Table of Contents

[3.1. ATN/OSI 4](#_Toc146279945)

[3.2. ATN/IPS Support 4](#_Toc146279946)

[3.2.1. Principle 4](#_Toc146279947)

[3.2.1. IOA Protocol sublayers 7](#_Toc146279948)

[3.2.1.1. IOA Segmentation function - Segmentation of IPv6 packets and DTLS data 7](#_Toc146279949)

[3.2.1.2. IOA Security function - Cyber Security support 8](#_Toc146279950)

[3.2.1.3. IOA DTLS function - DTLS protocol 9](#_Toc146279951)

[3.2.2. IOA Protocol features 9](#_Toc146279952)

[3.2.2.1. DTLS Traffic encapsulation 9](#_Toc146279953)

[3.2.2.2. IPv6 Traffic encapsulation 10](#_Toc146279954)

[3.2.3. IOA Segmentation function 11](#_Toc146279955)

[3.2.3.1. IOA Segment format 11](#_Toc146279956)

[3.2.3.2. IOA Segmentation state diagram 14](#_Toc146279957)

[3.2.3.2.1. Standby State 15](#_Toc146279958)

[3.2.3.2.2. Active State 17](#_Toc146279959)

[3.2.3.2.3. AVLC link handoff (HANDOFF event) 18](#_Toc146279960)

[3.2.3.3. Emission of IOA Message in Active state 21](#_Toc146279961)

[3.2.3.4. Reception of IOA Message in Active state 24](#_Toc146279962)

[3.2.3.5. Robustness 27](#_Toc146279963)

[3.2.3.5.1. AVLC FRMR event processing 27](#_Toc146279964)

[3.2.4. IOA Security function 28](#_Toc146279965)

[3.2.4.1. IOA Security principle and state diagram 28](#_Toc146279966)

[3.2.4.1.1. Standby State 30](#_Toc146279967)

[3.2.4.1.2. Active State 30](#_Toc146279968)

[3.2.4.2. Adding MIC to IPv6 sending traffic in Active state 31](#_Toc146279969)

[3.2.4.3. Checking MIC for IPv6 received traffic in Active state 31](#_Toc146279970)

[3.2.5. IOA DTLS function 32](#_Toc146279971)

[3.2.5.1. Principle and state diagram 32](#_Toc146279972)

[3.2.5.2. IOA DTLS configuration 36](#_Toc146279973)

[3.2.5.3. IOA DTLS data memorization 39](#_Toc146279974)

[3.2.5.4. MIC computation and MIC sequence numbers 40](#_Toc146279975)

[3.2.5.5. DTLS Handshake procedure 43](#_Toc146279976)

[3.2.5.6. DTLS Resumption procedure 44](#_Toc146279977)

[3.2.5.7. MIC Resynchronization procedure 45](#_Toc146279978)

[3.2.5.7.1. MIC Resynchronization specific messages format 46](#_Toc146279979)

[3.2.5.7.2. MIC Resynchronization on Aircraft request 48](#_Toc146279980)

[3.2.5.7.3. MIC Resynchronization on Ground request 50](#_Toc146279981)

[3.2.5.8. IOA DTLS function robustness 52](#_Toc146279982)

Table of Figures

[Figure 1 - IOA concept 5](#_Toc146279983)

[Figure 2 - IOA features 7](#_Toc146279984)

[Figure 3 – DTLS traffic encapsulation 10](#_Toc146279985)

[Figure 4 – IPv6 traffic encapsulation 11](#_Toc146279986)

[Figure 5 – IOA Segment in AVLC INFO frame context 12](#_Toc146279987)

[Figure 6 – IOA Segment format 13](#_Toc146279988)

[Figure 7 – IOA intermediate / final / single segment 14](#_Toc146279989)

[Figure 8 – IOA Segmentation function state diagram 15](#_Toc146279990)

[Figure 9 – Example of IOA message sending during handoff 20](#_Toc146279991)

[Figure 10 – Multi segment exchange example 25](#_Toc146279992)

[Figure 11 – AVLC retransmission example 26](#_Toc146279993)

[Figure 12 – FRMR / UA sequence example 28](#_Toc146279994)

[Figure 13 – IOA Security state diagram 29](#_Toc146279995)

[Figure 14 – IOA DTLS typical activity example 34](#_Toc146279996)

[Figure 15 – IOA DTLS state diagram 35](#_Toc146279997)

[Figure 16 – Ground MIC Resynchronization Request message 47](#_Toc146279998)

[Figure 17 – Aircraft MIC Resynchronization Request message 47](#_Toc146279999)

[Figure 18 – Ground MIC Resynchronization Response message 48](#_Toc146280000)

[Figure 19 – MIC Resynchronization sequence on Aircraft request 48](#_Toc146280001)

[Figure 20 – MIC Resynchronization sequence on Ground request 50](#_Toc146280002)

Table of Tables

[Table 1 – IOA DTLS configuration parameters 39](#_Toc146280003)

[Table 2 – Definition of MIC computing function parameters 42](#_Toc146280004)

* 1. ATN/OSI
  2. ATN/IPS Support
     1. Principle

With the introduction of ATN/IPS, a subnetwork adaptation is needed to convey the IPv6 frames (from ATN/IPS) over VDL Mode 2.

Therefore, as part of the VDL Mode 2 system, an adaptation layer named IPS Over AVLC (IOA) has been defined.

The IOA Protocol is a protocol designed to make VDL Mode 2 IPS-Capable. This protocol provides the features needed to adapt the IPS traffic over VDL Mode 2.

The IOA Protocol is a protocol operating between the Aircraft and Ground ATN/IPS stacks, and the VDL Mode 2 layers. The IOA Protocol is a new layer between the ATN/IPS communication stack and the VDL Mode 2 layers which is used on the Air-Ground link between the avionics systems and the Ground IOA centralized system as described in Figure 1 - IOA concept.

The VDL link layer is built up with three sublayers:

* A Media Access Control, MAC, based on Carrier Sense Multiple Access.
* The Data link Services, DLS, sublayer acts as a connection-oriented point-to-point or a connectionless broadcast link over the MAC sublayer.
* The Link Management Entity, LME, sublayer establishes and maintains connections between Aircraft and Ground Stations.

The IOA Protocol is located over the LME sublayers and adds the needed provision for segmentation and security.

*Note: The IOA Protocol has been defined to operate over numbered INFO Frames over the existing AVLC protocol.*



Figure - IOA concept

The IOA Protocol provides additional features needed for ATN/IPS over AVLC such as:

* Segmentation of ATN/IPS packets (IOA Segmentation),
* Cyber Security.

The IOA Protocol provides only the needed features for VDL Mode 2 to support ATN/IPS with the objective to minimize:

* The overhead on the VDL Mode 2 bandwidth,
* The protocol complexity.

The IOA Protocol relies on DTLS protocol for the security aspects.

The IOA Protocol is designed to operate over an AVLC link (over AVLC INFO frames) between the Aircraft and Ground Stations. It is designated as IOA mode.

The following are the key names used in this section of the document. This naming is used in Figure 3 – DTLS traffic encapsulation and Figure 4 – IPv6 traffic encapsulation that present the successive steps in the sending or receiving of data.

* **IPv6 packet –** An IOA entry that is either received from or sent to the IPv6 layer.
* **DTLS packet –** An IOA entry that is either received from sent to the IOA DTLS function.
* **IOA Message –** An IPv6 packet plus its additional MIC, or a DTLS packet (no additional MIC). This is the internal IOA element before segmentation for emission or after reassembly for reception.
* **IOA Segment –** This is the exchange unit between the IOA Protocol and the AVLC layer. An IOA Message will be segmented in to one or more IOA Segments (IOA Protocol output to AVLC layer) or reassembled from IOA Segments (IOA Protocol input from AVLC layer).
* **N1uplink** and **N1downlink** – These are the maximum AVLC frame size (INFO frame) parameters. Anytime **N1** is referenced without the additional “uplink” or “downlink” suffix, it must be understood as for uplinks for downlinks because those two N1 values can be different.
* **IOA Segmentation -** The IOA Segmentation function.
* **Sending IOA** **-** The IOA Segmentation function sending side.
* **Receiving IOA -** The IOA Segmentation function receiving side.
* **IOA Security -** The IOA Security function**.**
* **IOA DTLS -** The IOA DTLS function.
  + 1. IOA Protocol sublayers

The IOA Protocol is structured with three sublayers, one for segmentation function, one for security function and one for DTLS function. The below figure presents those three IOA Protocol features in their functional environment.



