DRIP Implementation Drafts

draft-wiethuechter-drip-auth-03

draft-wiethuechter-drip-identity-claims-00

Adam Wiethuechter

DRIP WG – AUG20 Interim; 26 AUG 2020

From the DRIP Charter

DRIP's goal is to specify how RID can be made trustworthy and available in both Internet and local-only connected scenarios

The DRIP HHIT Solution

- Use the HHIT as the UAS ID
  - See draft-moskowitz-drip-uas for details
- Use the small signature size of EdDSA25519
  - Easily fits in ASTM Authentication Message
    - UA HHIT (16) + Timestamp (4) + Signature (64) = 84 bytes out of 109 bytes
    - 25 bytes left for data to be signed
  - GEN 2: Provable Binding (when using HHIT as UAS ID)
- Increase Auth. Page limit from 5 to 10
  - We have approached ASTM and they have been receptive to this change
- Add Forward Error Correction to help loss of pages in Bluetooth 4.X
- Send short Certificate via Authentication Message making RID trustworthy in local-only scenarios
  - GEN 1: Provable Ownership; GEN 3: Provable Registration

DRIP Authentication Framing

General Frame, Wrapper Frame

General Frame

- Reed Solomon FEC always fills last page
  - Taken over all pages (inc. headers)
  - SHOULD on Bluetooth 4
  - SHOULD NOT on Bluetooth 5
- 223 bytes of data w/o FEC
- 200 bytes of data w/FEC

DRIP Header

- Independent FEC flag
  - Each DRIP AuthType specifies if using FEC
- 7 bit space broken into 5 areas
  - Half (8) of Wrapped Messages defined
  - One (1) Certificate defined
- 128 possible DRIP AuthTypes
  - 9 total currently defined

```
DRIP Header (1 byte):
      7     6     5     4     3     2     1     0
   +-----+-----+-----+-----+-----+-----+-----+-----+
   | FEC |              DRIP AuthType              |
   +-----+-----+-----+-----+-----+-----+-----+-----+

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

