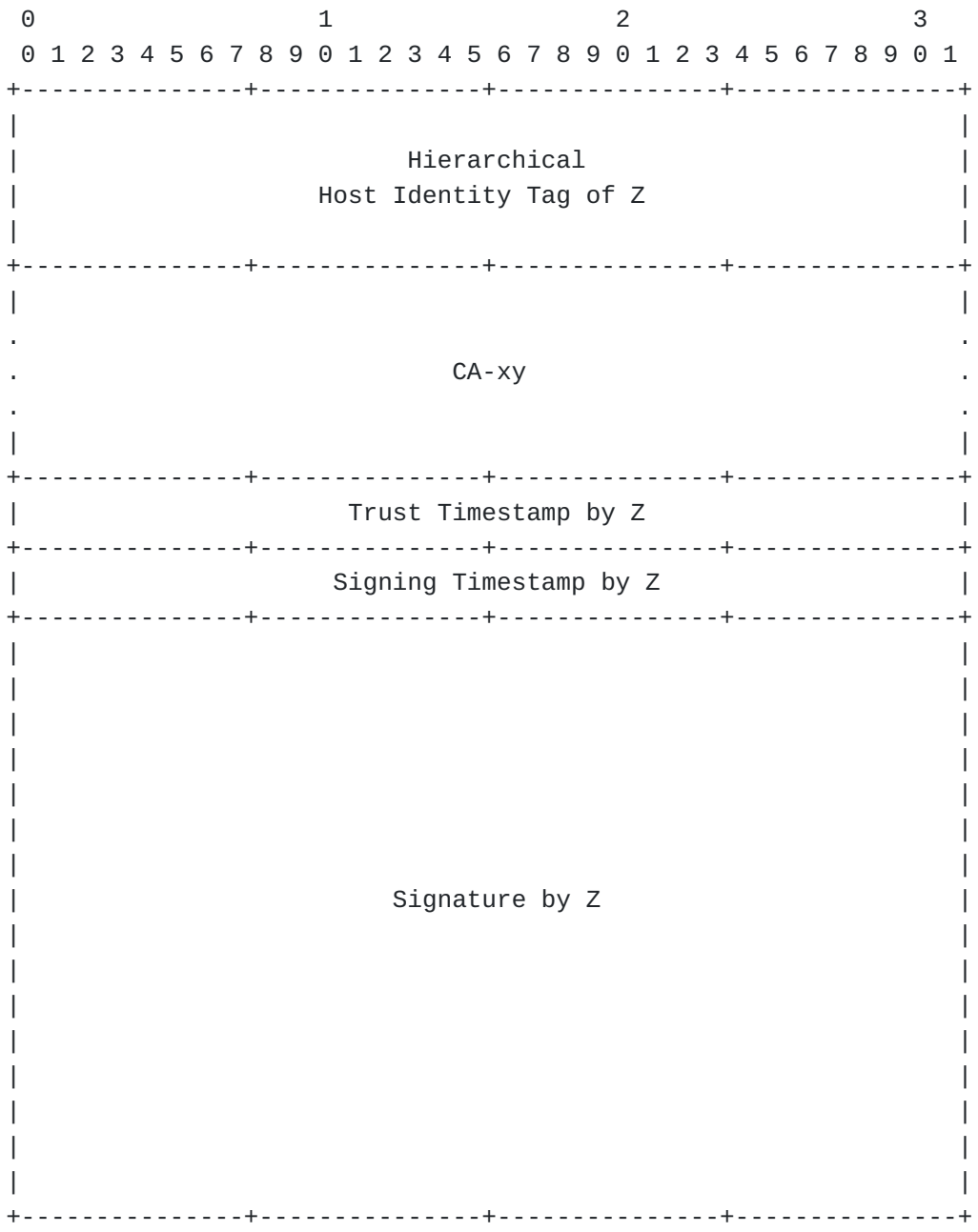
B.7. Attestation Certificate (AC-zxy)



Length = 504-bytes

Figure 13: DRIP Attestation Certificate

B.8. Concise Certificate (CC-zxy)



