DRIP
Internet-Draft
Intended status: Standards Track
Expires: 24 October 2025

R. Moskowitz
HTT Consulting
S. Card
AX Enterprize
A. Gurtov
Link旦ping University

22 April 2025

Aircraft to Anything AdHoc Broadcasts and Session draft-moskowitz-drip-a2x-adhoc-session-06

Abstract

Aircraft-to-anything (A2X) communications are often single broadcast messages that to be trustable, need to be signed with expensive (in cpu and payload size) asymmetric cryptography. There are also frequent cases of extended exchanges between two devices where trust can be maintained via a lower cost symmetric key protect flow.

This document shows both how to secure A2X broadcast messages with DRIP Entity Tags (DET) and DRIP Endorsement objects and to leverage these to create an AdHoc session key for just such a communication flow.

There is also provision for DETs within X.509 certificates, encoded in c509, as an alternative DET trust model.

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 October 2025.

Table of Contents

1. Introduction

Aircraft-to-anything (A2X) communications are currently an open area of debate and research. There are questions as to whether air-to-ground-to-air or air-to-air will prevail as the driving force and thus the impetus for solutions. One real need is in air-to-air Detect And Avoid (DAA). There are current DAA approaches for commercial/general aviation over Automatic Dependent Surveillance寰釘roadcast (ADS-B); this is unlikely to scale to the needs of Unmanned Aircraft (UA). UA-to-UA and UA - to - General Aviation will drive a different solution.

ADS-B cannot, directly, be secured. The data framing is too constrained, and the Manifest approach in [RFC9575] is also not possible within the ADS-B constraints. This is considered an acceptable risk, as these large airframes can support radar systems to validate presense reported by ADS-B along with crew visual sighting. Neither of these are options in UA-to-anything and thus a trustable messaging environment is needed.

Although all aspects of this document is applicable to all aircraft, the language will be Uncrewed Aircraft (UA) slanted. This is to avoid any confusion within the body of the document about what type of aircraft is under discussion.

A prevailing current approach for UA-to-anything messaging is to leverage the Broadcast Remote ID (B-RID) messages, as they provide situational awareness in the Vector/Location ASTM messages [F3411-22a]. But this message is not adequate by itself for situational awareness as it does not identify the UA (other than the wireless media MAC address). Thus although it is possible for UAs act on a set of B-RID messages, this document will define specific, singular, messages that are directly actionable by listening UAs.

The messages defined in this document leverage the DRIP Entity Tags (DET) [RFC9374] and its underlying Ed25519 keypair. It also relies on DET endorsement hierarchy for trust in the DETs.

ECDSA as in the ICAO defined International Aviation Common (IAC) PKI can be used in DETs (DET Suite ID of 3). With point-compression, the P-256 keys would have Host Idenities (HI) 1 byte larger than Ed25519 (i.e. 33 bytes). This increase may still be usable. It would also facilitate DAA between civil/general aviation and UAs.

1.1. X.509 Certificate in place of DET Endorsements

The DRIP DET Public Key Infrastructure, Section 4 of [drip-dki] defines two X.509 certificates profiles, a "Lite" and a "Full" profile to "shadow" the DET Endorsements. Even the Lite X.509 certificates are much larger than the DET Endorsements (e.g. 256 bytes for Lite and 286 bytes for Full). This may be too large for some transmission media.

There is an alternative C509 encoding for these X.509 certificates per [C509-Certificates]. These C509 certificates in initial testing are 172 bytes and 192 bytes respectively. This is still larger than the DET Endorsements, but may be small enough.

Author’s Note: More content is needed for those that wish to implement c509.

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. Notation

|
: Signifies concatenation of information (e.g., X | Y is the concatenation of X with Y).

Ltrunc (x, K)
: Denotes the lowest order K bits of the input x.