```
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)
```

FEC & Bluetooth

- Bluetooth (both 4 and 5) have a 3 byte CRC in every frame
  - Full frame is dropped if CRC check fails within Bluetooth stack
  - No signal to upper layers that a frame is being dropped
- To RID applications, we missed;
  - Under BT4 a full message or *Authentication page*
  - Under BT5 a full Message Pack

How does this help us?

- Authentication pages are numbered (part of the Auth. Header already defined by ASTM) so we know which pages are missing in a set
  - sets are defined using the AD Counter
- Reed Solomon can correct 23 bytes of error when we know positions of data lost (known as *erasures*) – which we do!
  - So if we rebuild frames filling in known header bytes (Message Type, ASTM Version, Authentication Type and Page Number) we can correct for 23 bytes which is missing page data

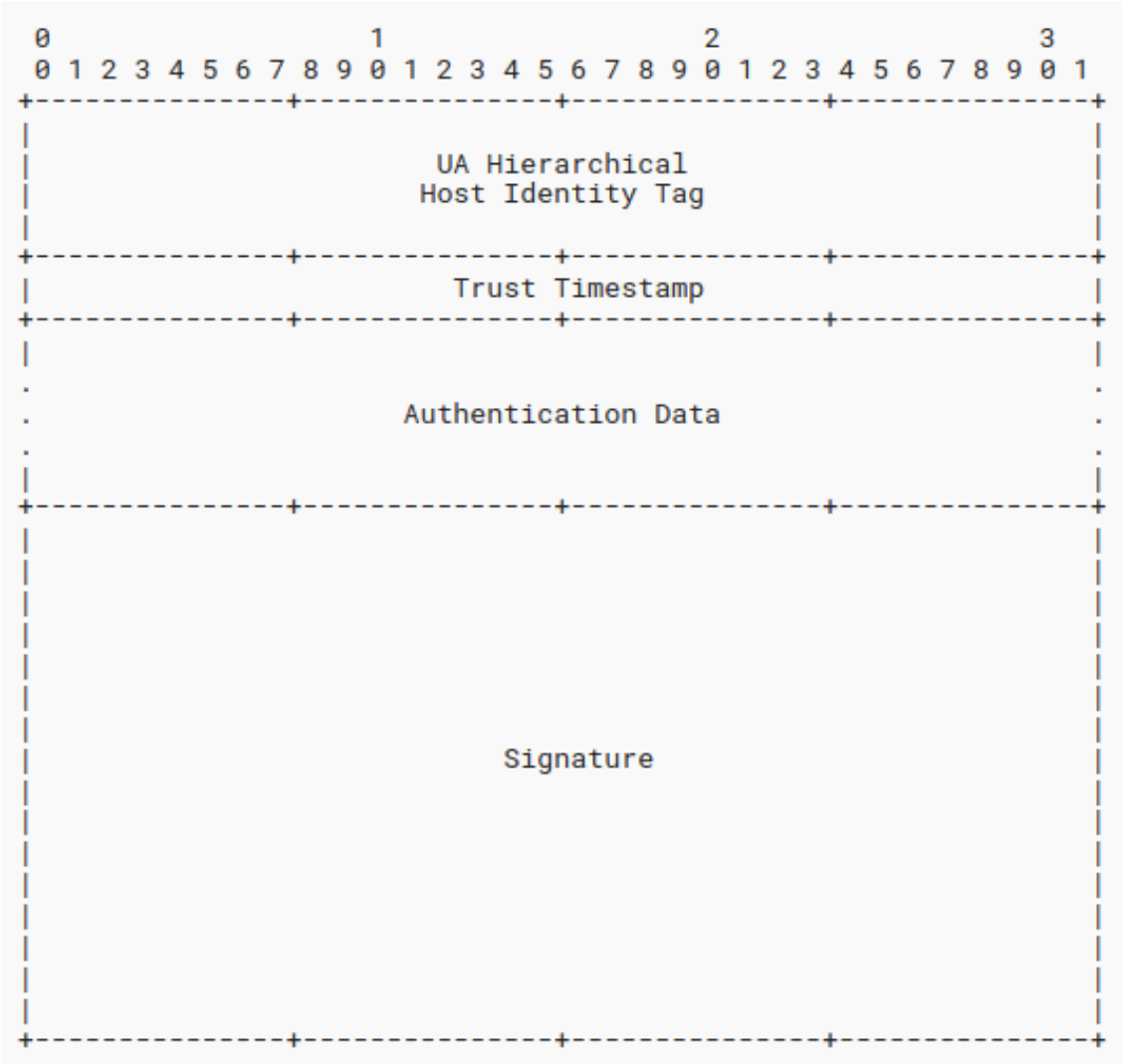
End results…

- For Bluetooth 4, FEC gives us an advantage of recovery if any single page is lost in transmission
  - If any more are lost recovery is impossible but if that happens probably more issues going on anyways

- For Bluetooth 5, FEC is useless as it already has FEC at the frame level before CRC check
  - Only with LE Coded PHY, which is what is specified by ASTM