Figure - IOA features

* + - 1. IOA Segmentation function - Segmentation of IPv6 packets and DTLS data

In the context of ATN/IPS, IOA must convey IPv6 packets (IPS traffic) as well as DTLS data. The IPv6 packets size can be up to 1280 Bytes. The DTLS packets size depends on DTLS segmentation that will be greater than AVLC frame size to minimize DTLS protocol overhead (see section 3.2.5.2 IOA DTLS configuration). The default recommendation for maximum AVLC frame size (INFO frame) is 251 bytes (N1uplink and N1downlink AVLC parameter set to 2008 bits), therefore, there is a need for segmentation of ATN/IPS IPv6 packets and DTLS data.

The IOA segmentation and reassembly capabilities are adapted to the AVLC N1uplink and N1downlink negotiated Air-Ground values. The IOA Segment size complies with N1 dynamically and independently between uplinks and downlinks. The downlink (respectively uplink) IOA Segment size needs to be adjusted to allow the maximum amount of data to be included in a downlink (respectively uplink) VDL Mode 2 INFO frame. This way, the IOA Protocol makes the most efficient use of the AVLC INFO frames capacity to minimize overhead impact on VDL Mode 2. As the AVLC protocol ensures the proper delivery of the data over the Air-Ground link (transmission and ordering of the IOA Segments), the IOA Protocol does not need to add more quality of service (acknowledgement, retransmissions).

*Note: AVLC allows the N1uplink value and N1downlink values to be different.*

With this approach, the overhead to support IPv6 packet and DTLS data segmentation is limited to the strict minimum.

* + - 1. IOA Security function - Cyber Security support

The IOA Security function adds asymmetric cryptography between Aircraft and Ground. It ensures proof of origin, integrity and anti-replay of data conveyed by AVLC frames by adding a Message Integrity Check (MIC) to each IPv6 packet.

The MIC is added to IPv6 traffic packets sent, and checked on IPv6 message reception, providing the necessary security for the IOA Protocol.

The security function relies on the IOA DTLS function to negotiate the Key with the peer. This Key is then used by the IOA Security function in concert with the IOA DTLS function, to:

* Compute and add MIC to output IPv6 packets,
* Check and remove MIC from input IPv6 packets.

*Note: The MIC is applied by IOA Security function only to IPv6 packets as the DTLS authentication traffic is first necessary to exchange the Key that will then be used for the MIC computation; moreover, the DTLS traffic is self-secured and so, a MIC does not need to be applied.*

* + - 1. IOA DTLS function - DTLS protocol

The DTLS function provides DTLS 1.3 protocol on which the Asymmetric cryptography relies.

The DTLS function implements:

* The DTLS Authentication (full Handshake or Resumption) and Key negotiation with peer,
* The MIC computing functionality on which IOA Security function relies for securing IPv6 traffic,
* The encryption of control traffic (such as MIC Resynchronization) between peers.

The DTLS Authentication between Aircraft and Ground is executed first to negotiate the Key. This Key is then used in the MIC computation.

* + 1. IOA Protocol features
       1. DTLS Traffic encapsulation

The DTLS traffic over IOA is the authentication and exchange of Key material used for MIC computation. The DTLS traffic, self-secured, is not secured via MIC processing.

The DTLS maximum packet size will depend on the DTLS segmentation size (see 3.2.5.2 IOA DTLS configuration).

IOA Protocol will segment / reassemble the DTLS packets (IOA Messages) as needed in accordance with AVLC frame size (N1uplink and N1downlink Air-Ground negotiated value on the AVLC link).

As illustrated in the figure below, an IOA header is added at the beginning of each IOA Segment for segment identification and properties. This header is used in reception for identification and reassembly of the IOA Segments.



Figure – DTLS traffic encapsulation

* + - 1. IPv6 Traffic encapsulation

The difference between IPv6 and DTLS packets management is the MIC. An additional step is added to compute and add a MIC to the IPv6 packet before segmentation on the sending side, while on the receiving side the MIC is checked and removed after the reassembly of the IPv6 packet, as shown in the figure below.



Figure – IPv6 traffic encapsulation

* + 1. IOA Segmentation function
       1. IOA Segment format

The IOA Segment format defines the format when data is being carried using IOA Protocol.

The IOA Segment is carried within the information field of the AVLC INFO frame as shown in the figure below.

The IOA sending and receiving segment size is defined by the AVLC negotiated N1uplink and N1downlink values (N1 value defines the maximum length in bits of AVLC sent and received INFO frames).



Figure – IOA Segment in AVLC INFO frame context

The IOA Segment format consists of a two-bytes IOA Segment header (or IOA header) plus the IOA Segment data. The layout of the IOA Protocol segment is shown in the figure below.

**REQ-IOA-001**

IOA Segment **shall** be encoded using network byte order (Big Endian), in accordance with the structure shown in Figure 6 – IOA Segment format where N = (N1 / 8) – 11 is the IOA Segment size in bytes and N1 is the AVLC negotiated size in bits for current AVLC link for the considered transfer direction (N1uplink or N1downlink).



Figure 6 – IOA Segment format

The first thirteen bits of the IOA Segment header identify the IOA Protocol and have a constant value.

**REQ-IOA-002**

The IPI **shall** be set to 0xFF.

**REQ-IOA-003**

The EIPI **shall** be set to 0xFy with y bit 3 set to 0.

The Sec bit (Bit 1 of Byte 1) of the IOA Segment header indicates that if the IOA Segments belongs to a MIC secured IPv6 IOA Message (Sec bit = 1) or to a DTLS IOA Message (Sec bit = 0), then all of the IOA Segments belonging to the same IOA Message will have the same Sec bit value.

The More bit (Bit 0 of Byte1 of IOA Segment Header) indicates that if the IOA Segment is an intermediate segment (More bit = 1) or a final/single segment (More bit = 0) of the IOA Message it belongs to, then a Multi Segments IOA Message will be composed of one or more intermediate segments (More bit to 1) followed by a final segment (More bit to 0). A single segment IOA Message will only be composed of a single segment (More bit to 0).



Figure 7 – IOA intermediate / final / single segment

The Spare bit is not used for now (provision).

**REQ-IOA-004**

The Spare bit (Bit 2 of Byte1 of IOA Segment Header) **shall** be set to 0 by Sending IOA and ignored by Receiving IOA.

**REQ-IOA-005**

Receiving IOA **shall** discard any received IOA Segment whose content is not compliant with the definition of an IOA Segment and record a security event.

*Note: Due to the low probability of this event, it is assumed it could cause MIC check error by dropping an IOA segment from the reception flow (see 3.2.5.4 MIC computation).*

* + - 1. IOA Segmentation state diagram

The following state diagram presents the different states of the IOA Segmentation function.



Figure 8 – IOA Segmentation function state diagram

The IOA Segmentation state diagram is quite simple. The IOA Segmentation function availability follows VDL mode 2 subnetwork status. When VDL mode 2 subnetwork becomes available for IOA traffic (JOIN received from VDL mode 2), the IOA Segmentation function enters the **Active state**. When VDL mode 2 subnetwork is no longer available for IOA traffic (final LEAVE received from VDL mode 2), the IOA Segmentation function enters the **Standby state**.

The VDL mode 2 JOIN, HANDOFF and LEAVE events referenced in ATN OSI environment play the same role for IOA Protocol, even if the data attached to each of them can be different for ATN IPS and for ATN OSI.

* + - * 1. Standby State

**REQ-IOA-006**

At startup, IOA Segmentation **shall**:

* Reset any internal data to its initialization value,
* Enter Standby state.

**REQ-IOA-007**

When VDL mode 2 subnetwork becomes unavailable (final AVLC LEAVE event received from VDL mode 2), IOA Segmentation **shall**:

* Discard any ongoing Multi Segments IOA Message reception and emission,
* Reset any internal data to its initialization value,
* Enter Standby state.