2.3. Definitions

This document uses the terms defined in Section 2.2 of Drip Requirements and Terminology [RFC9153] and in Section 2 of Drip Architecture [RFC9434]. The following new terms are used in the document:

A2X
: Communications from an aircraft to any other device. Be it another aircraft or some ground equipment.

cSHAKE customizable SHAKE function (cSHAKE) [NIST.SP.800-185]:
Extends the SHAKE scheme [NIST.FIPS.202] to allow users to customize their use of the SHAKE function.

KECCAK Message Authentication Code (KMAC) [NIST.SP.800-185]:
A Pseudo Random Function (PRF) and keyed hash function based on KECCAK.

3. Broadcast A2X messaging

The basic view of Aircraft-to-Anything (A2X) communications is broadcast based. What is there within radio range of a UA? In most situations, the UA has no apriori knowledge of other systems around it. Thus the UA would just broadcast any information of general knowledge, like "Hello, I am me and I am here. Oh, and I am out of fuel and crashing."

Thus the first concern is to be able to create a single, trusted message from the UA for all devices around it. The following sections detail the pieces of such a trusted message.

3.1. The Compressed DRIP HDA Endorsement of UA DET

The DRIP HDA Endorsement of UA DET is a DRIP Link authentication message as defined in Section 4.2 of [RFC9575]. It is the primary trust proof of UA DETs. This object is 136 bytes, but in the specific context where the UA DET has the same first 64 bits as the HDA DET (typical case), the 16 byte UA DET can be derived from the HDA DET and the UA HI, compressing the object to 120 bytes.

Bytes	Name	Explanation
4	VNB Timestamp	Current time at signing, set by HDA
4	VNA Timestamp	Timestamp denoting recommended time to trust Endorsement, set by HDA
32	HI of UA	Host Identity of UA
16	DET of HDA	DRIP Entity Tag of HDA
64	Signature by HDA	Signature over preceding fields using the keypair of the HDA DET

Table 1: 120-Byte Compressed HDA on UA Endorsement

3.2. Full UA Signed Evidence of the A2X message

The UA can now fully endorse a A2X message by signing it, along with the HDA Compressed Endorsement of the UA (Section 3.1). A Message ID field is needed to distinguish all the multiple messages for this datagram. The message also needs a Until timestamp for a total of 189 bytes plus n bytes of the actual message:

Bytes	Name	Explanation
4	VNA Timestamp	Timestamp denoting recommended time to trust Evidence
1	Message ID	A2X Message ID Number
n	A2X Message	Actual A2X Message
120	HDA/UA Endorsement	Compressed HDA on UA Endorsement
64	Signature by UA	Signature over preceding fields using the keypair of the UA DET

Table 2: 189+n Byte Full UA Signed A2X message

3.3. Compressed UA Signed Evidence of the A2X message

The UA also has the option to send a A2X message without the HDA Endorsement (Section 3.1). These shorter messages can be alternated with the full messages, based on an assumption that the other parties will receive the full endorsed message either prior or after this shorter message format. Or a block of these shorter messages could be sent based on other assumptions. This message is 85 bytes plus n bytes of the actual message, with the average overhead of the 2 messages (1:1 transmission) is 137 bytes:

Bytes	Name	Explanation
4	VNA Timestamp	Timestamp denoting recommended time to trust Evidence
1	Message ID	A2X Message ID Number
n	A2X Message	Actual A2X Message
16	DET of UA	DRIP Entity Tag of UA
64	Signature by UA	Signature over preceding fields using the keypair of the UA DET

Table 3: 85+n Byte Compressed UA Signed A2X message

3.4. IPv6 datagram for A2X message via SCHC

All the pieces are present in these messages for a SCHC [RFC8724] process to expand them to multicast addressed IPv6 datagrams. The destination IPv6 address would be ff02::1 (all nodes), the source address is the UA DET. The protocol would be UDP with the port numbers are still TBD and completely up to the receiving UA.

