ippm
Internet-Draft
Intended status: Standards Track
Expires: 16 July 2026

J. Iurman
University of Liege
T. Zhou
Huawei

12 January 2026

A YANG Data Model for In Situ Operations, Administration, and Maintenance (IOAM) Integrity Protected Options
draft-ietf-ippm-ioam-integrity-yang-05

Abstract

In Situ Operations, Administration, and Maintenance (IOAM) is an example of an on-path hybrid measurement method to collect operational and telemetry information. The collected data may then be exported to systems that will use it to, e.g., monitor, measure, or (re)configure the network. Integrity Protection of In Situ Operations, Administration, and Maintenance (IOAM) Data Fields (RFC YYYY) defines IOAM Options with integrity protection, also called Integrity Protected Options. This document defines a YANG module for the management of these Integrity Protected Options.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 16 July 2026.

Copyright Notice

Table of Contents

1. Introduction

In Situ Operations, Administration, and Maintenance (IOAM) is an example of an on-path hybrid measurement method [RFC7799] to collect operational and telemetry information. The collected data may then be exported to systems that will use it to, e.g., monitor, measure, or (re)configure the network. [I-D.ietf-ippm-ioam-data-integrity] defines IOAM Options with integrity protection, also called Integrity

Protected Options. This document defines a data model for the management of these Integrity Protected Options using the YANG data modeling language [RFC7950]. This YANG data model supports four IOAM Integrity Protected Options, which are as follows:

* Integrity Protected Incremental Trace-Option (Section 4.1 of [I-D.ietf-ippm-ioam-data-integrity])
* Integrity Protected Pre-allocated Trace-Option (Section 4.1 of [I-D.ietf-ippm-ioam-data-integrity])
* Integrity Protected Proof of Transit (POT) Option (Section 4.2 of [I-D.ietf-ippm-ioam-data-integrity])
* Integrity Protected Edge-to-Edge (E2E) Option (Section 4.3 of [I-D.ietf-ippm-ioam-data-integrity])

1.1. Editorial Note (to be removed by RFC Editor)

Note to the RFC Editor: this section is to be removed prior to publication.

This document contains placeholder values that need to be replaced with finalized values at the time of publication. This note summarizes all of the substitutions that are needed. No other RFC Editor instructions are specified elsewhere in this document, except in Appendix A.

Please apply the following replacements:

* XXXX --> the assigned RFC number for this document
* YYYY --> the assigned RFC number for [I-D.ietf-ippm-ioam-data-integrity]
* 2026-01-12 --> the actual date of the publication of this document

2. Conventions used in this document

2.1. Abbreviations

Abbreviations used in this document:

OAM: Operations, Administration, and Maintenance
IOAM: In Situ OAM
POT: Proof of Transit
E2E: Edge to Edge

2.2. Terminology

The following terms are defined in [RFC7950] and are used in this specification:

* augment
* data model
* data node

The terminology for describing YANG data models is found in [RFC7950].

2.3. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

3. Design of the IOAM Integrity YANG Data Model

3.1. Overview

The IOAM Integrity model is organized as a list of profiles, as shown in Figure 1. In this model, the "int" prefix refers to "INTegrity protection".

```
module: ietf-ioam-integrity

  augment /ioam:ioam/ioam:profiles/ioam:profile:
    +--rw int-incremental-tracing-profile! {int-incremental-trace}?
    |  ...
    +--rw int-preallocated-tracing-profile! {int-preallocated-trace}?
    |  ...
    +--rw int-pot-profile! {int-proof-of-transit}?
    |  ...
    +--rw int-e2e-profile! {int-edge-to-edge}?
```

Figure 1: Overview of the IOAM Integrity model

This document defines augmentations to the "ietf-ioam" YANG module [RFC9617] by adding integrity-related profiles. Each profile is associated with one flow and the corresponding IOAM information. These integrity-related profiles are indicated by four defined features, i.e., "int-incremental-trace", "int-preallocated-trace", "int-proof-of-transit", and "int-edge-to-edge". The structures of the new profiles follow what is defined in [RFC9617], but for distinct purposes.