Length = 192-bytes

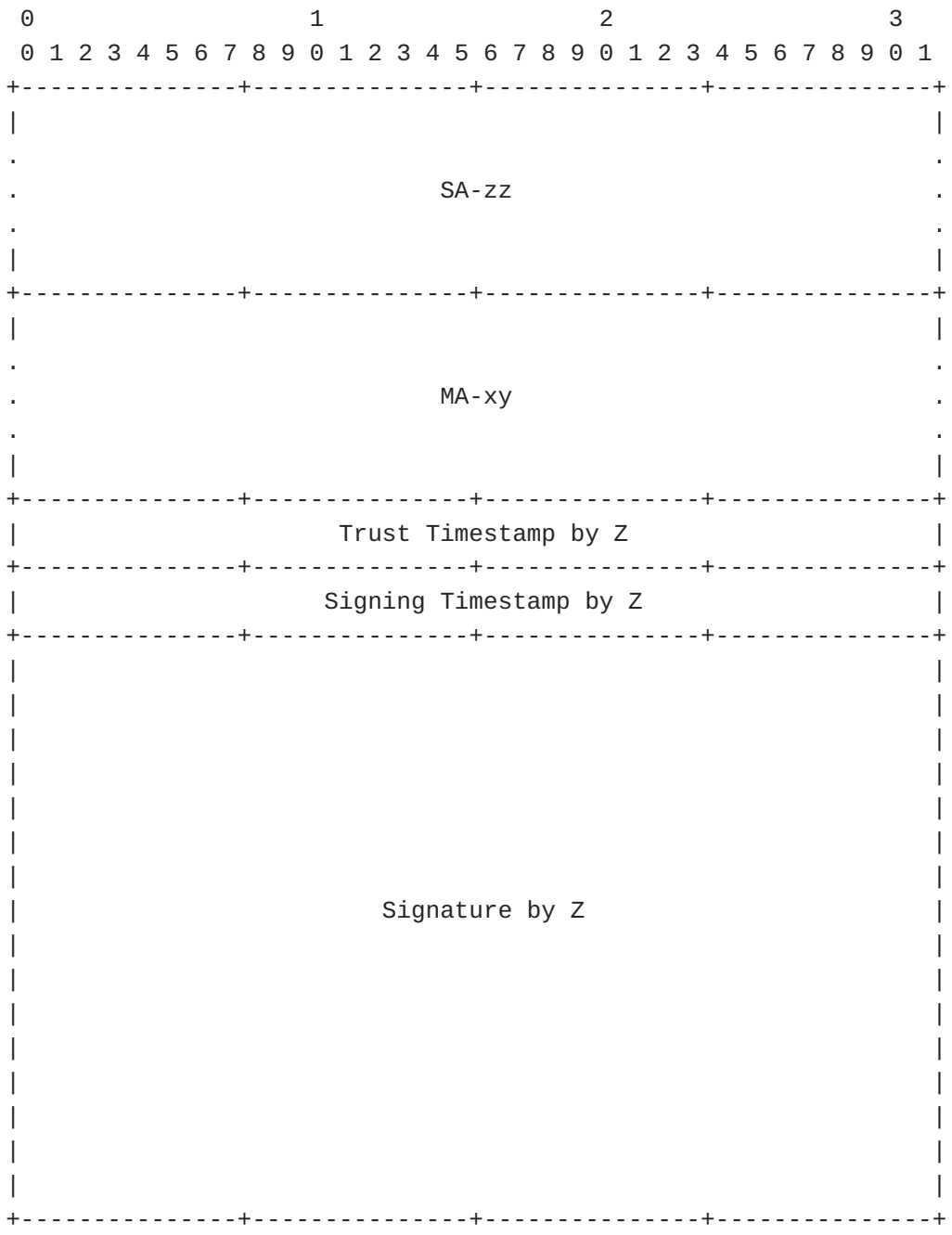
Figure 14: DRIP Concise Certificate

B.9. Link Certificate (LC-zxy)

Length = 300-bytes

Figure 15: DRIP Link Certificate

B.10. Mutual Certificate (MC-zxy)



Length = 576-bytes

Figure 16: DRIP Mutual Certificate

B.11. Example Registration with Attestation

A typical registration of a DET consists of at least three parties; an Aircraft (Party A), an Operator (Party O) and a Registry (Party R). A fourth party (Party Z) can be present as well but is not required.

All parties generate their own SelfAttestation (SA-aa, SA-oo, SA-rr and if present SA-zz) to be used during the registration process.

B.11.1. Operator to Registry

To register the Operator sends to their selected Registry their SelfAttestation (SA-oo).

The Registry, upon acceptance, returns the following items:

1. Attestation (A-ro) - using SA-rr and SA-oo [Mandatory]

2. ConciseAttestation (CA-ro) - using SA-oo [Optional]

3. BroadcastAttestation (BA-ro) - using SA-oo [Optional]

The Operator may choose to generate the following items after successful registration:

1. MutualAttestation (MA-ro) - using A-ro

2. LinkAttestation (LA-ro) - using CA-ro

The Operator may further choose to send A-ro, CA-ro, MA-ro and LA-ro to Party Z to obtain the following:

1. AttestationCertificate (AC-zro) - using SA-zz and A-ro

2. MutualCertificate (MC-zro) - using SA-aa and MA-ro

3. ConciseCertificate (CC-zro) - using CA-ro

4. LinkCertificate (LC-zro) - using LA-ro

B.11.2. Aircraft to Operator

After registering with a Registry the Operator can now start to provision an Aircraft. Extracting SA-aa from the Aircraft they can create the following items:

1. Attestation (A-oa) - using SA-oo and SA-aa [Mandatory]

2. ConciseAttestation (CA-oa) - using SA-aa [Optional]

3. BroadcastAttestation (BA-oa) - using SA-aa [Optional]

A-oa can be returned to the Aircraft to produce the following:

1. MutualAttestation (MA-oa) - using A-oa

2. LinkAttestation (LA-oa) - using CA-oa

B.11.3. Aircraft to Registry

Using the existing A-ro and new A-oa (and if present CA-oa, MA-oa and LA-oa) the Operator can attempt registration with the previous Registry.