- Also for Bluetooth 5, FEC is useless as per ASTM the Message Pack must be used
  - This uses the 255 byte extended Bluetooth 5 payload to fit multiple ASTM Messages in single frame
  - So if we lose a Bluetooth 5 frame we are already losing anyways as a full Authentication Message was together, not physically paged like Bluetooth 4

Wrapper Frame

- Fits inside General Frames DRIP Auth. Data
- Authentication Data
  - 116 bytes with FEC
  - 139 bytes w/o FEC
- Signature computed over all preceding data fields in Wrapper Frame
  - Avoid DRIP Header as can change (FEC bit) after signing



[Trust] Timestamp Details

- Different types of timestamp in ecosystem:
  - ASTM Authentication Message [4 bytes]
    - Offset from 01/01/2019 00:00:00
  - 32 bit unsigned UNIX [4 bytes]
  - UTM (ISO8601) [? bytes]
- Discussion on list concluded: use ASTM style for everything
  - No need for anything before 2019-01-01, so ASTM way of doing things is reasonable

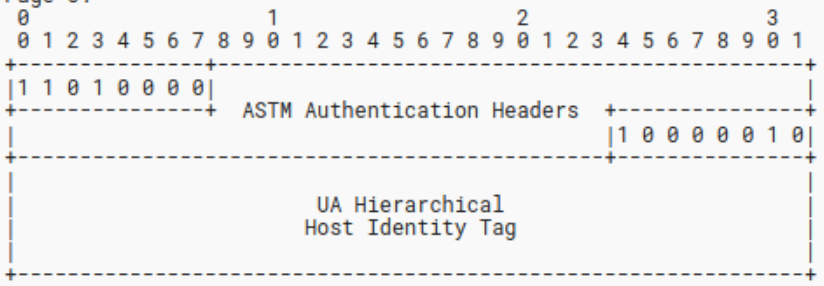
Bluetooth 4.X Auth. Formats

Wrapped ASTM Message(s), Certificate, Manifest(s)

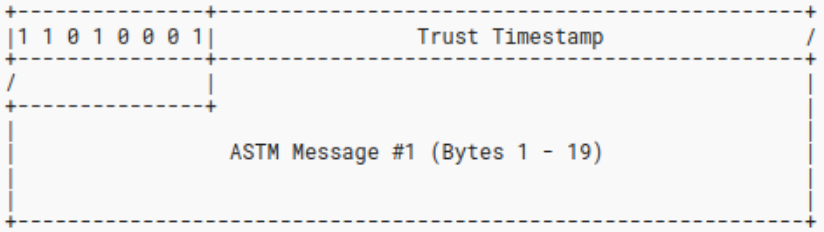
1-5 Wrapped ASTM Message(s)

- DRIP AuthTypes 1-5
  - AuthType signals number of messages being wrapped
- Wrapper Frame Auth. Data filled with ASTM Messages