3.2. Integrity Protected Pre-allocated Tracing Profile

As illustrated in Figure 2, the "int-preallocated-tracing-profile" container provides the detailed information for the pre-allocated tracing data with integrity protection. This information has the same structure as the Pre-allocated Tracing Profile (Section 3.2 of [RFC9617]), but has the following additional data node:

int-method: indicates which Integrity Protection Method is used, as defined in the "IOAM Integrity Protection Methods" IANA registry [IANA-IOAM]. It is only defined at the encapsulating node because the Integrity Protection Method is selected and initialized when IOAM data is encapsulated.

```
+--rw int-preallocated-tracing-profile! {int-preallocated-trace}?
   +--rw node-action?   ioam-node-action
   +--rw trace-types
   |  +--rw use-namespace?   ioam-namespace
   |  +--rw trace-type*      ioam-trace-type
   +--rw max-length?    uint32
   +--rw int-method?    iana-ioam-ipm:method-id
```

Figure 2: Integrity Protected Pre-allocated Tracing Profile

3.3. Integrity Protected Incremental Tracing Profile

As illustrated in Figure 3, the "int-incremental-tracing-profile" container provides the detailed information for the incremental tracing data with integrity protection. This information has the same structure as the Incremental Tracing Profile (Section 3.3 of [RFC9617]), but has the following additional data node:

int-method: indicates which Integrity Protection Method is used, as defined in the "IOAM Integrity Protection Methods" IANA registry [IANA-IOAM]. It is only defined at the encapsulating node because the Integrity Protection Method is selected and initialized when IOAM data is encapsulated.

```
+--rw int-incremental-tracing-profile! {int-incremental-trace}?
   +--rw node-action?   ioam-node-action
   +--rw trace-types
   |  +--rw use-namespace?   ioam-namespace
   |  +--rw trace-type*      ioam-trace-type
   +--rw max-length?    uint32
   +--rw int-method?    iana-ioam-ipm:method-id
```

Figure 3: Integrity Protected Incremental Tracing Profile

3.4. Integrity Protected Proof of Transit Profile

As illustrated in Figure 4, the "int-pot-profile" container is intended to provide the detailed information for the proof of transit data with integrity protection. This information has the same structure as the Proof of Transit Profile (Section 3.5 of [RFC9617]), but has the following additional data nodes:

node-action: imported from the "ietf-ioam" YANG module [RFC9617] with the same definition.

int-method: indicates which Integrity Protection Method is used, as defined in the "IOAM Integrity Protection Methods" IANA registry [IANA-IOAM]. It is only defined at the encapsulating node because the Integrity Protection Method is selected and initialized when IOAM data is encapsulated.

```
+--rw int-pot-profile! {int-proof-of-transit}?
   +--rw use-namespace?   ioam:ioam-namespace
   +--rw pot-type?        ioam:ioam-pot-type
   +--rw node-action?     ioam:ioam-node-action
   +--rw int-method?      iana-ioam-ipm:method-id
```

Figure 4: Integrity Protected Proof of Transit Profile

3.5. Integrity Protected Edge-to-Edge Profile

As illustrated in Figure 5, the "int-e2e-profile" container provides the detailed information for the edge-to-edge data with integrity protection. This information has the same structure as the Edge-to-Edge Profile (Section 3.6 of [RFC9617]), but has the following additional data node:

int-method: indicates which Integrity Protection Method is used, as defined in the "IOAM Integrity Protection Methods" IANA registry [IANA-IOAM]. It is only defined at the encapsulating node because the Integrity Protection Method is selected and initialized when IOAM data is encapsulated.

```
+--rw int-e2e-profile! {int-edge-to-edge}?
   +--rw node-action?   ioam-node-action
   +--rw e2e-types
   |  +--rw use-namespace?   ioam-namespace
   |  +--rw e2e-type*        ioam-e2e-type
   +--rw int-method?    iana-ioam-ipm:method-id
```

Figure 5: Integrity Protected Edge-to-Edge Profile

4. IOAM Integrity YANG Module

The "ietf-ioam-integrity" module defined in this document imports the "ietf-ioam" module defined in [RFC9617]. This document also references [I-D.ietf-ippm-ioam-data-integrity].