Thus the A2X application would just be an IPv6 application and A2X could be this broadcast method or a datagram that actually was routed over an IPv6 network. There is no transmission cost to this, just the SCHC mechanism. It is completely up to the receiving UA and the sending UA needs not be concerned if this is how its messages are processed.

4. Using full Endorsement messaging to set up a A2A session

The Ed25519 keys used for DETs can be converted to Curve25519 keys per [RFC7748] for use in an X25519 ECDH key establishment. The P-256 keys in ECDSA can be directly used in an ECDH key establishment [NIST.SP.800-56Ar3]. However, these keys are not enough for key establishment, nonces are needed to make each encounter has a unique session key. These nonces need to be at least 16 bytes each (from each party in the session). The security concerns of using potentially static, rather than ephemeral EC keys is discussed in Section 8.1.

One use case for this is in DAA. There may be a number of messages between two UA during the DAA event. The DAA event would probably start when each party receives a Vector/Location message showing a

potential encounter. If we use the ASTM F3411-22a Vector/Location message, less the timestamp information, along with a 16 byte nonce, we have the beginning of the additional information for this ECDH key establishment and use of the key in a authenticated session.

Bytes	Name	Explanation
4	VNA Timestamp	Timestamp denoting recommended time to trust Evidence
1	Message ID	A2X Message ID Number
16	Nonce	Random Session Nonce
20	Vector/ Location	Extracted from ASTM Vector/ Location message
120	HDA/UA Endorsement	Compressed HDA on UA Endorsement
64	Signature by UA	Signature over preceding fields using the keypair of the UA DET

Table 4: 225-Byte Nonce enabled A2A message

Such an authenticated session would look much like DTLS, needing a Connection ID to map in the systems to the symmetric connection keys and ending in a keyed-MAC. Note that authentication may be enough and encryption of the message content will rarely be of value. If specific data elements need confidentiality, they can be encrypted in place using AES-CFB.

Carefully constructed, though still random, nonces can be used to construct a 1-byte Connection ID as follows. Each V/L message includes a 16 byte nonce that changes regularly, say once a second. In a 16 (or 32) nonce window, the first 4 bits of the nonce MUST be unique. Then the Connection ID can be: .

```
ConnectionID = ltrunc(othernonce,4)|ltrunc(mynonce,4)
```

Figure 1

Note that there is no provision in the above message for specifying the session protection algorithm. Table 5, below, does a KMAC on the message; this does not provide for crypto agility.

Note that a 1-byte approach could be support of 64 possible algorithms and the above message provide a list of 4 choices in order of preference. This is similar to the DH_GROUP_LIST in the HIPv2 I1 message; at least this list would be protected from tamper by the message signature.

4.1. Session Key Derivation

Each UA converts its Ed25519 private key to an Curve25519 private key. Likewise it converts the received Ed25519 key to an Curve25519 public key (e.g. [Ed25519_Curve25519]). These are then used for each UA to compute the X25519 derived shared secret. Alternatively a P-256 key may be used.

Note that there is no way for one party to use P-256 and the other Curve25519. This method is reserved for cases where both have the same key algorithm.

Here, KMAC from [NIST.SP.800-185] is used with x25519. This is a single pass using the underlying cSHAKE function. The function call is:

```
OKM = KMAC128(salt | info, IKM, 128, S)

Where

IKM     = X25519 ECDH secret | sort(HI-my | HI-other)
salt    = sort(nonce-my | nonce-other)
info    = sort(DET-my | DET-other)
S       = the byte string 01001011 | 01000100 | 01000110
           which is the characters "K", "D", and "F"
           in 8-bit ASCII.
```

Figure 2: Session Key Derivation Function

Note that that for P-256, HKDF with SHA2 would be used. TBA.