*Note: The IPv6 or DTLS packet that was in the process of being transmitted to the peer is not retained in IOA waiting for the next AVLC link availability (waiting traffic) for several reasons:*

* *The next AVLC connection between the two same peers may take time and the waiting message could be out of context from above DTLS or IPv6 layers point of view at that time. Moreover, the connections/contexts of the above layers could have already expired and disconnected.*
* *In the case of the Aircraft, the next AVLC connection could possibly occur with another Ground Provider having different certificates (the waiting DTLS traffic would be irrelevant) and different Ground IPv6 Address (the waiting IPv6 traffic would be irrelevant).*
* *In the multilink context, the IPv6 traffic will be immediately routed and exchanged through another media than VDL Mode 2 and the waiting traffic would be irrelevant.*

*In all cases, it is the upper layer (IPv6 and above, or DTLS) that will determine how to recover from the AVLC link loss.*

**REQ-IOA-008**

In the Standby state, IOA Segmentation **shall** disable IOA traffic exchange with peer.

*Note: As a result, IOA segments between the IOA Protocol and VDL mode 2 are no longer exchanged.*

*Note: IOA Segmentation shares this disabling with* ***IOA Security*** *and* ***IOA DTLS****. It is the responsibility of the* ***IOA Security*** *to report it to* ***IPv6****.*

* + - * 1. Active State

**REQ-IOA-009**

When VDL mode 2 subnetwork becomes available (AVLC JOIN event received from VDL mode 2), IOA Segmentation **shall:**

* Share Peer Id with IOA DTLS (necessary for IOA DTLS to identify which Certificate to use, which DTLS Session Ticket to use if any available, and which valid Key to use if any available – See 3.2.5.3 IOA DTLS data memorization),
* Enter Active state.

**REQ-IOA-010**

In Active state, IOA Segmentation **shall** enable IOA traffic exchange with active peer.

*Note: IOA Segmentation shares this IOA traffic enabling with* ***IOA Security*** *and* ***IOA DTLS****. It is IOA Security responsibility to report it to* ***IPv6*** *when MIC computation is available.*

*Note: DTLS traffic can always be exchanged with a peer as soon as IOA traffic exchange is enabled. But a valid Key is necessary for IOA Security to be able to manage IPv6 traffic (MIC cannot be computed/checked if there is no valid Key available in IOA DTLS for the current peer).*

Each time VDL mode 2 subnetwork becomes available, IOA receives from VDL Mode 2 a JOIN event including the N1uplink and N1downlink for this new AVLC link. IOA will adapt its sending and receiving segmentation size for emission and reception through this AVLC link.

**REQ-IOA-011**

When receiving a JOIN event for an AVLC link, avionics IOA Segmentation **shall** configure its reassembly segment size to the N1uplink for reception on this AVLC link and its segmentation segment size to the N1downlink for emission on this AVLC link.

**REQ-IOA-012**

When receiving a JOIN event for an AVLC link, Ground IOA Segmentation **shall** configure its segmentation segment size to the N1uplink for emission on this AVLC link and its reassembly segment size to the N1downlink for reception on this AVLC link.

*Note: Those N1uplink and N1downlink values will be applied by IOA for sending/receiving on this AVLC link from the time the JOIN event is received for this AVLC link until a LEAVE event is received from AVLC layer for this AVLC link.*

**Robustness consideration.** It is possible that the Aircraft and Ground may have a different state for the AVLC link.

**REQ-IOA-013**

On the Ground side, if a XID\_CMD\_LE is received while an AVLC link is already established with the same Aircraft, then the Ground IOA **shall** consider that the existing AVLC link is disconnected and a new AVLC link is being established.

*Note: This is based on experience where the avionics and Ground systems may have a different state for the AVLC link.*

**Robustness consideration.** Even if Ground has a valid Key available for an Aircraft that has just connected on AVLC (so Ground could immediately exchange both DTLS and IPv6 traffic), the Ground should wait for the first frame received from the Aircraft to determine the Aircraft status. If the Aircraft does not have valid Key for the current Ground Provider, the Aircraft will immediately send a DTLS Authentication frame (Sec bit set to 0 in the first downlink IOA Segment). If Aircraft has a valid Key for the current Ground Provider, Aircraft will immediately send IPv6 frame with MIC (Sec bit set to 1 in the first downlink IOA Segment).

If Ground has a valid Key for an Aircraft and decides on its own to immediately send IPv6 traffic once an AVLC link is available with this Aircraft, the traffic could be dropped by the Aircraft if the Aircraft does not have a valid Key with the Ground on its side.

**REQ-IOA-014**

When receiving a JOIN event from VDL Mode 2, the Ground IOA **shall** wait for the first downlink IOA Segment from Aircraft to determine Aircraft configuration (valid Key available or not).

* + - * 1. AVLC link handoff (HANDOFF event)

The HANDOFF event received from VDL mode 2 subnetwork does not involve any state change for IOA Segmentation. Unlike JOIN and final LEAVE events, the HANDOFF has no effect on IOA Security and IOA DTLS (those two elements only need to know when VDL mode 2 subnetwork is usable or not, i.e., when JOIN or final LEAVE event is received). However, the HANDOFF event has an impact on IOA Segmentation that needs to be closely detailed.

Each time there is an AVLC link handoff, IOA will receive from VDL Mode 2 a HANDOFF event including the N1uplink and N1downlink for the new AVLC link. IOA will adapt its sending and receiving segmentation size for emission and reception through this new AVLC link. As two AVLC links can be available at the same time during handoff (TG5 period) and the negotiated N1uplink and N1downlink could be different for those two links, it is important IOA differentiates the sending and receiving segmentation size for those two AVLC links.

**REQ-IOA-015**

When receiving a HANDOFF event for a new AVLC link, avionics IOA Segmentation **shall** configure its reassembly segment size to the N1uplink for reception on the new AVLC link and its segmentation segment size to the N1downlink for emission on the new AVLC link.

**REQ-IOA-016**

When receiving a HANDOFF event for a new AVLC link, Ground IOA Segmentation **shall** configure its segmentation segment size to the N1uplink for emission on the new AVLC link and its reassembly segment size to the N1downlink for reception on the new AVLC link.

*Note: Those N1uplink and N1downlink values will be applied by IOA for sending/receiving on this new AVLC link from the time the HANDOFF event is received for this new AVLC link until a LEAVE event reports this AVLC link is no longer usable.*

**Sending operations during handoff**

When a handoff occurs, it is usually that the new AVLC link is healthier than the old one. As a result, the Sending IOA should switch to the new AVLC link as soon as the HANDOFF event is reported.

If a multi-segment transmission was on-going on the old AVLC link while a hand-off occurs, the multi-segment transmission is discarded, and the complete message is considered lost. The next message, single segment or multi-segments, will be sent on the new AVLC link.

*Note: it is the responsibility of the upper layer to retransmit the discarded message by resubmitting it to IOA.*



Figure 9 – Example of IOA message sending during handoff

**REQ-IOA-017**

When receiving a HANDOFF event and there is a Multi Segments IOA Message sending in progress, Sending IOA **shall** discard its remaining IOA Segments and consider that the multi-segment transmission is finished.

**REQ-IOA-018**

When receiving a HANDOFF event, Sending IOA **shall** send new IOA Message (single or multi IOA Segment message) on the new AVLC link.

**Receiving operations during handoff**

During the handoff, it can happen IOA segments can still be received on the old AVLC link. If a complete IOA message can be received on old AVLC link before end of TG5 period, the IOA message is considered as successfully received. On the other hand, a partially received message at the end of TG5 should be discarded.

**REQ-IOA-019**

IOA **shall** drop any partially received IOA multi segments message on the old AVLC link when the handoff period ends.

*Note: This leads to IOA message loss. But this can also lead to IOA message inversion between the new and the old AVLC links during handoff (possible retry on unhealthy old AVLC link can result in reception delay). This is assumed, even if this could lead to MIC checking errors. Also note that the above layers can manage reception of inverted IOA message (for example, DTLS and ATNPKT are able to manage it).*

* + - 1. Emission of IOA Message in Active state

IOA Protocol does not offer the possibility to transmit in parallel several messages, whether they are single or multi segments. The nesting of segments from different messages is not allowed. The same way, the interleaving of segments from different messages is not allowed. IOA Messages are transmitted serially. The transmission of the current Single or Multi Segments IOA Message must be completed before the next Single or Multi Segments IOA Message can be transmitted.

**REQ-IOA-020**

On an AVLC link, Sending IOA **shall** send all the IOA Segment(s) of an IOA Message (single or multi IOA Segments message) before selecting another IOA Message (single or multi IOA Segments message) for segmentation and sending.