<CODE BEGINS> file "ietf-ioam-integrity@2026-01-12.yang"

```
module ietf-ioam-integrity {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ioam-integrity";
  prefix ioam-int;

  import ietf-ioam {
    prefix ioam;
    reference
      "RFC 9617: A YANG Data Model for In Situ Operations,
                 Administration, and Maintenance (IOAM)";
  }

  import iana-ioam-integrity-protection-methods {
    prefix iana-ioam-ipm;
    reference
      "RFC XXXX: A YANG Data Model for In Situ Operations,
                 Administration, and Maintenance (IOAM)
                 Integrity Protected Options";
  }

  organization
    "IETF IPPM (IP Performance Measurement) Working Group";

  contact
    "WG Web:   <https://datatracker.ietf.org/wg/ippm>
     WG List:  <mailto:ippm@ietf.org>

     Editor:   Tianran Zhou
               <mailto:zhoutianran@huawei.com>
     Editor:   Justin Iurman
               <mailto:justin.iurman@uliege.be>";

  description
    "This YANG module specifies a vendor-independent data model for
     In Situ Operations, Administration, and Maintenance (IOAM)
     Integrity Protected Options.

     Copyright (c) 2026 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
```

```
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Revised BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (https://trustee.ietf.org/license-info).

        All revisions of IETF and IANA published modules can be found
        at the YANG Parameters registry group
        (https://www.iana.org/assignments/yang-parameters).

        This version of this YANG module is part of RFC XXXX; see the
        RFC itself for full legal notices.

        Operational Considerations:

         - This model does not treat as mutually exclusive an IOAM
           Option and its integrity protected equivalent (i.e., an IOAM
           Integrity Protected Option). For example, an implementation
           may support the simultaneous configuration of an
           IOAM-Namespace with the Pre-allocated Trace Option and
           another IOAM-Namespace with the Integrity Protected
           Pre-allocated Trace Option. Therefore, the model does not
           impose constraints that would prevent such use cases.";

      revision 2026-01-12 {
        description
          "Initial version.";
        reference
          "RFC XXXX: A YANG Data Model for In Situ Operations,
                     Administration, and Maintenance (IOAM)
                     Integrity Protected Options";
      }

      /*
       * FEATURES
       */

      feature int-incremental-trace {
        description
          "This feature indicates that the Integrity Protected
           Incremental Trace-Option is supported.";
        reference
          "RFC YYYY: Integrity Protection of In Situ Operations,
                     Administration, and Maintenance (IOAM)
                     Data Fields, Section 4.1";
      }

      feature int-preallocated-trace {
```

```
    description
      "This feature indicates that the Integrity Protected
       Pre-allocated Trace-Option is supported.";
    reference
      "RFC YYYY: Integrity Protection of In Situ Operations,
                 Administration, and Maintenance (IOAM)
                 Data Fields, Section 4.1";
  }

  feature int-proof-of-transit {
    description
      "This feature indicates that the Integrity Protected Proof of
       Transit Option is supported.";
    reference
      "RFC YYYY: Integrity Protection of In Situ Operations,
                 Administration, and Maintenance (IOAM)
                 Data Fields, Section 4.2";
  }

  feature int-edge-to-edge {
    description
      "This feature indicates that the Integrity Protected
       Edge-to-Edge Option is supported.";
    reference
      "RFC YYYY: Integrity Protection of In Situ Operations,
                 Administration, and Maintenance (IOAM)
                 Data Fields, Section 4.3";
  }

  /*
   * GROUP DEFINITIONS
   */

  grouping int-method-grouping {
    description
      "A grouping for Integrity Protection Methods.";
    leaf int-method {
      when "derived-from-or-self(../node-action,
            'ioam:action-encapsulate')";
      type iana-ioam-ipm:method-id;
      description
        "This object indicates the Integrity Protection Method for
         this profile. 'int-method' is only defined at the
         encapsulating node.";
    }
  }

  /*
```

```
      * DATA NODES
      */

     augment "/ioam:ioam/ioam:profiles/ioam:profile" {
       description
         "This augmentation adds 4 profiles for the Integrity Protected
          Options.";
       container int-incremental-tracing-profile {
         if-feature "int-incremental-trace";
         presence "Enables the Integrity Protected
                   Incremental Trace-Option.";
         description
           "This container describes the profile for the Integrity
            Protected Incremental Trace-Option.";
         uses ioam:ioam-incremental-tracing-profile;
         uses int-method-grouping;
       }
       container int-preallocated-tracing-profile {
         if-feature "int-preallocated-trace";
         presence "Enables the Integrity Protected
                   Pre-allocated Trace-Option.";
         description
           "This container describes the profile for the Integrity
            Protected Pre-allocated Trace-Option.";
         uses ioam:ioam-preallocated-tracing-profile;
         uses int-method-grouping;
       }
       container int-pot-profile {
         if-feature "int-proof-of-transit";
         presence "Enables the Integrity Protected
                   Proof of Transit Option.";
         description
           "This container describes the profile for the Integrity
            Protected Proof of Transit Option.";
         leaf use-namespace {
           type ioam:ioam-namespace;
           default "ioam:default-namespace";
           description
             "This object indicates the namespace used for the
              POT types.";
         }
         leaf pot-type {
           type ioam:ioam-pot-type;
           description
             "The type of a particular POT variant that specifies
              the POT data that is included.";
         }
         leaf node-action {
```

```
            type ioam:ioam-node-action;
            default "ioam:action-transit";
            description
              "This object indicates the action the node needs to
               take, e.g., encapsulation.";
          }
          uses int-method-grouping;
        }
        container int-e2e-profile {
          if-feature "int-edge-to-edge";
          presence "Enables the Integrity Protected
                    Edge-to-Edge Option.";
          description
            "This container describes the profile for the Integrity
             Protected Edge-to-Edge Option.";
          uses ioam:ioam-e2e-profile;
          uses int-method-grouping;
        }
      }
    }
    <CODE ENDS>
```