4.2. A2A Secure Message

Now a session message would be (33 bytes plus n bytes of the actual message):

Bytes	Name	Explanation
16	Destination DET	DRIP Entity Tag of the destination UA
1	Connection ID	Source UA Connection ID
1	Sequence Number	Source UA Sequence Number
1	Message ID	A2X Message ID Number
n	A2A Message	Actual A2A Message
12	MAC by Source	KMAC on message by Source UA

Table 5: 31+n Byte A2A secure message

Author's Note: Probably should explicitly include the source DET (rather than a SCHC expansion rule) or at least make it explicit ver part of the SCHC expansion rules. This is to protect against cases where multiple UA connected to one destination UA and how to manage ConnectionID to avoid collisions in this 1-byte ID.

4.3. Selection of Nonced Messages

Author's Note: This section needs the state machine clearly drawn out. Otherwise too hard to follow and implement correctly.

In theory once one UA sends its Nonce Enabled A2A message and received a Nonce Enabled A2A message from a nearby UA, it can immediately set up and use the A2A secure messages in Section 4.2. In practice messages will not be received and one UA may operate on the basis that a secure session is possible when the other UA does not have the needed information for the secure session. This section will detail a procedure for UAs to follow to reach that point of common knowledge and thus transition to a secure session.

The Connection ID shown in Figure 1 is unidirectional. It is part of a tuple (Source DET, Destination DET, Connection ID) that contains the operational state of the unidirectional secure messages (e.g. session key, sequence number). Each UA in the secure session link MUST have this tuple, but may have multiple such (at least two, one for each direction) and could be using one Connection ID and session for sending and another for receiving. the basis for the tuple for the secured messaging.

A Connection ID MUST NOT have each 4-bit component the same; this would break the unidirectional feather of the secure connection. When a UA receives a Nonce Enabled A2A message, it MUST send its own and the contained nonce MUST NOT have the same first 4 bits as the one received. After sending its Nonce Enabled A2A message, it MAY immediately switch to a secure session mode.

If a UA receives a secure session message (addressed to its DET), but does not have the security tuple for the contained Connection ID and thus the source DET, it MUST continue to use the Nonce Enabled A2A messages. A UA that receives a Nonce Enabled A2A message MUST work on the basis that the sending UA does not have similar from it. The UA MUST send a Nonce Enabled A2A message and switch to these two messages as

This process MAY be repeated for 4 attempts. At which point it should be assumed something is interfering with message transmissions and act accordingly.

4.4. SCHC compression of DTLS datagram

Author Note: the above datagram was produced from a full IPv6|DTLS message. The MACing is on this original message. The receiver needs to reverse the SCHC before authenticating the message, but this might be a DOS risk.

TBD

5. A2X Messages

Below are the initial defined messages for use in A2X.

ID	Bytes	Name	Explanation
1	20	Location/Vector	ASTM Location/Vector Message less timestamp fields
2	36	L/V and Nonce	Msg 1 plus Nonce

Table 6: A2X Messages

6. Wireless Transport for A2X messaging

Author’s Note: Still need work on what wireless technologies are practical for this approach and even if it is appropriate to discuss actual wireless transport here. There are messages here that are too large for the ASTM Remote ID BT4 limit of 200 bytes, but may well fit into the practical limits of 250 bytes over BT5 and WiFi Beacons. It is even possible of a multi-RF approach, separating the large messages from the short ones.

Author’s Note: We define here a new ASTM Auth message content that is only sent over BT5 and WiFi Beacons.

Author’s Note: Using SCHC as an Ethertype we can position these messages to work over multiple wireless tech. Most noteably IEEE 802.11ah (HiLo) with 802.11bc (Enhanced Broadcast Service). Also 802.16 and 802.15.16t (Licensed Narrowband) addendum to 802.16.

TBD

7. IANA Considerations

TBD

8. Security Considerations

Author’s note: We need to have a discussion on the size of the MAC in the A2A (Table 5) message. Currently a 12-byte MAC is specified. Considering this is an authentication MAC and the messages are timestamped, an 8-byte MAC should be adequate. But the discussion on this is needed and other examples of 8-byte authentication MACs provided.