*Note: This is also the case during handoff, the emission in progress on an old link is immediately cancelled before emission of a new IOA message begins on new AVLC link (it is AVLC exchanges that can cause IOA message inversion, not IOA module on its own) – See 3.2.3.2.3 AVLC link handoff (HANDOFF event).*

The DTLS traffic should always have priority over the IPv6 traffic. DTLS traffic concerns authentication, Key negotiation for MIC computation, MIC resynchronization procedure, that are necessary (as they allow to get new Key) before IPv6 traffic can be exchanged.

**REQ-IOA-021**

In Active state, IOA Segmentation **shall** always give priority to sending DTLS traffic over sending IPv6 traffic.

As a corollary of previous requirement …

**REQ-IOA-022**

In IOA Segmentation Active state, when IOA DTLS switches to the DTLS Handshake, DTLS Resumption or MIC Resynchronization procedure, IOA Segmentation **shall:**

* Drop any sending IPv6 traffic in progress at that time,
* Stop managing new sending IPv6 traffic as long as the IOA DTLS procedure is in progress,
* Drop any received IPv6 traffic as long as the IOA DTLS procedure is in progress.

*Note: This is because IPv6 traffic first embeds a MIC computed with Key / MIC sequence number that are being renegotiated (so they would most probably be incorrect) and second, sending IPv6 traffic in progress prevents from immediately sending the DTLS traffic to the peer.*

The first step for Sending IOA is to determine if the IOA message to send fits in a single IOA segment or not.

**REQ-IOA-023**

When IOA header (2 bytes) + DTLS packet can be sent in one AVLC frame as defined by N1, Sending IOA **shall** build a single IOA Segment message, otherwise a multiple IOA Segment message is built.

**REQ-IOA-024**

When IOA header (2 bytes) + IPv6 packet + MIC (4 bytes) can be sent in one AVLC frame as defined by N1, Sending IOA **shall** build a single IOA Segment message, otherwise a multiple IOA Segment message is built.

The second step for Sending IOA is to fill the IOA header fields for the IOA Segment(s).

**REQ-IOA-025**

IOA Segmentation **shall:**

* Set the More bit to 0 for IOA single segment and for the last IOA segment of a multi segment message,
* Set the More bit to 1 for intermediate IOA segment(s) of a multi segment message.

**REQ-IOA-026**

IOA Segmentation **shall:**

* Set the Sec bit to 0 for (all) the segment(s) of an DTLS protocol IOA message (no MIC present),
* Set the Sec bit to 1 for (all) the segment(s) of an IPv6 protocol IOA message (MIC present).

**REQ-IOA-027**

For maximum efficiency and minimal protocol overhead, Sending IOA **shall** fill with data each intermediate IOA Segment up to its maximum possible size (i.e. AVLC header and tail (11 bytes) + IOA Segment header (2 bytes) + IOA Segment data size equal to the AVLC frame size as defined by N1 – See Figure 5 – IOA Segment in AVLC INFO frame context).

*Note: Each AVLC INFO frame for an intermediate IOA Segment will be filled with maximum data size.*

**REQ-IOA-028**

Sending IOA **shall** fill single IOA Segment and final IOA segment of the multi segment with the remaining data to send.

*Note: Concerning IPv6 traffic, the data part includes the MIC as the last 4 bytes of the single or final IOA segment of the IOA message.*

The final step is for Sending IOA to send the IOA Segment(s) of the IOA message to the AVLC layer.

**REQ-IOA-029**

Sending IOA **shall** sequentially send the successive IOA Segments of the single or multiple IOA Segments Message to the AVLC layer until all the IOA Segments are sent.

*Note: The VDL Mode 2 is limited in its sending capacities (transmit window size of 4 INFO frames) and speed. On the opposite IOA and above protocols can generate traffic far quicker and can saturate the AVLC transmit queue, resulting in possible IOA Segments dropping. It is highly recommended to put in place a flow control mechanism between IOA and VDL Mode 2 to grant that the IOA Segments sent to VDL Mode 2 will not be dropped. This flow control mechanism depends on local design implementation.*

* + - 1. Reception of IOA Message in Active state

On an AVLC link, the AVLC protocol grants all the IOA Segments will be received and received in the exact order they have been sent.

**REQ-IOA-030**

When Receiving IOA Segment from the peer, Receiving IOA **shall** store it ordered as received in the IOA Segmentation receiving queue for the corresponding AVLC link.

*Note: The IOA Segment(s) of an IOA Message is(are) stored in the order they are received until IOA Message fully received and reassembled.*

*Note: During handoff, the received IOA segments are managed on a per AVLC link basis. An IOA message will never be split between the two AVLC links.*

**REQ-IOA-031**

When a received IOA Segment has a More bit set to 0 (single or final segment), Receiving IOA **shall** reassemble the IOA Message (i.e., concatenate all the IOA Segment data parts in the received order, any previous IOA segments with More bit set to 1 if any, until this segment with More bit set to 0).

**REQ-IOA-032**

When an IOA Message with Sec bit set to 0 (for all its segments if more than one) is successfully reassembled, Receiving IOA **shall** send it to IOA DTLS.

**REQ-IOA-033**

When an IOA Message with Sec bit set to 1 (for all its segments if more than one) is successfully reassembled and IOA Security in Active state (MIC computation is activated), Receiving IOA **shall** send it to IOA Security function for authentication (MIC checking), otherwise (IOA Security in Standby state) the IOA message is dropped, and a security event recorded.

The following diagram illustrates a multi segments IOA Message transmission (IPv6 traffic in this example).



Figure 10 – Multi segment exchange example

The following diagram illustrates the case where some AVLC frames are retransmitted without impact at IOA Protocol level.



Figure 11 – AVLC retransmission example

*Note: The example above illustrates that AVLC provides the quality of service needed to convey IOA Segments properly without the need for an IOA acknowledgement mechanism.*

**REQ-IOA-034**

In IOA Segmentation Active state, if IOA DTLS is in DTLS Handshake, DTLS Resumption or MIC Resynchronization procedure, IOA Segmentation **shall** drop any received IPv6 traffic.

*Note: This traffic embeds a MIC computed with Key / MIC sequence number that are being renegotiated (so most probably incorrect).*

**REQ-IOA-035**

If the segments belonging to a reassembled IOA multi segment message do not have the same Sec bit value, Receiving IOA **shall** drop the IOA message and record a security event.

**REQ-IOA-036**

If the reassembly of successive IOA segments would lead to an IPv6 packet greater than the max IPv6 packet size (1280 bytes excluding the 4 MIC bytes) for an IOA message with Sec bit set to 1 or the max DTLS size (DTLS segmentation size defined in 3.2.5.2 IOA DTLS configuration) for an IOA message with Sec bit set to 0, Receiving IOA **shall** drop the IOA segments and record a security event.

* + - 1. Robustness
         1. AVLC FRMR event processing

When DM or DISC are exchanged on an AVLC link, the AVLC link is disconnected resulting in a LEAVE event to IOA: however, with the FRMR/UA AVLC sequence, the AVLC link resets silently without being disconnected (no LEAVE event in this case). The reset of an AVLC link result in the loss of the uplink and downlink INFO frames present at that time in the VDL Mode 2 internal transmission and reception queues. This can lead to the silent loss of IOA uplink / downlink segments and the desynchronization of Aircraft and Ground sending / receiving MIC Sequence Numbers (i.e. MIC Resynchronization procedure needed to resynchronize Aircraft and Ground IOA DTLS function, see 3.2.4.3 Checking MIC for IPv6 received traffic in Active state). To avoid that, the AVLC link should provide an end of FRMR/UA sequence occurrence event to IOA.

**REQ-IOA-037**

When an AVLC link reports an FRMR/UA sequence is achieved (UA frame exchanged), the IOA context **shall** be reset:

* The Sending/Receiving MIC Sequence Number are reset to 0 (IOA DTLS function),
* Any ongoing multi segments reception is discarded (received segments dropped),
* Any ongoing multi segments emission is discarded (Sending IOA message in progress fully dropped).

*Note: IOA Segmentation will report this FRMR/UA event to* ***IOA DTLS*** *for resetting Sending/Receiving MIC Sequence Number to 0. This is the only action expected from* ***IOA DTLS*** *concerning this event.*

The figure bellow provides an example of an FRMR / UA sequence showing the loss of IOA Messages and IOA Segments.