5. Security Considerations

This section is modeled after the template described in Section 3.7 of [I-D.ietf-netmod-rfc8407bis].

The "ietf-ioam-integrity" YANG module defines a data model that is designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. These YANG-based management protocols (1) have to use a secure transport layer (e.g., SSH [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and (2) have to use mutual authentication.

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., "config true", which is the default). All writable data nodes are likely to be sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) and delete operations to these data nodes without proper protection or authentication can have a negative effect on network operations. The following subtrees and data nodes have particular sensitivities/vulnerabilities:

/ioam:ioam/ioam:profiles/ioam:profile: The entries in the "profile" list include the whole IOAM profile configurations. Unexpected changes to these entries could lead to incorrect IOAM behavior for the corresponding flows. Consequently, such changes would impact performance monitoring, data analytics, and associated interactions with network services. Also, unauthorized access to integrity-related parameters may impact the integrity protection service, thus preventing the interpretation and validation of IOAM data.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. Specifically, the following subtrees and data nodes have particular sensitivities/ vulnerabilities:

/ioam:ioam/ioam:profiles/ioam:profile: The information contained in this subtree might reveal information about the services deployed for customers. For instance, a customer might be given access to monitor the status of their services. In this scenario, the customer’s access should be restricted to nodes representing their services so as not to divulge information about the underlying network structure or services.

This YANG module uses groupings from other YANG modules that define nodes that may be considered sensitive or vulnerable in network environments. Refer to the Security Considerations of [RFC9617] for information as to which nodes may be considered sensitive or vulnerable in network environments.

6. IANA Considerations

IANA is requested to register the following URIs in the "ns" registry within the "IETF XML Registry" group [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-ioam-integrity
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

URI:
urn:ietf:params:xml:ns:yang:iana-ioam-integrity-protection-methods
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

IANA is requested to register the following YANG modules in the "YANG Module Names" registry [RFC6020] within the "YANG Parameters" registry group:

Name: ietf-ioam-integrity
Maintained by IANA?: N
Namespace: urn:ietf:params:xml:ns:yang:ietf-ioam-integrity
Prefix: ioam-int
Reference: RFC XXXX

Name: iana-ioam-integrity-protection-methods
Maintained by IANA?: Y
Namespace:
urn:ietf:params:xml:ns:yang:iana-ioam-integrity-protection-methods
Prefix: iana-ioam-ipm
Reference: RFC XXXX

6.1. IANA-Maintained "iana-ioam-integrity-protection-methods" Module

This document defines the initial version of the IANA-maintained "iana-ioam-integrity-protection-methods" YANG module. The most recent version of the YANG module is available from the "YANG Parameters" registry group [IANA-YANG-PARAMETERS].

IANA is requested to add this note to the registry:

| New values must not be directly added to the "iana-ioam-integrity-
| protection-methods" YANG module. They must instead be added to
| the "IOAM Integrity Protection Methods" registry.

When a value is added to the "IOAM Integrity Protection Methods" registry, a new "enum" statement must be added to the "iana-ioam-integrity-protection-methods" YANG module. The "enum" statement, and sub-statements thereof, should be defined:

"enum": Prefix "method-" to the decimal value of the ID from the registry.
"value": Contains the decimal value of the IANA-assigned ID value.
"status": Is included only if a registration has been deprecated or obsoleted. IANA "deprecated" maps to YANG status "deprecated", and IANA "obsolete" maps to YANG status "obsolete".
"description": Replicates the description from the registry.
"reference": Replicates the reference(s) from the registry with the title of the document(s) added.