The Registry upon acceptance can return the following set of items:

1. BroadcastAttestation (BA-ra) - generated using the embedded SA-aa from A-oa [Mandatory]

2. AttestationCertificate (AC-roa) - using A-oa [Mandatory]

3. MutualCertificate (MC-roa) - using MA-oa [Optional]

4. ConciseCertificate (CC-roa) - using CA-oa [Optional]

5. LinkCertificate (LC-roa) - using LA-oa [Optional]

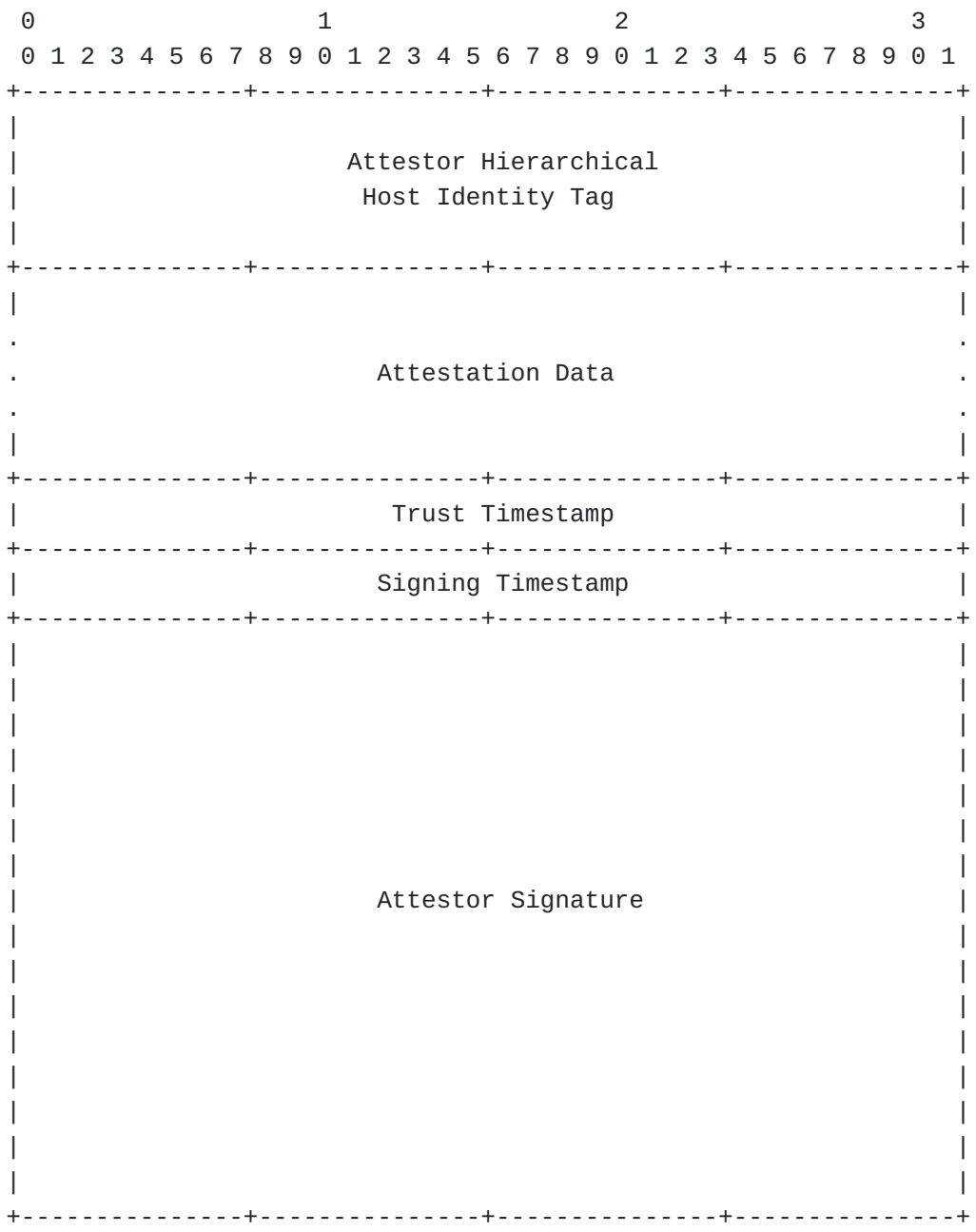
6. BroadcastAttestation's of parent Registries in chain [Recommended]

The BroadcastAttestation's are used during the flight of the Aircraft being sent via DRIP Link Authentication Messages.