  - Messages must be in Message Type order
- Special Case: 5 Wrapped Messages
  - Acts as a pseudo-ASTM Message Pack (Type 0xF) over Bluetooth 4
  - FEC MUST be disabled to fit all messages
  - Can fit all ASTM Messages excluding an Auth. Message

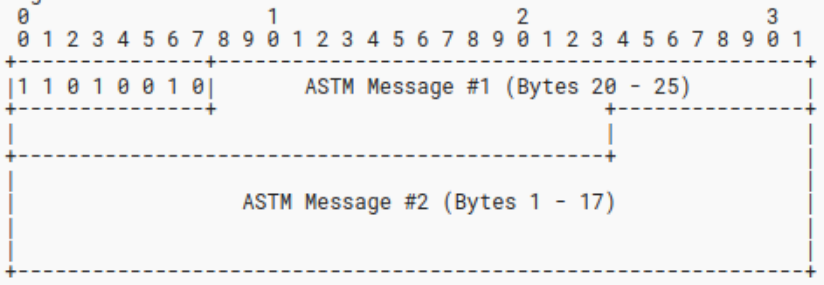
Page 0:



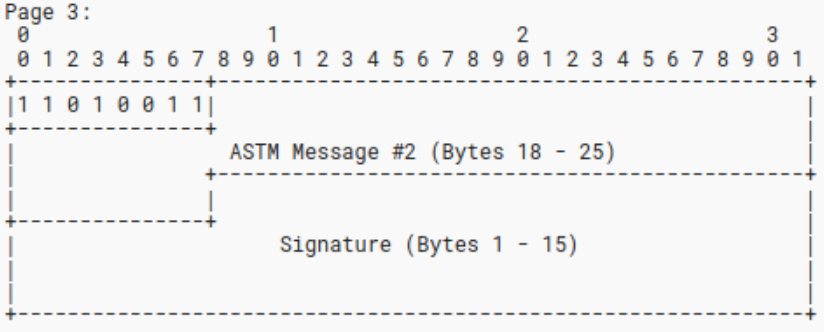
Page 1:



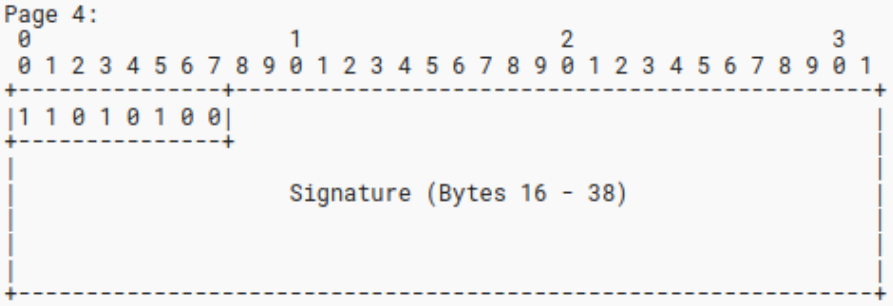
Page 2:



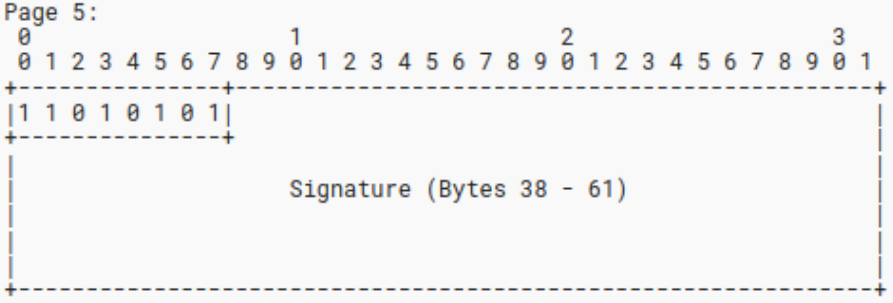
Page 3:



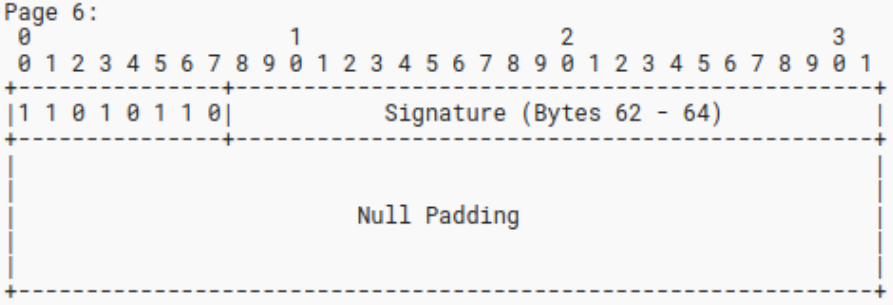
Page 4:



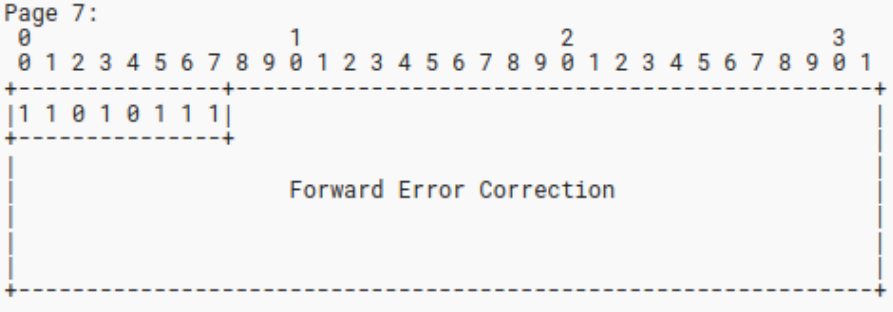
Page 5:



Page 6:



Page 7:



Manifests

- DRIP AuthTypes 6, 7
- Wrapper Frame Auth. Data filled with hashes
  - Hashes are of previous non-paged messages sent
- Two special hashes for pseudo-blockchain
  - Links manifests together
  - Hash of previous manifest
  - Hash of current manifest
    - Order of operations?
- Two variants based on hash length; 8 bytes and 4 bytes
  - 27 hashes with 4 bytes, 12 hashes with 8 bytes
  - Uses same hash algorithm as HHIT (in UAS RID this is cSHAKE128)
    - Can use OGA ID of HHIT to signal different hashing methods

Certificate: Registry on Aircraft (Cra)

- DRIP AuthType 16
- General Frame DRIP Auth. Data filled with Cra
  - Exactly 200 bytes in length
  - Binding between entities, asserting trust
