Workgroup: DRIP
Internet-Draft:
draft-wiethuechter-drip-identity-claims-01
Published: 20 September 2020
Intended Status: Standards Track
Expires: 24 March 2021
Authors: A. Wiethuechter S. Card R. Moskowitz
AX Enterprize AX Enterprize HTT Consulting

DRIP Identity Claims

Abstract

This document describes 7 Identity Claims for use in various Drone Remote ID Protocols (DRIP).

Status of This Memo

This Internet-Draft will expire on 24 March 2021.

Copyright Notice

Table of Contents

1. Introduction

This document expands on Hierarchical HIT Registries [hhit-registries], defining 7 Identity Claims that are created and distributed through the provisioning process of a Unmanned Aircraft in Trustworthy Multipurpose Remote ID (TMRID).

These claims establish trust for Hierarchical HITs [hierarchical-hit]. They are then used in various Drone Remote ID Protocols to establish the trustworthy claims needed to safely use the data provided.

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

HDA (Hierarchical HIT Domain Authority): The 16 bit field identifying the HIT Domain Authority under a RAA.

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

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

3. Host Identity Claims

Under DRIP there are a total of 7 Identity Claims created during the provisioning of the UA to enable trustworthiness. This document specifies the Host Identity Claims forms in detail, the individual Host Identity Claims created and their use in DRIP.

3.1. Why the term: Host Identity Claims

Host Identity Claims are a form of digital certificates, specially crafted for the UAS/USS ecosystem. The term "certificates" has been avoided due to the significant technology and legal baggage around X.509 certificates.

X.509 certificates and Public Key Infrastructure invokes more legal and public policy considerations than probably any other electronic communication sector. It emerged as a governmental platform for trusted identity management and was pursued in intergovernmental bodies with links into treaty instruments.

Thus there is a common expectation whenever the term "Certificates" are used, and the term "Host Identity Claims" to carefully separate these objects from X.509 objects and to emphasize their role as claims.

3.2. Cxx Form

Cxx is a self-signed Host Identity Claim on entity 'x' with the following format:

```
 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
+---------------+---------------+---------------+---------------+
|                                                               |
|                          Hierarchical                         |
|                       Host Identity Tag                       |
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                              Host                             |
|                            Identity                           |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
|                     Expiration Timestamp                      |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                            Signature                          |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
```

This Host Identity Claim form is used to stake an unverified claim onto the specified HI/HHIT pairing, until an expiration time, to be used in DRIP for entity 'x'. All Identity Claims of this form are 116 bytes in length.

The Expiration Timestamp MUST be current UNIX time, at the time of signing, plus an offset into the future. This offset SHOULD be of significant length (possibly years).

3.2.1. Cxx Claims

DRIP uses this form to create three self-signed Host Identity Claims, which are then nested into other claims. The three claims created are:

Aircraft on Aircraft (Caa)

Operator on Operator (Coo)

Registry on Registry (Crr)

These claims (and keypairs needed to create them) SHOULD be created on the entities that own them. Crr on the Registry, Coo on the Operator device and Caa on the Aircraft itself (if able).

These claims could also be stored in DNS under the CERT RR [RFC4398]. The value of doing so is currently unknown.

3.3. Cxy Form

Cxy is a binding Host Identity Claim with the following format:

```
 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
+---------------+---------------+---------------+---------------+
|                                                               |
.                                                               .
.                              Cxx                              .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
.                                                               .
.                              Cyy                              .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
|                           Timestamp                           |
+---------------+---------------+---------------+---------------+
|                     Expiration Timestamp                      |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                            Signature                          |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
```

In the Cxy form a binding is asserted between the entities of 'x' and 'y'. The self-signed Host Identity Claims of Cxx and Cyy are used in the new claim. The new Identity Claims is signed by the first party (in the example above owner of Cxx) using their keypair.

During Host Identity Claim creation Timestamp and Expiration Timestamp MUST be UNIX time (with Expiration Timestamp being created using an offset setting it into the future) and MUST be no later than the Expiration Timestamps found in Cxx and Cyy.

3.3.1. Cxy Claims

In DRIP two Identity Claims are created in the Cxy way, and third one which is a special nesting of the created Host Identity Claims (but following the Cxy form). These claims are as follows:

Registry on Operator (Cro): Using Crr and Coo as Cxx and Cyy respectively, this claims asserts the Registry's Acceptance of the claims in Coo. This MUST be performed on the Registry. It is 304 bytes in length.

Operator on Aircraft (Coa): Using Coo and Caa (Cxx and Cyy respectively) this claim asserts a binding between an Operator and Aircraft. This MUST be performed on the Operator device. It is 304 bytes in length.

Registry on Operator on Aircraft (Croa): A special claim created, asserting the transitivity between Registry, Operator and Aircraft. It uses Cro as Cxx and Coa as Cyy. This MUST be performed on the Registry. It is 680 bytes in length.