Appendix C. DRIP Broadcast Attestation Structure

(Editors Note: move into main document? Between Section 3 and 4 and after FEC section?)

When possible the following format SHOULD be used in the DRIP Authentication Data (Section 4.4.1.2) field under Authentication Type 0x5 or the Authentication Data / Signature of Figure 2.



Attestor Hierarchial Host Identity Tag (16 bytes):
    The Attestors HHIT in byte form (network byte order).

Attestation Data (0 to 112 bytes):
    Opaque attestation data.

Trust Timestamp (4 bytes):
    Timestamp denoting recommended time to trust data to.

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

Attestor Signature (64 bytes):

Signature over preceding fields using the keypair of the Attestor.

Figure 17: DRIP Broadcast Attestation Data Structure

C.1. Attestor Hierarchical Host Identity Tag

The HHIT 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]. Nominally the Aircraft HHIT is used here when signing over dynamic formats such as DRIP Wrapper and DRIP Manifest.

C.2. Attestation Data

This field has a maximum of 112 bytes in length. It is nominally filled with data as defined by the SAM Type being set or other sub-multiplexer in the authentication payload.

C.3. Trust Timestamp

Follows 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.

The offset used against the UNIX 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.

C.4. Signing Timestamp

Follows the format defined in [F3411-19]. That is a UNIX timestamp offset by 01/01/2019 00:00:00.

C.5. Attestor Signature

The signature is generated over all the preceding data. Only information within the Broadcast Attestation Structure (Appendix C) is signed.

Appendix D. Forward Error Correction

(Editors Note: move into main document? Between Section 3 and 4 (preferably becoming Section 4)? or move specifics of FEC (everything below) into its own draft for titled Integrity Protection?)

Remote ID data can be sent across different broadcast link media, all with different characteristics. To enable robustness in Remote ID transmission media a Forward Error Correction capability SHOULD be used.

In cases where FEC is not available below the equivalent of the transport layer (known as Legacy Advertisements) DRIP Authentication REQUIRES that an application level FEC scheme is used. In cases where FEC is available below the equivalent of the transport layer (known as Extended Advertisements) DRIP MUST NOT use any application level FEC and instead SHALL rely on the lower layers FEC functionality.

For current Remote ID the media options are the following:

Legacy Advertisements: Bluetooth 4.X

Extended Advertisements: Wi-Fi NAN, Wi-Fi Beacon, Bluetooth 5.X

(Editors Note: add in self-protecting and more-than-self-protecting options, with their justifications)

D.1. Encoding

D.1.1. Single Page FEC

To generate the parity a simple XOR operation using the previous and current page is used. For Page 0, a 25-byte null pad is used for the previous page. The resulting parity fills the Additional Data field of [F3411-19] with the Additional Data Length field being set to 25.

D.1.2. Multi Page FEC

TODO (Reed Solomon)

(Editors Note: probably need a table to check against ADLs to which parameters of Reed Solomon are being used?) (Editors Note: this is the place to define if we are self-protecting or global-protecting with FEC...another multiplex byte here directly after the ADL?)

D.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 pages 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 Data Length field SHOULD also be performed.

To determine if Single Page FEC or Multi-Page FEC has been used a simple check of the Additional Data Length (ADL) field can be used. If the ADL is equal to 25, then Single Page FEC is present, anything larger signals Multi-Page FEC.

D.2.1. Single Page FEC

Using the same methods as encoding, an XOR operation is used between the previous and current page (a 25-byte null pad is used as the start). The resulting 25-bytes should be the missing page.

D.2.2. Multi Page FEC

TODO (Reed Solomon)

(Editors Note: probably need a table to check against ADLs to which parameters of Reed Solomon are being used?) (Editors Note: this is the place to define if we are self-protecting or global-protecting with FEC...another multiplex byte here directly after the ADL?)

D.3. FEC Limitations

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.