Figure 12 – FRMR / UA sequence example

* + 1. IOA Security function
       1. IOA Security principle and state diagram

IOA Security is based on a Message Integrity Check (MIC) added to each IPv6 traffic message based on a negotiated Key and incremental sending and receiving MIC sequence number. The Key is obtained during a DTLS Authentication (see 3.2.5 IOA DTLS function). The IPv6 traffic can only be managed once the Key is available in IOA DTLS to compute the MIC. Both Key and MIC sequence numbers are managed by IOA DTLS.

IOA Security is responsible for:

* Adding MIC to sending IPv6 packets,
* Checking and removing MIC from received IPv6 packets.

The following figure presents a state diagram for the IOA Security function depending on the events that can impact the function availability.



Figure 13 – IOA Security state diagram

IOA Security **Standby state** is the initialization state, as well as the state that is immediately entered when VDL mode 2 subnetwork is no longer available, when a Key with the current peer is no longer valid, when no Key is available for current peer or when there is a MIC check failure (i.e. MIC resynchronization is in progress).

IOA Security **Active state** is the state where VDL mode 2 subnetwork is active, Key is available and valid for the current peer, and there is no MIC resynchronization in progress. It is the only IOA Security state where IPv6 traffic can be exchanged with peer.

* + - * 1. Standby State

**REQ-IOA-038**

IOA Security **shall** be in standby state and disable traffic to/from IPv6 layer when:

* VDL mode 2 subnetwork is not available (Join event not received or final Leave event received from VDL mode 2) or
* There is no valid Key for the current peer in IOA DTLS (not negotiated at all or expired) or
* There is an IOA DTLS MIC Resynchronization procedure in progress following a MIC check failure.

*Note: It is a local design choice on whether and how this unavailability is reported to IPv6 layer and above.*

*Note: No additional IPv6 sending traffic to the peer is provided to IOA Segmentation while IOA Security is in Standby state. Any IPv6 received traffic from the peer that would reach IOA Security in Standby state is discarded.*

*Note: See 3.2.5 IOA DTLS function for the different reason that could result in Key not available in IOA DTLS.*

* + - * 1. Active State

**REQ-IOA-039**

IOA Security **shall** be in active state and enable traffic to/from IPv6 layer when:

* VDL mode 2 subnetwork is available (Join event received and no final Leave event received from VDL mode 2) and
* There is valid Key for current peer in IOA DTLS and
* There is no IOA DTLS MIC Resynchronization procedure in progress in IOA DTLS.

*Note: It is a local design choice on whether and how this availability is reported to IPv6 layer and above.*

* + - 1. Adding MIC to IPv6 sending traffic in Active state

**REQ-IOA-040**

For sending IPv6 packet, IOA Security **shall:**

* Request to IOA DTLS to compute the MIC for the IPv6 packet,
* Concatenate the 4 MIC bytes returned by IOA DTLS at the end of the IPv6 packet,
* Make the IPv6 packet + 4 MIC bytes (IOA Message) available to IOA Segmentation for sending to the peer IOA.

*Note: IOA Segmentation should be the one to ask IOA Security to compute the MIC and give the next IPv6 message to send. Computing MIC in advance would consume sending MIC sequence number values that could possibly be dropped (for example following a VDL mode 2 FRMR/UA sequence) and further leading to a possible MIC resynchronization procedure. So, it is highly recommended IOA Segmentation requests next IPv6 sending message to IOA Security when IOA Segmentation is ready to send it to ground. And IOA Security should add the MIC to an IPv6 sending message only when an Ipv6 sending message is requested by IOA Segmentation.*

* + - 1. Checking MIC for IPv6 received traffic in Active state

**REQ-IOA-041**

When IOA Security receives an IOA Message (IPv6 packet + 4 MIC bytes) from IOA Segmentation, IOA Security **shall** request to IOA DTLS to check the IPv6 packet versus its associated MIC value.

**REQ-IOA-042**

When IOA DTLS reports a successful MIC check procedure for a received IPv6 packet, IOA Security **shall** forward the IPv6 packet with the 4 MIC bytes truncated to the IPv6 layer.

**REQ-IOA-043**

When IOA DTLS reports a failed MIC check procedure for a received IPv6 packet, IOA Security **shall**:

* Drop the IPv6 packet that failed the MIC check,
* Disable traffic to/from IPv6 layer,
* Enter the Standby state.

*Note: On its side, IOA DTLS having failed the MIC check enters the MIC resynchronization procedure.*

* + 1. IOA DTLS function

This section describes the means used to provide ATN/IPS secured communications over VDL Mode 2 subnetwork.

* + - 1. Principle and state diagram

The security aspect is based on a combination of DTLS and MIC (Message Integrity Check). DTLS is used to provide peer authentication while the MIC is used for data integrity and anti-replay (see 3.2.4 IOA Security function for use of the MIC mechanism).

At the initialization of communications, DTLS is used to authenticate the peers. Once the peer authentication is performed, the Key negotiated during the DTLS Authentication remains available in IOA DTLS for further MIC computation. Once the Key is available, a MIC is computed for each received and sent IPv6 packet to provide data integrity and authentication to secure the VDL Mode 2 subnetwork. The MIC, based on the negotiated Key and the sending/receiving MIC sequence numbers, is needed to avoid replay attacks.

The VDL Mode 2 subnetwork and AVLC protocol ensure the delivery of the IOA Segments between the Aircraft and Ground Systems. Nevertheless, to cover exceptional cases where the MIC is out of synchronization between the peers, the MIC Resynchronization procedure is a means for avionics and Ground systems to realign their MIC sequence numbers.

When a Key is no longer valid (limited lifetime for security consideration), a DTLS renegotiation is necessary to get new Key. The DTLS renegotiation can be a full DTLS Handshake procedure, or a limited procedure based on the use of DTLS SessionTicket if any available, the DTLS Resumption (a DTLS Resumption based on SessionTicket minimizes the DTLS traffic to get new Key).

The following figure presents the three procedures IOA DTLS oversees:

* A full DTLS Handshake with the peer,
* A DTLS Resumption with the peer,
* A MIC Resynchronization.

Also note that in the following figure, each time a MIC is computed to add to an IPv6 packet or to check integrity of an IPv6 packet, IOA DTLS provides the MIC computing function to IOA Security (IOA Security provides the data while IOA DTLS uses the uplink/downlink MIC Sequence Numbers, the negotiated Key for the concerned peer and the PRF function to compute the MIC – See 3.2.5.4 MIC computation).



Figure 14 – IOA DTLS typical activity example

The following figure presents a state diagram for the IOA DTLS.



Figure 15 – IOA DTLS state diagram

The IOA DTLS **Standby state** is the initialization state. This state is also immediately entered each time the VDL 2 subnetwork is no more available. All internal data are reset to initialization value in this state except SessionTicket and Key.

The IOA DTLS **Active state** is the state where a DTLS session is active, and MIC can be computed on IOA Security request. IOA DTLS is in this state when valid Key is available with the current peer.

The IOA DTLS **Handshake state** is entered when there is the need for DTLS negotiation and there is no valid Key nor valid SessionTicket with the current peer. In this case, a full handshake procedure with current peer is necessary to get a valid Key with the current peer.

The IOA DTLS **Resumption state** is entered when there is the need for DTLS negotiation, and there is valid SessionTicket with the current peer but no valid Key with it. In this case, a DTLS Resumption procedure is executed (new air/ground authentication but with limited exchanges compared to a full handshake) to get a valid Key with the current peer.

The IOA DTLS **MIC Resynchronization state** is entered when a MIC check failure occurs. In this case, the peers try to resynchronize each other by negotiating new set of MIC uplink/downlink sequence numbers. Once new sequence numbers are obtained, the **Active state** is entered.

*Note: At Aircraft side, it is recommended to anticipate the Key expiration by executing a DTLS Handshake or DTLS Resumption some time before the Key expires. For example, Key could be negotiated each time the Aircraft is on the ground between flights, when the critical communication need is reduced. However, this is not always possible as this requires to be connected to the concerned Ground provider at that time (i.e. the Key for a Ground provider can expire while VDL mode 2 connected to another Ground provider).*

Note that there are transitions from IOA DTLS **MIC Resynchronization state** to IOA DTLS **Handshake state** and IOA DTLS **Resumption state** in the IOA DTLS state diagram for robustness reason. They correspond to the fact a DTLS Authentication (DTLS Handshake if no SessionTicket available, DTLS Resumption if SessionTicket available) has priority over a DTLS MIC Resynchronization and can interrupt it. For example, if a Key timeout occurs during a MIC Resynchronization.