Unassigned or reserved values are not present in the module.

When the "iana-ioam-integrity-protection-methods" YANG module is updated, a new "revision" statement with a unique revision date needs to be added in front of the existing revision statements.

IANA is requested to add this note to [reference-to-the-iana-foo-registry]:

| When this registry is modified, the YANG module "iana-ioam-integrity-protection-methods" [IANA_FOO_URL] must be updated as defined in RFC XXXX.

7. Acknowledgements

The authors would like to thank Alex Huang Feng, Will Hawkins, and Mohamed Boucadair for their valuable feedback.

8. References

8.1. Normative References

[I-D.ietf-ippm-ioam-data-integrity]
Brockners, F., Bhandari, S., Mizrahi, T., and J. Iurman, "Integrity Protection of In Situ Operations, Administration, and Maintenance (IOAM) Data Fields", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-data-integrity-15, 1 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-data-integrity-15>.

[IANA-IOAM]
"In Situ OAM (IOAM)", n.d., <https://www.iana.org/assignments/ioam/ioam.xhtml>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC9617] Zhou, T., Ed., Guichard, J., Brockners, F., and S. Raghavan, "A YANG Data Model for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9617, DOI 10.17487/RFC9617, August 2024, <https://www.rfc-editor.org/info/rfc9617>.

8.2. Informative References

[I-D.ietf-netmod-rfc8407bis] Bierman, A., Boucadair, M., and Q. Wu, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", Work in Progress, Internet-Draft, draft-ietf-netmod-rfc8407bis-28, 5 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-netmod-rfc8407bis-28>.

[IANA-YANG-PARAMETERS] "YANG Parameters", n.d., <https://www.iana.org/assignments/yang-parameters>.

[RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <https://www.rfc-editor.org/info/rfc4252>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

Appendix A. Initial Version of the IOAM Integrity Protection Methods IANA-Maintained Module

RFC Ed.: please remove this section.

```
<CODE BEGINS>
  file "iana-ioam-integrity-protection-methods@2026-01-12.yang"
module iana-ioam-integrity-protection-methods {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:iana-ioam-integrity-protection-methods";
  prefix iana-ioam-ipm;

  organization
    "Internet Assigned Numbers Authority (IANA)";

  contact
    "Internet Assigned Numbers Authority

     ICANN
     12025 Waterfront Drive, Suite 300
     Los Angeles, CA 90094

     Tel: +1 424 254 5300

     <mailto:iana@iana.org>";

  description
    "This YANG module is maintained by IANA and reflects
     the 'IOAM Integrity Protection Methods' registry.

     Copyright (c) 2026 IETF Trust and the persons
     identified as authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Revised BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).
```

```
        All revisions of IETF and IANA published modules can be found
        at the YANG Parameters registry group
        (https://www.iana.org/assignments/yang-parameters).

        The initial version of this YANG module is part of RFC XXXX;
        see the RFC itself for full legal notices.

      // RFC Ed.: replace the IANA_FOO_URL and remove this note

        The latest version of this YANG module is available at
        <IANA_FOO_URL>.";

      reference
        "IOAM Integrity Protection Methods
         (https://www.iana.org/assignments/ioam/ioam.xhtml)";

      revision 2026-01-12 {
        description
          "Initial version.";
        reference
          "URL of the published initial version of the module

      // RFC Ed.: replace above with the URL of the module
      //          and remove this note

           RFC XXXX: A YANG Data Model for In Situ Operations,
                     Administration, and Maintenance (IOAM)
                     Integrity Protected Options";
      }

      /*
       * Type definitions
       */

      typedef method-id {
        type enumeration {
          enum method-0 {
            value 0;
            description
              "IPM 0: AES-GMAC, 16-octet (full) Authentication Tag,
                       12-octet Initialization Vector.";
            reference
              "RFC YYYY: Integrity Protection of In Situ Operations,
                         Administration, and Maintenance (IOAM)
                         Data Fields, Section 5";
          }
        }
        description
```

```
        "Identifier for an IOAM Integrity Protection Method,
         matching the 'IOAM Integrity Protection Methods'
         IANA registry.";
      reference
        "https://www.iana.org/assignments/ioam/ioam.xhtml";
    }
  }
  <CODE ENDS>
```