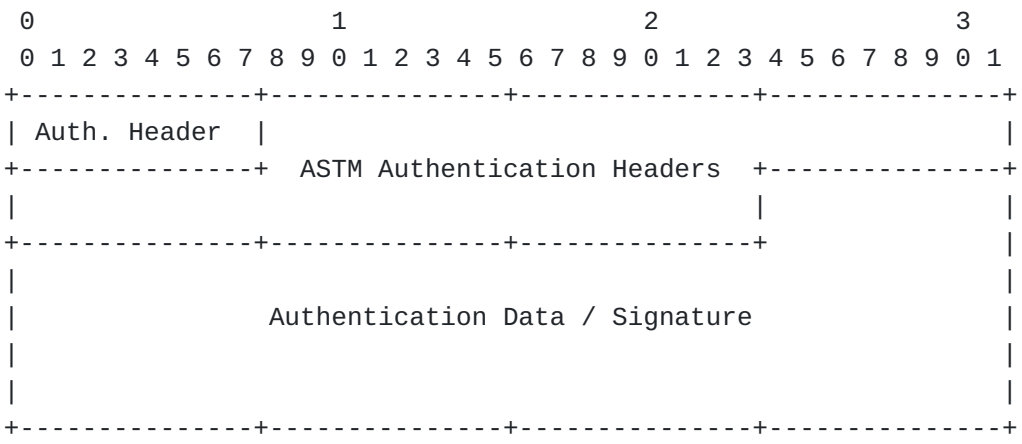
(Editors Note: Is this valid anymore, for XOR yes but for multi-page FEC?)

Appendix E. Example Authentication Messages

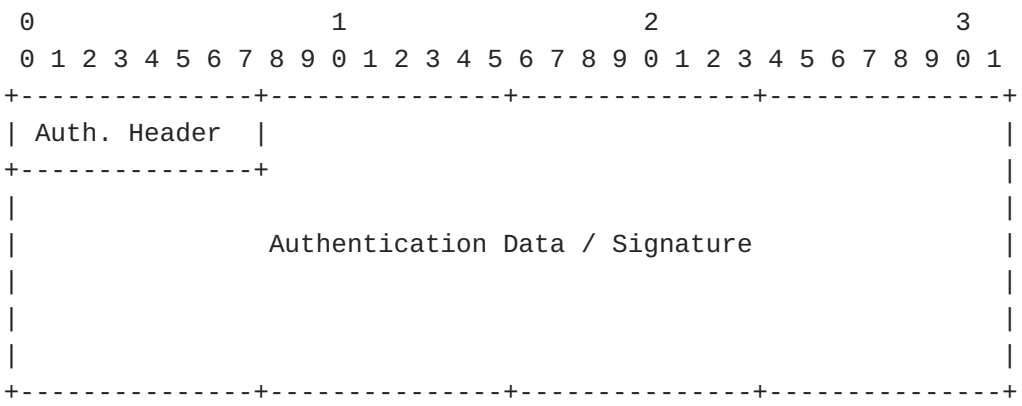
E.1. Authentication Data Only

This is an example of an Authentication Message with 52-bytes of Authentication Data.

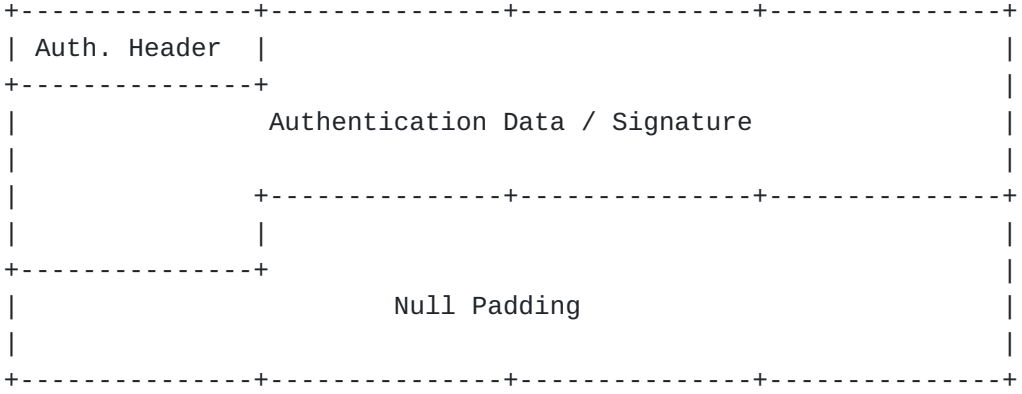
Page 0:



Page 1:



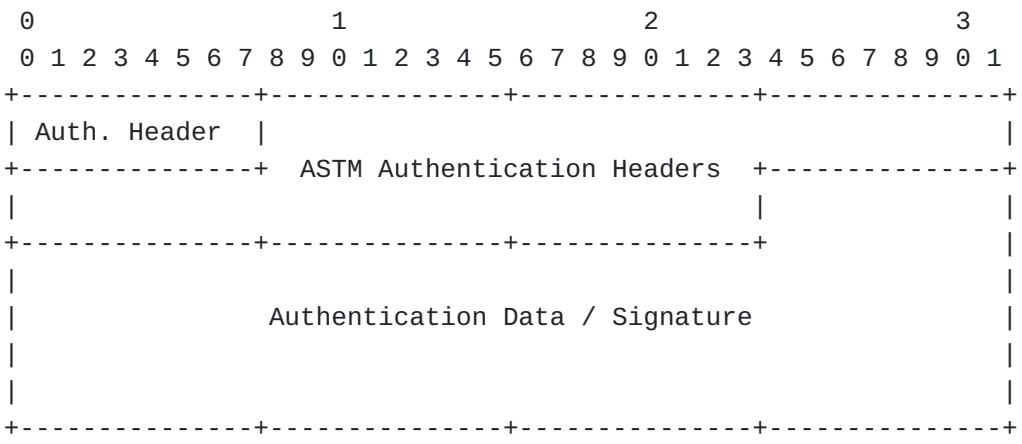
Page 2:



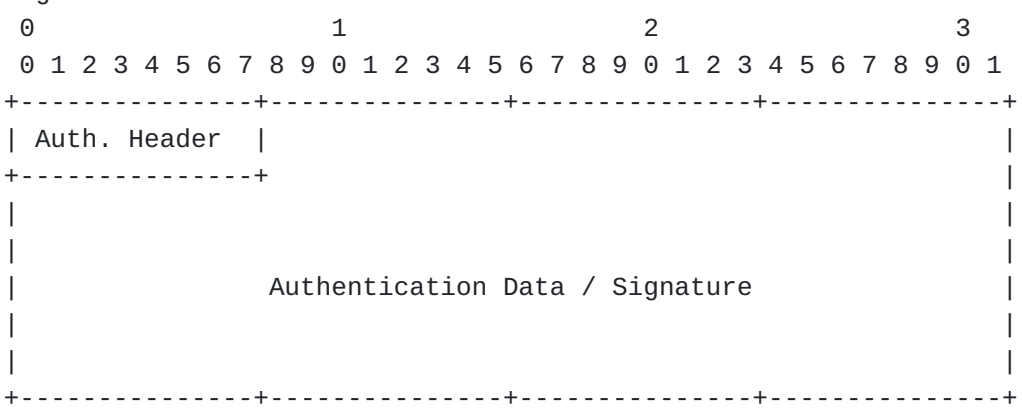
E.2. Authentication Data & Additional Data

This example has 52-bytes of Authentication Data and 20-bytes of Additional Data.

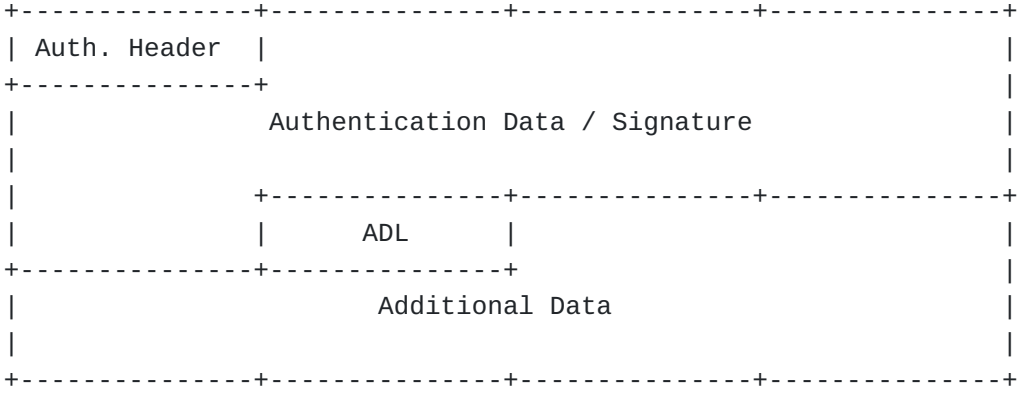
Page 0:



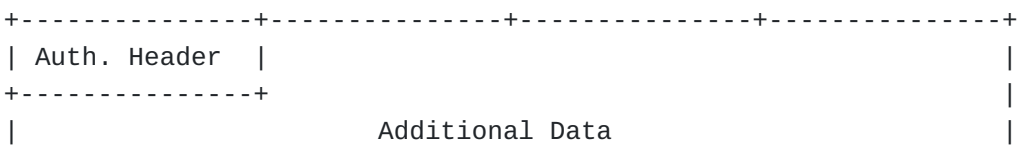
Page 1:

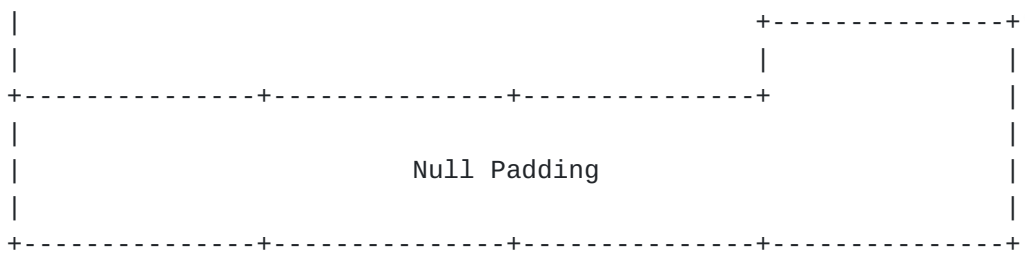


Page 2:



Page 3:

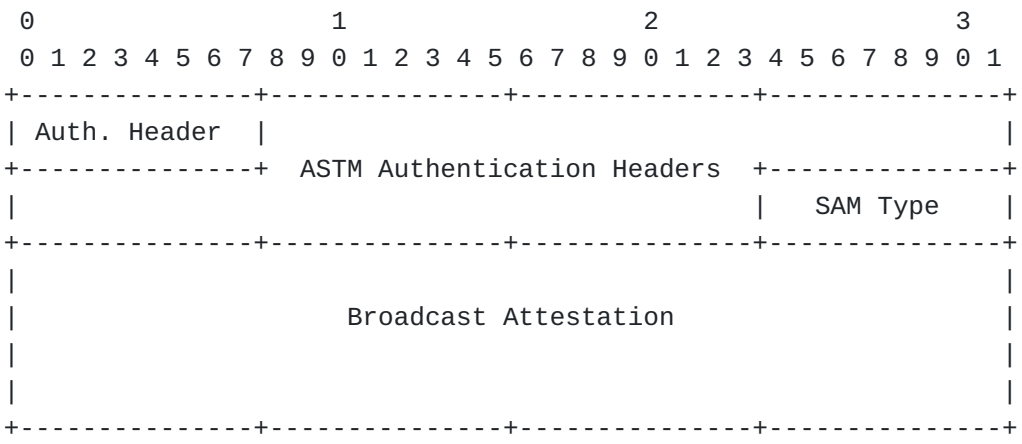



Null Padding

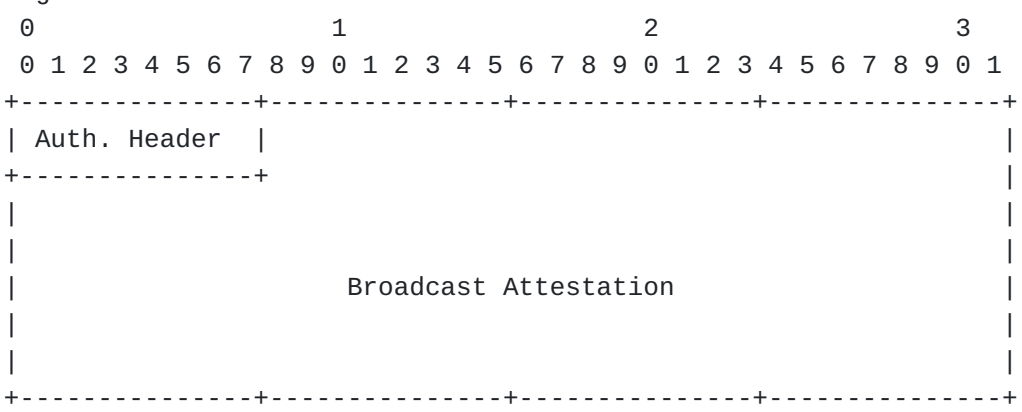
E.3. DRIP Link Example

This DRIP Link example includes FEC for a single page.

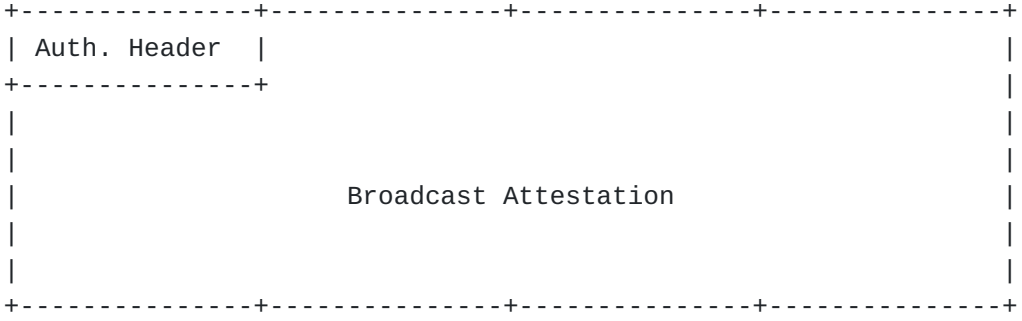
Page 0:



Page 1:

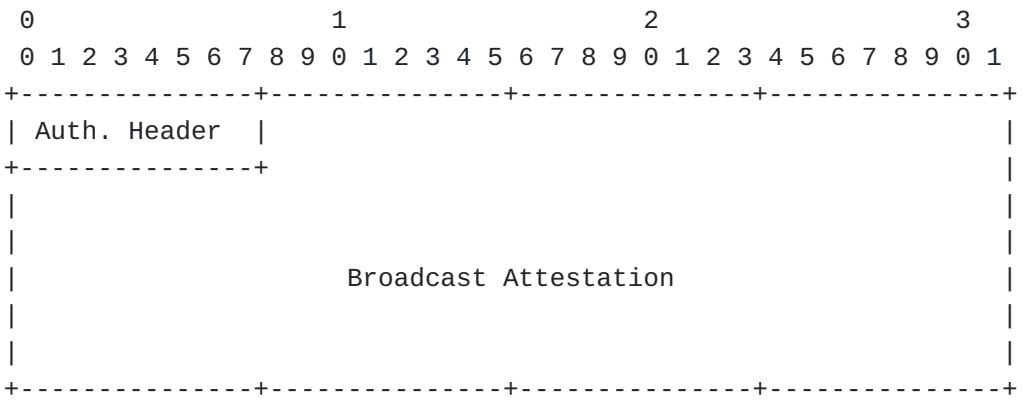


Page 2:

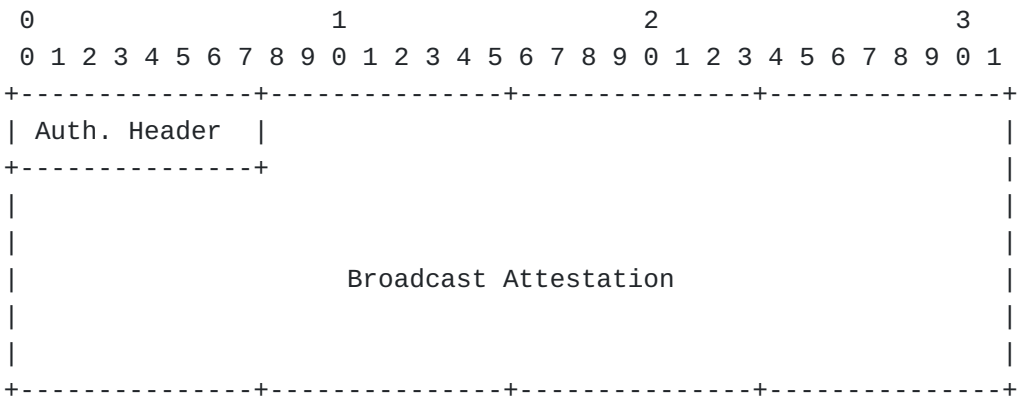


Page 3:

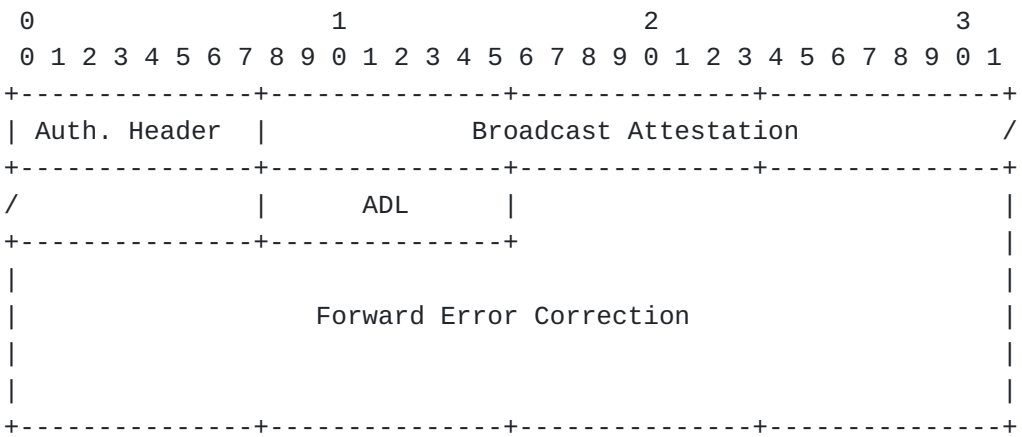
Page 4:



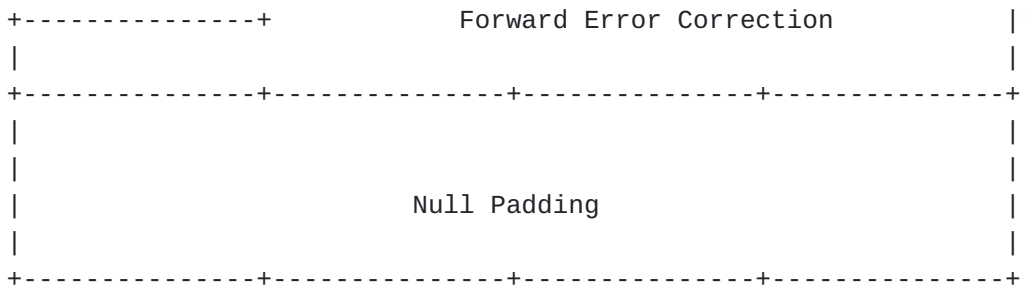
Page 5:



Page 6:



Page 7:


Forward Error Correction
Null Padding

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