* + - 1. IOA DTLS configuration

**General Considerations:**

**REQ-IOA-044**

IOA DTLS **shall** implement DTLS 1.3 protocol as defined in RFC 9147.

*Note: This objective can be reached by embedding a DTLS COTS.*

**REQ-IOA-045**

Ground IOA DTLS **shall** be configured as DTLS Server.

**REQ-IOA-046**

Aircraft IOA DTLS **shall** be configured as DTLS Client.

**REQ-IOA-047**

The DTLS session **shall** remain opened at end of authentication.

*Note: MIC resynchronization can happen at any time after a DTLS authentication, requiring the DTLS session to remain open.*

**REQ-IOA-048**

IOA DTLS **shall** be configured with a segmentation size of 1024 bytes.

*Note: DTLS COTS protocol supports segmentation size on power of 2 bytes (256 bytes, 512 bytes, 1024 bytes, 2048 bytes …). The IPv6 max size is 1280 bytes. Using 1024 bytes segmentation size for DTLS protocol allows having quite similar IPv6 and DTLS max frame sizes. 2048 for DTLS protocol would be significantly larger than IPv6 traffic and would require much more RAM resource.*

*Note: Having a similar limit for both DTLS and IPv6 maximum traffic size allows for a maximum IOA message size for IOA module implementation.*

**REQ-IOA-049**

To reduce the DTLS protocol overhead during DTLS handshake and resumption procedures, IOA DTLS **shall** concatenate the DTLS frames belonging to the same flight and having the same type (Plaintext or Cyphertext) before exchanging them with IOA Segmentation.

*Note: It is a DTLS protocol limitation to forbid the concatenation of frames from different type (Plaintext or Cyphertext).*

*Note: DTLS COTS configuration usually provides a means to activate and manage on its own the concatenation.*

**REQ-IOA-050**

IOA DTLS **shall** be configured with a maximum DTLS Handshake or DTLS Resumption duration of 30 seconds.

*Note: It is a local design consideration as to what sanction should be taken when the maximum duration limit is reached. Recommended practice would be a limited number of retries and if the problem remains, Aircraft should declare the VDL mode 2 subnetwork with current ground peer as not usable (possibly incompatible certificates - no need to replay infinitely a negotiation that would always fail and consume bandwidth).*

*Note: Implementation shows a DTLS full handshake execution time between 11s to 13s.*

*Note: TLS/DTLS COTS are provided by default with typical Ethernet traffic timing configuration … i.e., DTLS/TLS flight max duration and DTLS/TLS handshake duration set to about 1s …* ***It is necessary to pay attention to the default DTLS/TLS COTS timing configuration and redefine them for VHF communication use as there can be a factor 10 or more between them.***

**DTLS session resumption and SessionTicket considerations:**

**REQ-IOA-051**

The DTLS Resumption based on SessionTicket **shall** be supported.

**REQ-IOA-052**

The Ground DTLS Server **shall** be configured to send one single NewSessionTicket message at the end of each successful authentication (full handshake or resumption).

*Note: By default, DTLS COTS are configured to send 2 NewSessionTicket message as per RFC 9147 at the end of a DTLS negotiation. The size of a NewSessionTicket message can be more than 1 Kbyte. And only one SessionTicket is needed for IOA use (only 1 Air-Ground DTLS session at a time).*

**REQ-IOA-053**

The lifetime of SessionTicket **shall** be 72 hours.

**Key management and MIC resynchronization considerations:**

**REQ-IOA-054**

The lifetime of the negotiated Key for MIC computation **shall** be 48 hours.

**REQ-IOA-055**

The maximum duration for a Resynchronization procedure **shall** be 10 seconds.

**Additional RFCs to implement:**

RFCs possibly to be considered :

* RFC 6066 Extension Definitions - Specifically Server Name and Status request
* RFC 8879 Certificate compression
* RFC 7924 Cached Information Extension

Any other RFC that would reduce the DTLS protocol overhead during the authentication phase can be considered.

**Configuration items definition:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter name** | **Parameter definition** | **Lower bound** | **Upper bound** | **IOA default** |
| DTLS\_SEGMENTATION\_SIZE | DTLS protocol segmentation size | N/A | N/A | 1024 |
| DTLS\_MAX\_NEGO\_ DURATION | Maximum DTLS Handshake or DTLS Resumption duration | 20 s | 1 minute | 30 s |
| DTLS\_MAX\_RESYNC\_DURATION | DTLS max Resyncrhonization procedure duration | 5 s | 30 s | 10 s |
| DTLS\_SESSION\_TICKET\_LIFETIME | Lifetime of SessionTicket | 48 hours | 1 week | 72 hours |
| DTLS\_KEY\_LIFETIME | Lifetime of negotiated key | 24 hours | 1 week | 48 hours |

Table 1 – IOA DTLS configuration parameters

* + - 1. IOA DTLS data memorization

There are three kind of data that both the Air and Ground side should locally memorize in IOA DTLS:

* The certificate for DTLS negotiation on a per peer basis. It should always be available. It is a static configuration element that persists power ON / OFF. It is rarely updated.
* The SessionTicket. It is provided by DTLS server to the DTLS client at the successful end of a DTLS Handshake or DTLS Resumption. It is used for the next DTLS Resumption to minimize the air/ground DTLS negotiation traffic. It should be memorized on a per peer basis until no longer valid or replaced by a newly negotiated one, independently of the VDL mode 2 subnetwork availability.
* The Key, result of a DTLS Handshake or DTLS Resumption with a given peer. It is used for MIC computation and MIC checking. It should be memorized on a per peer basis until no longer valid or replaced by a newly negotiated one, independently of the VDL mode 2 subnetwork availability.

*Note: It is a local design choice to have SessionTicket and Key survive a power OFF/ON at Aircraft side.*

**Certificates memorization:**

**REQ-IOA-056**

DTLS negotiation **shall** be limited to the final certificates exchange.

*Note: Needed Root certificate and intermediate certificates should always be locally available to minimize the DTLS negotiation traffic overhead.*

**REQ-IOA-057**

Aircraft and Ground **shall** store locally in a persistent way (i.e. survive power OFF/ON) Certificates data necessary for DTLS authentication with the peer.

*Note: How the certificates are saved and updated is a local design consideration. Root and intermediate certificates should be available so that only the final certificate is exchanged for correct authentication.*

**SessionTicket and Key memorization:**

**REQ-IOA-058**

IOA DTLS **shall** store the SessionTicket and Key data on a per peer basis.

**REQ-IOA-059**

Stored SessionTicket and Key data **shall** survive IOA DTLS cleaning (i.e. survive to Standby state) as they are managed by their own lifetime.

**REQ-IOA-060**

IOA DTLS **shall** discard:

* expired SessionTicket (DTLS\_SESSION\_TICKET\_LIFETIME),
* expired Key (DTLS\_KEY\_LIFETIME).

*Note: SessionTicket and Key can also be replaced or cleared on IOA DTLS state transition (see next sections).*

* + - 1. MIC computation and MIC sequence numbers

The Message Integrity Check (MIC) includes a sequence number to avoid replay attacks. This sequence number is initialized to 0 and increments by 1 at every IPv6 packet. There is one Sending and one Receiving MIC Sequence Number, as sending and receiving traffic are independent.

**REQ-IOA-061**

The Sending MIC Sequence Number and the Receiving MIC Sequence Number **shall** be 6 bytes long each.

**REQ-IOA-062**

Each time the VDL mode 2 subnetwork becomes available, the Sending MIC Sequence Number and the Receiving MIC Sequence Number **shall** be initialized to 0.