  - Contains HI of UA; instant verification of UA
  - Registry HHIT used for lookup on local cached Registry list
    - On Observer device, only ones trusted by User
- See draft-wiethuechter-drip-identity-claims for details

Bluetooth 5.X Auth. Formats

0 Wrapped ASTM Message(s), Certificate

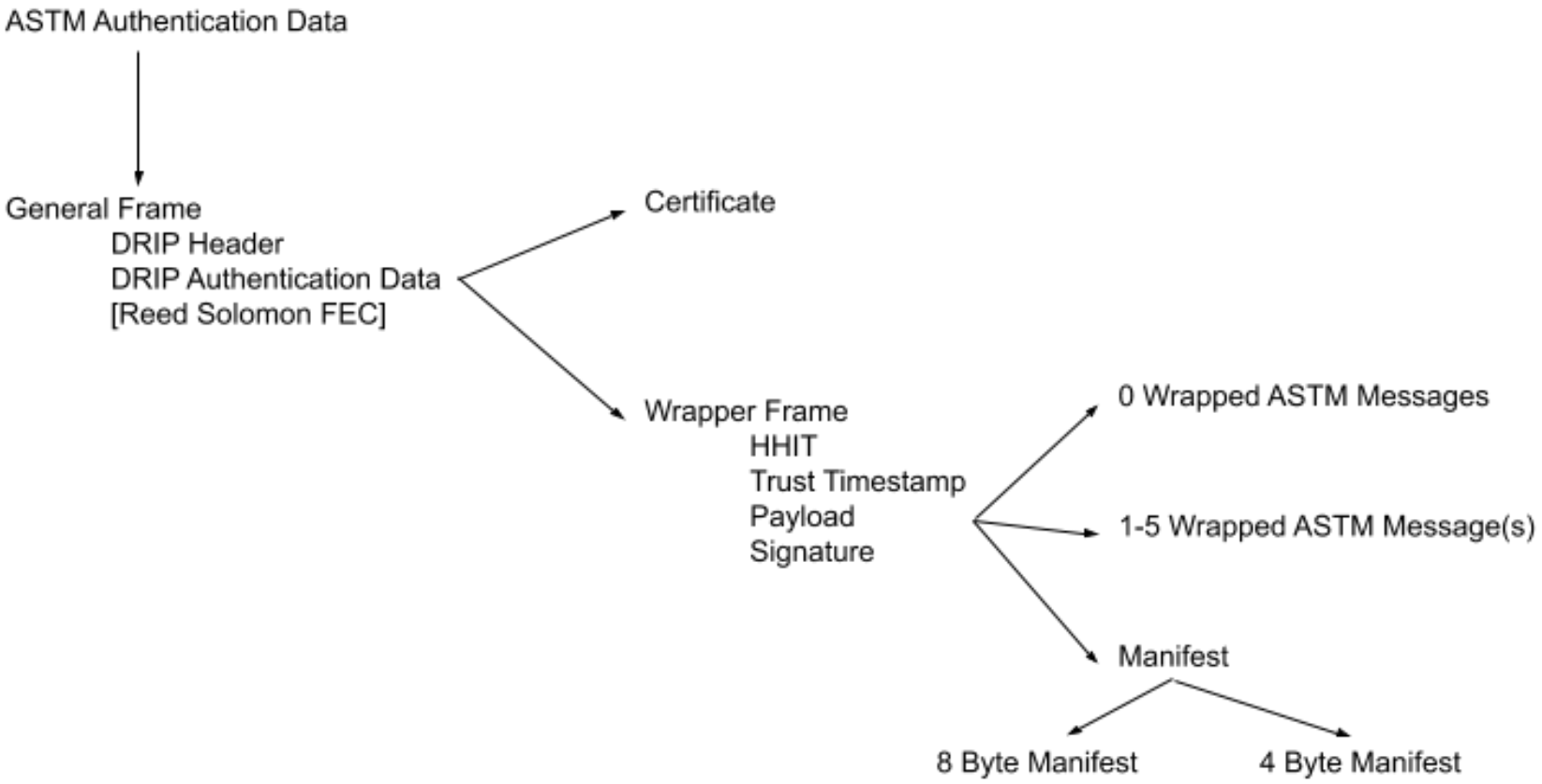
Certificate: Registry on Aircraft (Cra)

- DRIP AuthType 16
- General Frame DRIP Auth. Data filled with Cra
  - See draft-wiethuechter-drip-identity-claims
- Last 25 bytes of Message Pack can be filled with another ASTM Message
  - Suggested to use Location Message

0 Wrapped ASTM Message(s)

- DRIP AuthType 0
- Special case of Wrapped ASTM Message(s) format
  - Only used for Message Pack under Bluetooth 5.X
- Wrapper Frame Auth. Data *virtually* filled with ASTM Messages in Message Pack
  - Messages must be in Message Type order

DRIP AuthType Tree



Identity Claims/Certificates

Building a trustworthy chain for Broadcast RID

Overview

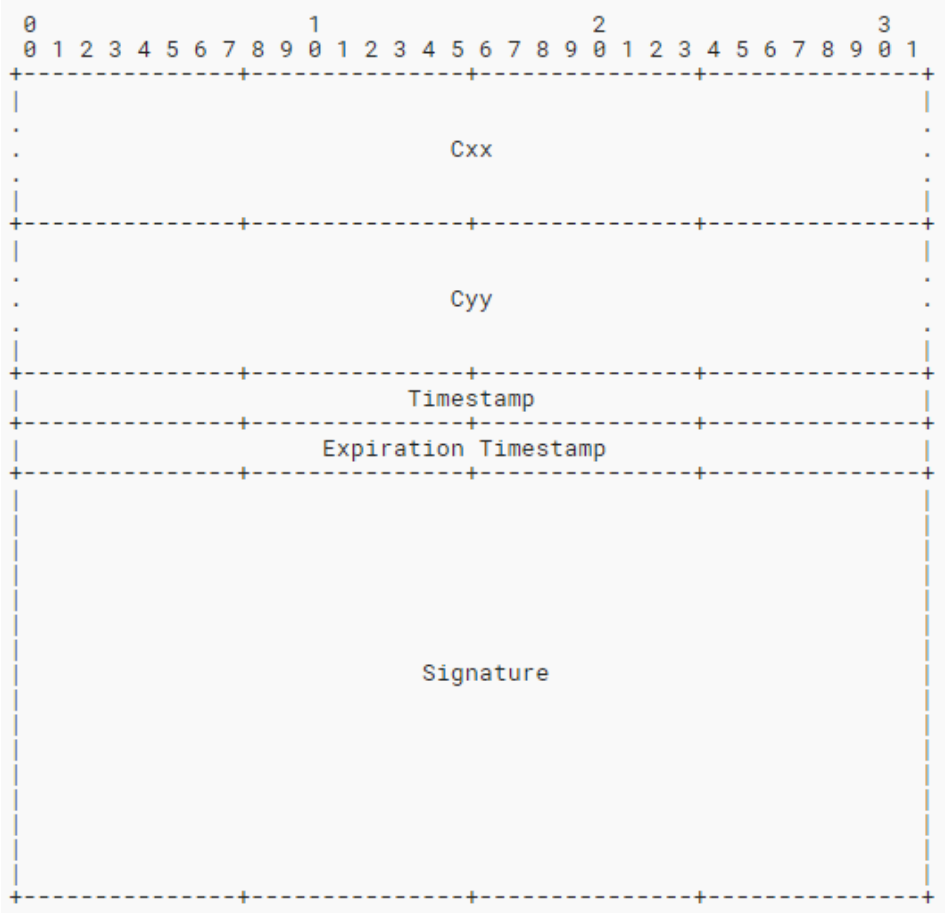
- Claim vs Certificate
  - Claim was chosen initially as “certificate” has a pre-establish connotation
  - Legal and technology baggage with the term and want to avoid confusion
  - This decision is in flux and we would like feedback on it! (we are now back on Certificate)