TBD

8.1. Potentially static EC keys for ECDH

In Section 4 the UA Identity keys are used in a ECDH key derivation operation. The common practice is to use "right now" ephemeral EC keys. This is not practical in this case, as there is very limited time and bandwidth to carry out a full key exchange. Another factor is in most cases the data protected with the derived key is of only immediate value, rarely having any historical worth.

Additionally these Identity keys are normally specific to this mission; new keys are typically used for each flight. Thus, in large measure, they are ephemeral.

Finally, most uses of this approach is only to authenticate an extended exchange between two UA rather than the more bandwidth costly digitally signed messages.

Thus, all factors considered, ECDH key establishment with these Identity keys are practical and within reasonable security bounds.

9. References

9.1. Normative References

[NIST.FIPS.202]
(US), N. I. O. S. A. T. and National Institute of Standards and Technology (U.S.), "SHA-3 standard :", DOI 10.6028/nist.fips.202, 2015, <https://doi.org/10.6028/nist.fips.202>.

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

[NIST.SP.800-56Ar3]
Barker, E., Chen, L., Roginsky, A., Vassilev, A., Davis, R., and National Institute of Standards and Technology, "Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography", DOI 10.6028/nist.sp.800-56ar3, April 2018, <https://doi.org/10.6028/nist.sp.800-56ar3>.

[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>.

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

[RFC9374] Moskowitz, R., Card, S., Wiethuechter, A., and A. Gurtov, "DRIP Entity Tag (DET) for Unmanned Aircraft System Remote ID (UAS RID)", RFC 9374, DOI 10.17487/RFC9374, March 2023, <https://www.rfc-editor.org/info/rfc9374>.

[RFC9434] Card, S., Wiethuechter, A., Moskowitz, R., Zhao, S., Ed., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Architecture", RFC 9434, DOI 10.17487/RFC9434, July 2023, <https://www.rfc-editor.org/info/rfc9434>.

[RFC9575] Wiethuechter, A., Ed., Card, S., and R. Moskowitz, "DRIP Entity Tag (DET) Authentication Formats and Protocols for Broadcast Remote Identification (RID)", RFC 9575, DOI 10.17487/RFC9575, June 2024, <https://www.rfc-editor.org/info/rfc9575>.

9.2. Informative References

[C509-Certificates]
Mattsson, J. P., Selander, G., Raza, S., Hテカglund, J., and M. Furuhed, "CBOR Encoded X.509 Certificates (C509 Certificates)", Work in Progress, Internet-Draft, draft-ietf-cose-cbor-encoded-cert-13, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-cbor-encoded-cert-13>.

[drip-dki] Moskowitz, R. and S. W. Card, "The DRIP DET public Key Infrastructure", Work in Progress, Internet-Draft, draft-moskowitz-drip-dki-09, 23 October 2023, <https://datatracker.ietf.org/doc/html/draft-moskowitz-drip-dki-09>.

[Ed25519_Curve25519]
Libsodium Documentation, "Ed25519 to Curve25519", 2021, <https://libsodium.gitbook.io/doc/advanced/ed25519-curve25519>.

[F3411-22a]
ASTM International, "Standard Specification for Remote ID and Tracking - F3411奸2a", July 2022, <https://www.astm.org/f3411-22a.html>.

[RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/info/rfc7748>.

[RFC8724] Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and JC. Zuniga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation", RFC 8724, DOI 10.17487/RFC8724, April 2020, <https://www.rfc-editor.org/info/rfc8724>.

Acknowledgments

Adam Wiethuechter of AX Enterprize provided review and implementation insights.

Authors' Addresses

Robert Moskowitz
HTT Consulting
Oak Park, MI 48237
United States of America
Email: rgm@labs.htt-consult.com

Stuart W. Card
AX Enterprize
4947 Commercial Drive
Yorkville, NY 13495
United States of America
Email: stu.card@axenterprize.com

Andrei Gurtov
Link旦ping University
IDA
SE-58183 Link旦ping
Sweden
Email: gurtov@acm.org