*Note: Some specific events like AVLC FRMR/UA sequence can also cause a reinitialization to 0 of the Sending and Receiving MIC Sequence Numbers (see 3.2.3.5.1 AVLC FRMR event processing). On its side, the MIC Resynchronization procedure (see 3.2.5.7 MIC Resynchronization procedure) allows to resynchronize Ground and Aircraft with specific negotiated Sending and Receiving MIC Sequence Numbers values.*

**REQ-IOA-063**

The MIC for an IPv6 packet **shall** be computed by applying the following function to the concerned message data content:

MIC = Truncate (4, PRF(Session Key, App Data + SN#))

|  |  |
| --- | --- |
| **Variable Name** | **Description** |
| Truncate | A Truncate function that reduces the size of the operator to a number of bytes. In this case the **first 4 bytes** of the message hash function will be used as a message integrity check (i.e. MIC value for the IPv6 packet). |
| PRF | Pseudo Random Function: algorithm used to generate the Key. The PRF is an HMAC384 as per RFC 4868. The PRF is provided as a service by DTLS COTS. |
| App data | The IPv6 packet whose MIC is to be computed. |
| SN# (6 Bytes) sequence number | The Sending MIC Sequence Number or the Receiving MIC Sequence Number, depending if MIC is computed for sending an IPv6 packet to peer or for checking the MIC on an IPv6 packet received from peer.  The SN# is big endian formatted. |
| Session Key (32 Bytes) | This is the lower 32 bytes of the session Key derived as per RFC 5246 Section 6.3. Both the gateway (Server) and Aircraft (Client) have a session or master Key and compute the counter parties’ Key using the procedure recorded in the RFC. This value is never transmitted making the PRF function difficult to duplicate by third parties. |

Table 2 – Definition of MIC computing function parameters

**REQ-IOA-064**

IOA DTLS **shall** report to IOA Security:

* The MIC computed for IPv6 packet to send,
* The result of the MIC checking procedure for IPv6 received packet.

**REQ-IOA-065**

In case of MIC check failure, IOA DTLS **shall** initiate a MIC Resynchronization procedure.

**REQ-IOA-066**

After each MIC computation at sending side or MIC checking at reception side on IOA Security request, IOA DTLS **shall** increment by 1 the corresponding uplink or downlink MIC Sequence Number.

*Note: It is a local design choice to implement a MIC checking window mechanism or not. If no window mechanism, the MIC is checked only with expected receiving MIC sequence number value and if fails, this immediately causes a MIC resynchronization procedure. If a window mechanism is implemented, then if the check with expected receiving sequence number fails, some successive incremented MIC sequence number values can be attempted (depending on the window width). This can help minimize the number of MIC Resynchronization procedures (limited consecutive IOA message loss can be workaround this way) but computing several successive MICS can be time and CPU consuming. Moreover, this can only workaround (a few successive) message loss, but not message inversion. Based on prototyping experience, this kind of mechanism proved useful.*

* + - 1. DTLS Handshake procedure

As Aircraft is the DTLS Client, it is always the Aircraft side IOA DTLS that will initiate DTLS Handshake procedure. On the other hand, it is when detecting the beginning of a Handshake procedure with an Aircraft (DTLS traffic received from IOA Segmentation) that Ground IOA DTLS executes DTLS Handshake procedure with the Aircraft.

**REQ-IOA-067**

When there is the need for a DTLS handshake (at initialization or when Key needed) and there is no SessionTicket available for the current Ground, Aircraft IOA DTLS **shall:**

* Command IOA Security to enter Standby state,
* Inform IOA Segmentation that DTLS Handshake procedure is in progress,
* Initiate and execute a DTLS Handshake procedure with the current Ground.

**REQ-IOA-068**

If Ground IOA DTLS identifies the beginning of a DTLS Handshake in downlink traffic from the Aircraft and Ground IOA DTLS has an active DTLS session or a DTLS session already in negotiation with this Aircraft, the Ground IOA DTLS **shall** immediately close this active or in progress DTLS session and execute the new DTLS Handshake procedure with the Aircraft.

**REQ-IOA-069**

A DTLS Handshake procedure between Aircraft and Ground **shall** be executed in compliance with RFC 9147 (Authentication and DTLS Session Key negotiation with the Certificates available for the corresponding peer).

**REQ-IOA-070**

As a post handshake message, Ground IOA DTLS **shall** send to the Aircraft a single NewSessionTicket message.

*Note: The data (i.e. SessionTicket) included in the NewSessionTicket message allows the DTLS Client to further open a DTLS session with the Ground DTLS Server having generated it, with less exchanged messages and data than for a full handshake procedure.*

*Note: The reception of the NewSessionTicket by the Aircraft ends the authentication procedure.*

**REQ-IOA-071**

At the end of successful DTLS Handshake procedure, IOA DTLS **shall:**

* Store the new negotiated Key to be used with this peer,
* Store the new SessionTicket to be used with this peer,
* Turn to Active state,
* Command IOA Security to enter Active state,
* Inform IOA Segmentation DTLS Handshake procedure is finished,
* Keep the DTLS session open (used for MIC Resynchronization procedure).

*Note: It is a local Aircraft design consideration on how to manage the case of the DTLS Handshake failure (retries or not? - VDL mode 2 Ground provider switch on repeated error possibly caused by incompatibles certificates?). The Ground side is in a wait-and-see policy.*

*Note: For example, a DTLS Handshake procedure can fail if DTLS\_MAX\_NEGO\_ DURATION is reached, if one of the peers’ certificates expired …*

* + - 1. DTLS Resumption procedure

When SessionTicket is available at Aircraft DTLS side for the current Ground, a DTLS session Resumption procedure can be executed instead of a full handshake procedure. Similarly, when detecting the beginning of a Resumption procedure with an Aircraft (DTLS traffic received from IOA Segmentation), then Ground IOA DTLS executes DTLS Resumption procedure with the Aircraft.

**REQ-IOA-072**

When there is the need for a DTLS handshake (at initialization or when a Key is needed) and there is SessionTicket available for the current Ground, Aircraft IOA DTLS **shall:**

* Command IOA Security to enter Standby state,
* Inform IOA Segmentation DTLS Resumption procedure in progress,
* Initiate and execute a DTLS Resumption procedure with the current Ground.

**REQ-IOA-073**

If Ground IOA DTLS identifies the beginning of a DTLS Resumption in downlink traffic from Aircraft and Ground IOA DTLS still has an active DTLS session with this Aircraft, Ground IOA DTLS **shall** immediately close this active DTLS session and execute the DTLS Resumption procedure with the Aircraft.

**REQ-IOA-074**

A DTLS Resumption procedure between Aircraft and Ground **shall** be executed in compliance with RFC 9147 (Authentication and DTLS Session Key negotiation with the Certificates available for the corresponding peer).

**REQ-IOA-075**

At the end of the DTLS Resumption procedure, Ground DTLS **shall** send to the Aircraft a single NewSessionTicket message.

**REQ-IOA-076**

At the end of successful DTLS Resumption procedure, IOA DTLS **shall:**

* Store the new negotiated Key to be used with this peer,
* Store the new SessionTicket to be used with this peer,
* Turn Active state,
* Command IOA Security to enter the Active state,
* Inform IOA Segmentation DTLS Resumption procedure is finished,
* Keep the DTLS session open (used for MIC Resynchronization procedure).

**REQ-IOA-077**

If the DTLS Resumption with Ground peer fails (for example, because Ground does not have a SessionTicket for this Aircraft), Aircraft IOA DTLS **shall**:

* Clear the SessionTicket with this Ground peer,
* Execute a DTLS Handshake procedure with this Ground peer.

*Note: For example, a DTLS Resumption procedure can fail if DTLS\_MAX\_NEGO\_ DURATION is reached, if one of the peers do not have SessionTicket …*

* + - 1. MIC Resynchronization procedure

The MIC Resynchronization procedure can be either:

* Initiated by local IOA DTLS when detecting a MIC check failure. In this case, it is local IOA DTLS that will initiate the MIC Resynchronization procedure with the peer IOA DTLS, or
* Initiated by the peer IOA DTLS. In this case, local IOA DTLS receives a DTLS encrypted message from the peer IOA DTLS initiating the MIC Resynchronization procedure.

A symmetric MIC Resynchronization procedure would allow a crisscross case (both sides initiating a MIC resynchronization at the same time). As a result, the MIC Resynchronization procedure is Air/Ground asymmetric.

Note that a MIC resynchronization procedure can happen at any time during IPv6 traffic exchange, when either of the peers detects a MIC error. This means IOA Segmentation can receive DTLS traffic (Sec bit set to 0 in IOA segment header) in the flow of IPv6 traffic (Sec bit set to 1 in IOA segment header).

Also note that MIC Resynchronization procedure messages are exchanged between Air / Ground peers as encrypted messages in a DTLS active session. MIC Resynchronization procedure can be interrupted by a DTLS Authentication request (it is higher priority – as negotiating new MIC Sequence Numbers is of no meaning when the Key is no longer valid and a DTLS authentication is necessary to get new Key …).