The exact methods of transferring data between the entities are out of scope of this document. It MUST be secure, especially when the Registry sends back Croa. It is RECOMMENDED that HIP be used if possible, considering that HHITs are being used already.

Croa is special in that it is similar to an issued automobile registration. The Operator, once it receives Croa via a secure channel from the Registry, should store it somewhere safe to be recalled if required. It SHOULD not be public available, as it can be classified as Personally Identifiable Information (PII).

It is possible that Cro and/or Coa are stored in DNS and are public available as a result. If so, the CERT RR [RFC3498] should be used to store them. It is unknown the value of storing them in DNS gives.

3.4. Claim: Registry on Aircraft

The Registry on Aircraft claim is a special Host Identity Claim defined as follows:

```
 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
+---------------+---------------+---------------+---------------+
|                                                               |
|                          Hierarchical                         |
|                       Host Identity Tag                       |
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
.                                                               .
.                              Caa                              .
.                                                               .
|                                                               |
+---------------+---------------+---------------+---------------+
|                     Expiration Timestamp                      |
+---------------+---------------+---------------+---------------+
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                            Signature                          |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+---------------+---------------+---------------+---------------+
```

This Identity Claim uses the HHIT of the Registry and Caa to form a short (200 byte) certificate to be used on the Aircraft in Broadcast Remote ID.

During Host Identity Claim creation the Expiration Timestamp MUST be UNIX time (with an offset added to it, setting it into the future) and also MUST be no later than the Expiration Timestamp found in Caa.

The Registry HHIT is substituted for Crr to keep the Host Identity Claim within the constraints of Broadcast RID payload size. This optimization does allow for an attacker to attempt a hash collision on the HHIT. This, the authors argue, would be incredible hard as the attacker would need to corrupt DNS to go undetected. That is if a collision on the HHIT is even found in time as it is expected that standard operating procedure for UAS would be to use "one-time" identifiers.

Cra could also be stored in DNS using the CERT RR [RFC3498]. If this is of any benefit has not been explored.

4. Provisioning

This section gives an overview of how an Operator then Aircraft are provisioned under DRIP.

First keypairs are generates on the required devices. Due to limitations in hardware and connectivity it is acceptable to generate the Aircraft keypairs and Host Identity Claims on the Operator device and later embed the data into the Aircraft at the end of provisioning. The methods to securely perform the action of handling the data and embedding it into the Aircraft hardware are out of scope for this document. This section of the document assumes that the Operator is acting on behalf of the Aircraft.

4.1. HHIT Delegation

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

> The entity generates its own HHIT, discovering and using the RAA and HDA for the target Registry. The method for discovering a Registry's RAA and HDA is out of scope here. This allows for the device to generate an HHIT to send to the Registry to be accepted (thus generating the required Host Identity Claim) or denied.

> The entity sends to the Registry its HI for it to be hashed and result in the HHIT. The Registry would then either accept (returning the HHIT to the device) or deny this pairing.

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

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

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

The subsequent sections follow that the device is generating its own HHIT.

4.2. Manufacturer

```
            +--------------+      Ca0a0 +-----------------+
            | Manufacturer | <--------> | Manufacturer CA |
            +--------------+ Cma0       +-----------------+
               ^    |
               |    |
               |    |
       Ca0a0   |    |   Cma0
               |    |
               |    v
            +----------+
            | Aircraft |
            +----------+
```

During the initial configuration and production at a factory the Aircraft MUST be configured to have a Serial Number. This is defined by ASTM as being an ANSI-CTA2063A. Under DRIP a HHIT can be encoded as such (out of scope for this document).

Under DRIP the Manufacturer SHOULD be using HHITs and have their own keypair and Cxx (known here as Cmm). A new Aircraft should generate its own keypairs and generate a Cxx certificate (known as Ca0a0)

Ca0a0 is extracted by the manufacturer and sent to their Certificate Authority (CA) to be verified and added. The resulting certificate (Cma0 here) SHOULD be a certificate of the Cxy form - however it could be a X.509 certificate binding the Serial Number ID to the manufacturer.

4.3. Operator

```
            +----------+            +---------+
            | Registry | ---------> | HDA DNS |
            +----------+  [HIP RR]  +---------+
               ^    |
               |    |
               |    |
         Coo   |    |   Cro
               |    |
               |    v
            +----------+
            | Operator |
            +----------+
```

The Operator generates his HHIT then Coo and sends Coo (along with other relevant information if required) to the desired Registry.

The Registry performs Operator registration, by confirming that no HHIT collisions occur. Coo undergoes a signature verification. If everything passes the Registry optionally adds the HIP RR and other RRs (such as CERT and TXT) into the HDA DNS, generates Cro and transmits it back to the Operator.

Upon receiving Cro the Operator is now registered in the Registry and can proceed to provision any Aircraft. Further verification of Cro can be done, if desired.