- Special to the UAS ecosystem for Remote ID
  - Asserts bindings between entities and objects
  - Created during provisioning of UA/Operator/Registry

Form Cxx

- Self-signed unverified claim
- Used to assert binding of HHIT/HI to a given entity (x)
  - Contains: HHIT, HI, Expiration Timestamp, Signature
  - 116 bytes in length
- Three specific entities:
  - Aircraft on Aircraft (Caa)
  - Operator on Operator (Coo)
  - Registry on Registry (Crr)
- Used in other forms

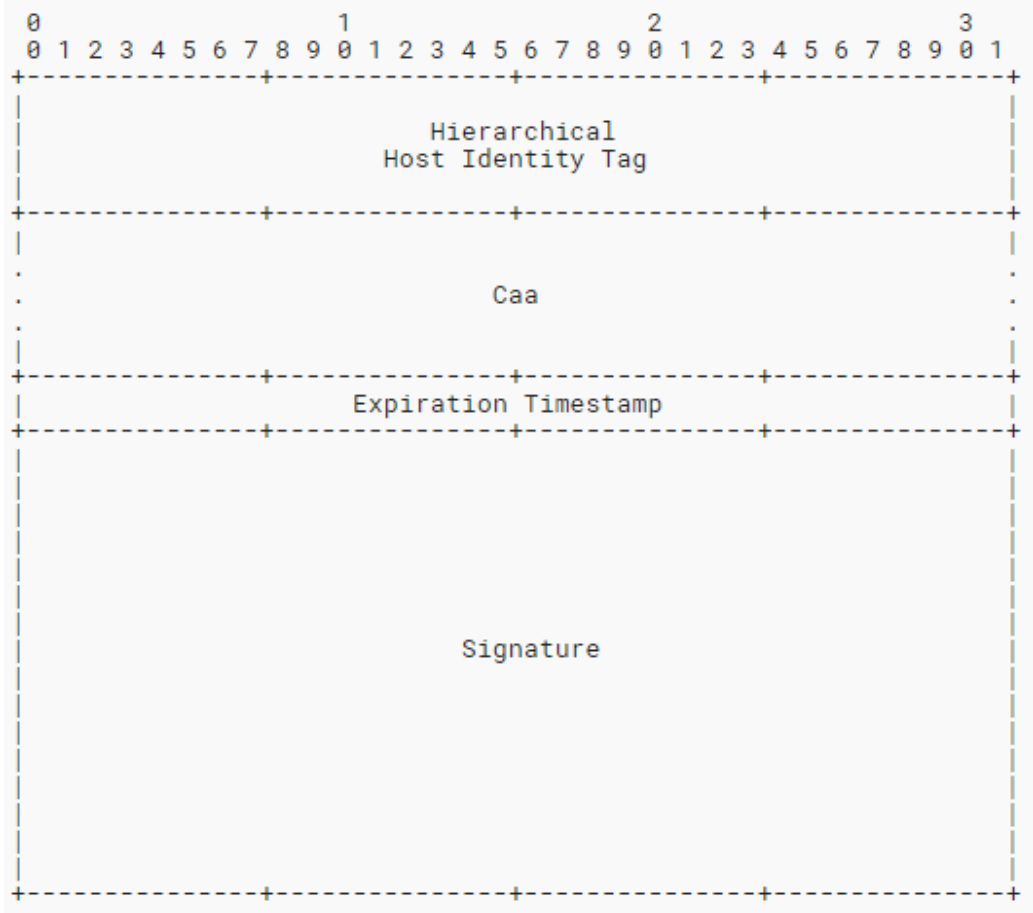
Form Cxy

- Asserts binding between two entities (x and y)
  - Generally ‘x’ is an entity attesting ‘y’s claim (or adding a relationship)
  - Contains: Cxx, Cyy, Timestamp, Expiration Timestamp, Signature
  - 304/608 bytes in length
- 3 specific implementations of this form:
  - Registry on Operator (Cro)
  - Operator on Aircraft (Coa)
  - Registry on Operator on Aircraft (Croa)

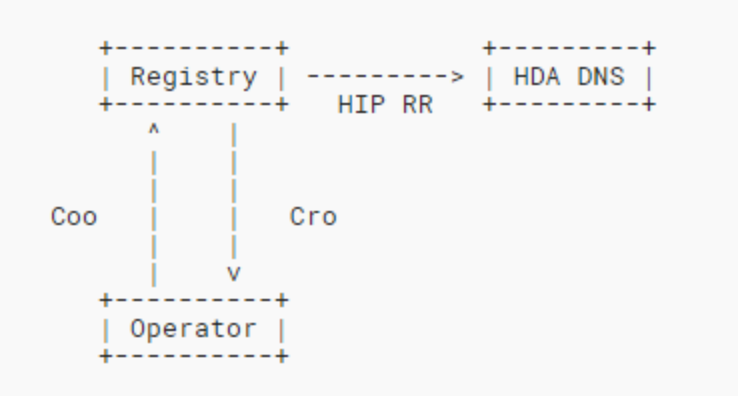


Certificate: Registry on Aircraft

- Special as it is used in authentication messages of Broadcast RID
  - Contains: HHIT of Registry, Caa, Expiration Timestamp, Signature
  - 200 bytes long
- Asserts the binding between a Registry and Aircraft

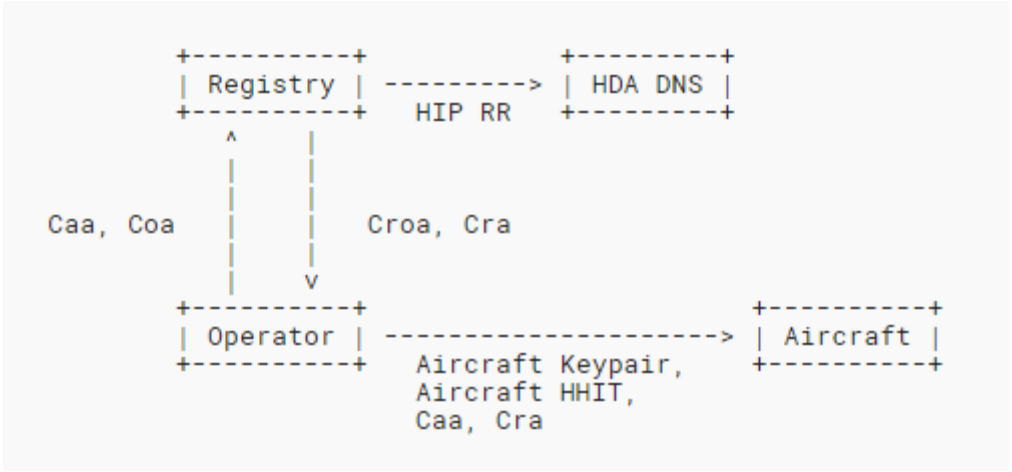


Provisioning: Operator