**REQ-IOA-078**

A DTLS negotiation (whether a full DTLS Handshake or a DTLS Resumption) **shall** have priority over MIC Resynchronization procedure.

*Note: In this case, IOA DTLS internal data are reset to forget the MIC resynchronization.*

**REQ-IOA-079**

If MIC Resynchronization procedure fails, IOA DTLS **shall** attempt a DTLS negotiation (whether a full DTLS Handshake or a DTLS Resumption depending on SessionTicket availability).

*Note: It is a local design choice to fallback to DTLS Handshake or a DTLS Resumption procedure immediately or after several successive MIC Resynchronization procedure failures.*

*Note: For example, a MIC Resynchronization procedure can fail if DTLS\_MAX\_RESYNC\_DURATION is reached, if one of the peers lost its DTLS session context …*

* + - * 1. MIC Resynchronization specific messages format

The MIC Resynchronization messages will be exchanged in the DTLS active session. As a result, they will be DTLS encrypted messages (encrypted before sending to IOA Segmentation and decrypted upon reception from IOA Segmentation). Each of the messages to be exchanged as encrypted in the DTLS session are associated with a key tag (specific value of the first byte that allow to determine the purpose of the decrypted message).

There are 3 different MIC Resynchronization messages, whose use is presented in next sections.

The **Ground MIC Resynchronization request** message, a 1 byte long message (only embedding the key tag byte) allowing the Ground to report a MIC check failure to the Aircraft and requesting a start of the MIC Resynchronization procedure:



Figure 16 – Ground MIC Resynchronization Request message

The **Aircraft MIC Resynchronization request** message, a 2 bytes long message (embedding the key tag byte, and the new value to be used as MIC sequence number basis for both uplink and downlink MIC sequence number) allowing to Aircraft to either:

* Answer to a Ground MIC Resynchronization request, or
* Report to the Ground a MIC problem detected at Aircraft side.

It is the Aircraft that will always propose the new MIC Sequence number value for both sides to resynchronize with.

The proposed new MIC sequence number value is only 1 byte long even if the uplink and downlink MIC sequence number values are 6 bytes long. Security analysis shown that it is not necessary to exchange a 6 bytes value. Simply negotiating the lower byte value and forcing both side the 5 upper bytes to 0 is sufficient. The same way, security analysis shown that applying the same negotiated MIC sequence number to both uplink and downlink sides is acceptable.



Figure 17 – Aircraft MIC Resynchronization Request message

At last, the **Ground MIC Resynchronization response** message, a 2 bytes long message (embedding the key tag byte, and the value it received from Aircraft as MIC sequence number basis for both uplink and downlink MIC sequence number) allowing the Ground to answer (and confirm new MIC Sequence number value) to the Aircraft MIC Resynchronization Request message.



Figure 18 – Ground MIC Resynchronization Response message

* + - * 1. MIC Resynchronization on Aircraft request

The following figure presents the Aircraft initiated MIC Resynchronization sequence:



Figure 19 – MIC Resynchronization sequence on Aircraft request

**REQ-IOA-080**

When Aircraft IOA DTLS detects a MIC check failure, Aircraft IOA DTLS **shall:**

* Command Aircraft IOA Security to enter Standby state,
* Inform IOA Segmentation MIC Resynchronization procedure in progress,
* Generate a random 1-byte MIC Sequence Number value,
* Send an Aircraft MIC Resynchronization Request message to report MIC Resynchronization procedure in progress and the new MIC sequence number value to the Ground peer.

**REQ-IOA-081**

When Ground IOA DTLS receives a MIC Resynchronization Request message from Aircraft, Ground IOA DTLS **shall:**

* Configure its 6 bytes uplink and downlink MIC sequence numbers with the 5 upper bytes set to 0 and the lower byte set to the newly received byte MIC sequence number value from Aircraft,
* Send a Ground MIC Resynchronization Response message embedding the MIC sequence number value received in the Aircraft Resynchronization Request message (Ground repeating it in its response message as a confirmation).

**REQ-IOA-082**

When Aircraft IOA DTLS receives a MIC Resynchronization Response message from Ground embedding the MIC sequence number previously sent to the Ground in Aircraft MIC Resynchronization Request, Aircraft IOA DTLS **shall:**

* Configure its 6 bytes uplink and downlink MIC sequence numbers with the 5 upper bytes set to 0 and the lower byte set to the negotiated MIC sequence number value,
* Command Aircraft IOA Security to enter Active state,
* Inform IOA Segmentation that MIC Resynchronization procedure is finished.

*Note: Both Air and Ground sides will now use the newly negotiated MIC Sequence Number as basis for the uplink and downlink MIC Sequence Number value.*

The Figure 19 – MIC Resynchronization sequence on Aircraft request introduces the Dropping period. As already specified in 3.2.3 IOA Segmentation function, 3.2.4 IOA Security function and in above requirements, IOA DTLS informs IOA Segmentation and IOA Security of DTLS Handshake, DTLS Resumption or MIC Resynchronization procedure execution, so that IOA Segmentation can drop any sending or receiving IPv6 traffic and IOA Security can stop managing IPv6 traffic during those procedures. This is what is represented in the above figure as the dropping period.

* + - * 1. MIC Resynchronization on Ground request

The following figure presents the Ground initiated MIC Resynchronization sequence:



Figure 20 – MIC Resynchronization sequence on Ground request

**REQ-IOA-083**

When it detects a MIC check failure, Ground IOA DTLS **shall:**

* Command Ground IOA Security to enter Standby state,
* Inform IOA Segmentation that MIC Resynchronization procedure in progress,
* Send a Ground MIC Resynchronization Request message compatible with the above figure to request a MIC Resynchronization to the Aircraft.

**REQ-IOA-084**

When Aircraft IOA DTLS receives a MIC Resynchronization Request message from Ground, Aircraft IOA DTLS **shall:**

* Command Aircraft IOA Security to enter Standby state,
* Generate a random 1-byte MIC Sequence Number value,
* Send an Aircraft MIC Resynchronization Request message to report MIC Resynchronization procedure in progress and the new MIC sequence number value to the Ground peer.

**REQ-IOA-085**

When Ground IOA DTLS receives a MIC Resynchronization Request message from Aircraft, Ground IOA DTLS **shall:**

* Configure its 6 bytes uplink and downlink MIC sequence numbers with the 5 upper bytes set to 0 and the lower byte set to the newly received byte MIC sequence number value from Aircraft,
* Send a Ground MIC Resynchronization Response message embedding the MIC sequence number value received in the Aircraft Resynchronization Request message (Ground repeating it in its response message as a confirmation),
* Command Ground IOA Security to enter Active state.

**REQ-IOA-086**

When Aircraft IOA DTLS receives a MIC Resynchronization Response message from Ground embedding the MIC sequence number previously sent to Ground in Aircraft MIC Resynchronization Request, Aircraft IOA DTLS **shall:**

* Configure its 6 bytes uplink and downlink MIC sequence numbers with the 5 upper bytes set to 0 and the lower byte set to the negotiated MIC sequence number value,
* Command IOA Security to enter Active state,
* Inform IOA Segmentation MIC Resynchronization procedure is finished.

*Note: Both Air and Ground sides will now use the newly negotiated MIC Sequence Number as basis for the uplink and downlink MIC Sequence Number value.*

The Figure 20 – MIC Resynchronization sequence on Ground request introduces the Dropping period. As already specified in 3.2.3 IOA Segmentation function, 3.2.4 IOA Security function and in above requirements, IOA DTLS informs IOA Segmentation and IOA Security of DTLS Handshake, DTLS Resumption or MIC Resynchronization procedure execution, so that IOA Segmentation can drop any sending or receiving IPv6 traffic and IOA Security can stop managing IPv6 traffic during those procedures. This is what is represented in the above figure as the dropping period.

* + - 1. IOA DTLS function robustness

The case of the Ground DTLS Server having lost its DTLS connection while Aircraft DTLS Client still have it active is to be discussed … How can the Ground DTLS Server inform the Aircraft DTLS Client a new DTLS session negotiation is needed ... It is the same problem than at DTLS / IPv6 level … But the need is not the same at VDL mode 2 level as the DTLS session directly negotiated between Ground provider gateway and Aircraft (point to point environment).