4.4. Aircraft

4.4.1. Standard Provisioning

```
            +----------+
            | Registry |
            +----------+
                 ^
                 |
                 |
                 |  Cro, CoaN
                 |
                 |
            +----------+                      +----------+
            | Operator | <--------------------- | Aircraft |
            +----------+         Ca0aN        +----------+
```

First the Operator instructs the onboard Aircraft system to generate a keypair and then extracts from the Aircraft (through a secure mechanism) Ca0aN.

Using Ca0aN, the Operator generates a CoaN and sends to the Registry:

  *Cro; for verification of the Operator

  *CoaN; claim to bind aircraft identity aN to Operator

```
         +----------+
         | Registry |
         +----------+
              |
              |
              |
              |  Token
              |
              v
         +----------+                        +----------+
         | Operator | ---------------------> | Aircraft |
         +----------+       Token            +----------+
```

The Registry performs checks on the received Cro to confirm the identity of the Operator and a check on CoaN to confirm signatures.

In response the Registry sends to the Operator a Token to pass to the Aircraft to continue provisioning.

```
            +---------+
            | HDA DNS |
            +---------+
                 ^
                 |
                 | HIP RR
                 |
                 |
                 |
            +----------+ <-----------------------------+
            | Registry |                               |
            +----------+ -------------------------+    |
                 |                                |    |
                 |                                |    |  Token,
                 |  CroaN                   CraN  |    |  Cma0, Ca0aN
                 |                                |    |
                 |                                |    |
                 v                                v    |
            +----------+                        +----------+
            | Operator |                        | Aircraft |
            +----------+                        +----------+
```

Using the Token as authentication the Aircraft connects via a secure channel to the Registry. Once connected the Aircraft sends:

- *Cma0; authentication for itself to confirm its identity

- *Ca0aN; the Cxx for the new identity it wants to use with the Operator

The Registry first checks that Cma0 checks out via external methods (the manufacturer CA).

The Registry also checks that the aN in Ca0aN matches the original requested aN in the Operators CoaN sent.

In response the Registry generates and issues CroaN to the Operator and CraN to the Aircraft. A HIP RR is also generates and added to the HDA DNS for the new Aircraft identifier.

4.4.2. Operator Assisted Provisioning

The goal of Operator Assisted Provisioning is to support the case where the Aircraft can not itself connect to the Registry and instead uses the Operator as a proxy.

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




         +----------+                       +----------+
         | Operator | ---------------------> | Aircraft |
         +----------+      aN, CaNaN        +----------+
```

To start the Operator generates the Aircraft's new keypair and Cxx certificate. These are inject into the Aircraft (with a secure mechanism).

The Aircraft then generates on its own Ca0aN.

```
         +----------+
         | Registry |
         +----------+
              ^
              |
              |
              |  Cro, Cma0, Ca0aN, CoaN
              |
              |
         +----------+                       +----------+
         | Operator | <--------------------- | Aircraft |
         +----------+       Cma0, Ca0aN     +----------+
```

The Operator now extracts Cma0 and Ca0aN to continue provisioning.

The Operator generates CoaN using Ca0aN and sends to the Registry:

*Cro; for verification of the Operator

*Cma0; for verification of the Aircraft

*Ca0aN; for verification of the binding

*CoaN; claim to bind aircraft identity aN to Operator

```
         +----------+            +---------+
         | Registry | ---------> | HDA DNS |
         +----------+   HIP RR   +---------+
              ^
              |
              |
              |  CroaN, CraN
              |
              |
         +----------+                        +----------+
         | Operator | ---------------------> | Aircraft |
         +----------+          CraN          +----------+
```

After verification of all the received components the Registry generates CroaN and CraN and sends them back to the Operator.

The Operator then securely injects CraN into the Aircraft for use.

4.4.3. Initial Provisioning

A special form of provisioning is used when the Aircraft is first sold to a Operator. Instead of generating a new keypair, the built in keypair (a0, Ca0a0) is used to provision and register the aircraft to the owner.

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

4.5. Registry

It should be noted that the Registry can undergo a similar process as Operator/Aircraft to provision them to an RAA (as a Registry is most likely the HDA). This is currently not specified here for brevity of the document.

5. IANA Considerations

TBD

6. Security Considerations

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

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

7. Acknowledgments

TBD

8. References

8.1. Normative References

[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>.

8.2. Informative References

[hhit-registries] Moskowitz, R., Card, S., and A. Wiethuechter, "Hierarchical HIT Registries", Work in Progress, Internet-Draft, draft-moskowitz-hip-hhit-registries-02, 9 March 2020, <https://tools.ietf.org/html/draft-moskowitz-hip-hhit-registries-02>.

[hierarchical-hit] Moskowitz, R., Card, S., and A. Wiethuechter, "Hierarchical HITs for HIPv2", Work in Progress, Internet-Draft, draft-moskowitz-hip-hierarchical-hit-05, 13 May 2020, <https://tools.ietf.org/html/draft-moskowitz-hip-hierarchical-hit-05>.

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