- Keypair generation
- HHIT derived from HI (public half of keypair)
  - Select Registry and use RRA/HDA to format valid HHIT
- Create Coo, send to Registry
- Registry perform verification check and adds HHIT/HI to DNS in the form of HIP RR
  - Verification check MUST include looking for HHIT collisions in current database of Registered HHITs
- Registry if successful, creates Cro and sends it back to Operator
- Registry if failed, sends error back asking to start over

Provisioning: Aircraft



- Keypair generation
- HHIT derived from HI (public half of keypair)
  - Registry selected and RRA/HDA used to format valid HHIT
- Create Caa, create Coa
- Send Caa, Coa to Registry
- Registry perform verification check and adds HHIT/HI to DNS in the form of HIP RR
  - Verification check MUST include looking for HHIT collisions in current database of Registered HHITs
- Registry if successful, creates Croa, Cra and sends them back
- Registry if failed, sends error back asking to start over

Implementation Progess

AX Enterprize Implementation

- ASTM F3411-19 for Broadcast RID
- Trustworthy Multipurpose Remote ID (TMRID)
  - Specific implementation of DRIP drafts at AX Enterprize
  - Supporting drafts
    - auth-00 (ugrading to 03 soon™)
    - identity-claims-00
    - uas-rid-06
- HHIT Registry
  - Identity-claims-00
  - API endpoint to provision aircraft and store HHIT/HIs in BIND9 zone dynamically
- Been flying and demoing since June 2020

https://xkcd.com/364/

Discussion

Questions, Comments, Concerns?

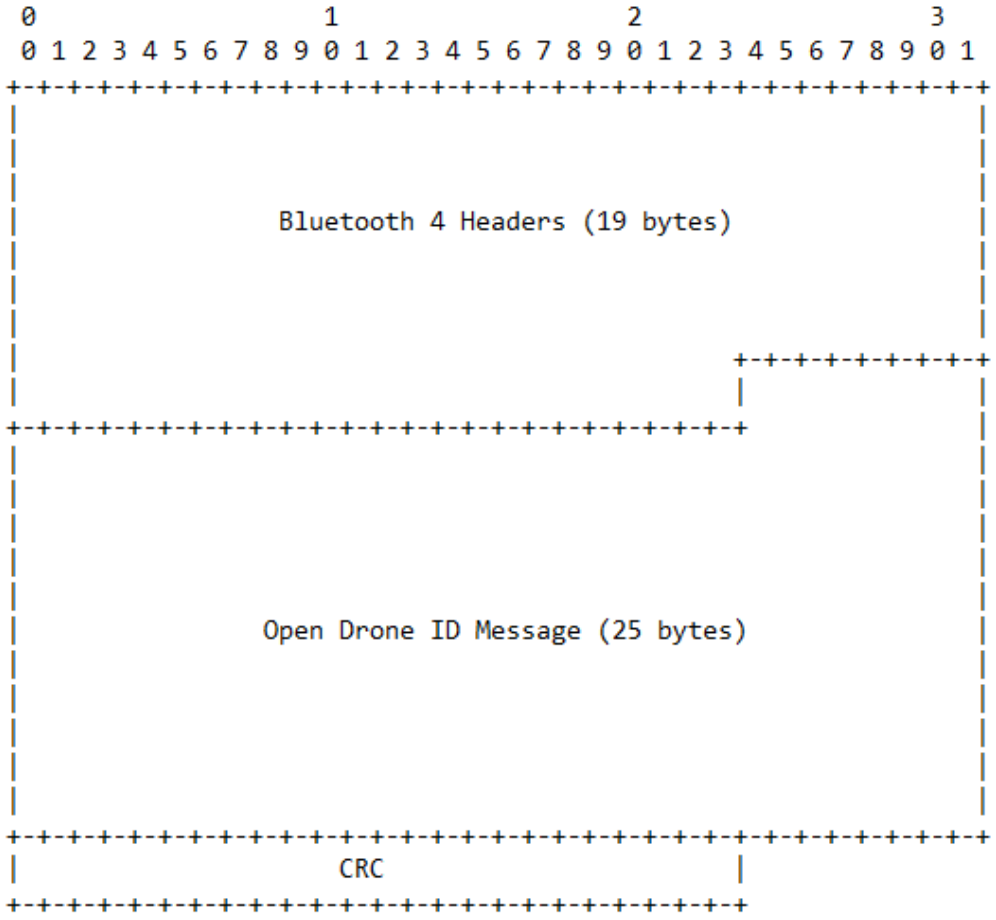
Backup Slides

Background & Problem

- ASTM F3411-19 Broadcast RID
  - Disjointed information delivery
    - Identity information of UA sent in Basic ID
    - Position information of UA sent in Location
      - But no ID in the Location Message
    - Authentication information of UA sent in Auth
    - All of these are sent and received separately (under Bluetooth 4.X)!
  - Fragmented data across Authentication Message pages
- Overall a lack of trust in Broadcast messages
  - Especially in Bluetooth 4.X

Bluetooth Background

- Why so small?
  - Bluetooth 4 legacy frames only give 25 bytes to play with (after Bluetooth headers)
  - 1 byte is for a main header in ASTM format that is always present – now only 24 bytes of data to work with per frame/page



ASTM Authentication

- ASTM F3411-19 “Standard Specification for Remote ID and Tracking”
- Authentication Message
  - 5 pages long with a 109 byte max payload (17 + 23 * 4)
  - Designed to authenticate Message Packs (of up to 5 messages in Bluetooth 5.X frame)