Appendix B. Full tree of the IOAM Integrity YANG Data Model

Figure 6 illustrates the full tree of the IOAM Integrity YANG Data Model.

```
module: ietf-ioam-integrity

  augment /ioam:ioam/ioam:profiles/ioam:profile:
    +--rw int-incremental-tracing-profile! {int-incremental-trace}?
    |  +--rw node-action?   ioam-node-action
    |  +--rw trace-types
    |  |  +--rw use-namespace?   ioam-namespace
    |  |  +--rw trace-type*      ioam-trace-type
    |  +--rw max-length?    uint32
    |  +--rw int-method?    iana-ioam-ipm:method-id
    +--rw int-preallocated-tracing-profile! {int-preallocated-trace}?
    |  +--rw node-action?   ioam-node-action
    |  +--rw trace-types
    |  |  +--rw use-namespace?   ioam-namespace
    |  |  +--rw trace-type*      ioam-trace-type
    |  +--rw max-length?    uint32
    |  +--rw int-method?    iana-ioam-ipm:method-id
    +--rw int-pot-profile! {int-proof-of-transit}?
    |  +--rw use-namespace?   ioam:ioam-namespace
    |  +--rw pot-type?        ioam:ioam-pot-type
    |  +--rw node-action?     ioam:ioam-node-action
    |  +--rw int-method?      iana-ioam-ipm:method-id
    +--rw int-e2e-profile! {int-edge-to-edge}?
       +--rw node-action?   ioam-node-action
       +--rw e2e-types
       |  +--rw use-namespace?   ioam-namespace
       |  +--rw e2e-type*        ioam-e2e-type
       +--rw int-method?    iana-ioam-ipm:method-id

  grouping int-method-grouping:
    +-- int-method?   iana-ioam-ipm:method-id
```

Figure 6: Full tree of the IOAM Integrity YANG Data Model

Appendix C. Example of the Integrity Protected Incremental Tracing Profile

A JSON [RFC7951] example of the Integrity Protected Incremental Tracing Profile is depicted in Figure 7. This configuration is received by an IOAM ingress node. This node encapsulates the IOAM data in the IPv6 Hop-by-Hop option header. The Integrity Protection Method to be used is method 0. The trace type indicates that each on-path node needs to capture the transit delay and add the data to the IOAM node data list. The incremental tracing data space is variable; however, the node data list must not exceed 512 bytes.

```
{
  "ietf-ioam:ioam": {
    "admin-config": {
      "enabled": true
    },
    "profiles": {
      "profile": [
        {
          "profile-name": "ietf-test-profile",
          "protocol-type": "ietf-ioam:ipv6",
          "ietf-ioam-integrity:int-incremental-tracing-profile": {
            "node-action": "ietf-ioam:action-encapsulate",
            "trace-types": {
              "use-namespace": "ietf-ioam:default-namespace",
              "trace-type": [
                "ietf-ioam:trace-transit-delay"
              ]
            },
            "max-length": 512,
            "int-method": "method-0"
          }
        }
      ]
    }
  }
}
```

Figure 7: JSON-encoded Integrity Protected Incremental Tracing Profile

Appendix D. Example of the Integrity Protected Pre-allocated Tracing Profile

A JSON [RFC7951] example of the Integrity Protected Pre-allocated Tracing Profile is depicted in Figure 8. This configuration is received by an IOAM ingress node. This node first identifies the target flow by using the ACL parameter "test-acl" and then encapsulates the IOAM data in the NSH. The Integrity Protection Method to be used is method 0. The trace type indicates that each on-path node needs to capture the namespace-specific data in short format and add the data to the IOAM node data list. This node pre-allocates the node data list in the packet with 512 bytes.

```
{
  "ietf-ioam:ioam": {
    "admin-config": {
      "enabled": true
    },
    "profiles": {
      "profile": [
        {
          "profile-name": "ietf-test-profile",
          "filter": {
            "filter-type": "ietf-ioam:acl-filter",
            "ace-name": "test-acl"
          },
          "protocol-type": "ietf-ioam:nsh",
          "ietf-ioam-integrity:int-preallocated-tracing-profile": {
            "node-action": "ietf-ioam:action-encapsulate",
            "trace-types": {
              "use-namespace": "ietf-ioam:default-namespace",
              "trace-type": [
                "ietf-ioam:trace-namespace-data"
              ]
            },
            "max-length": 512,
            "int-method": "method-0"
          }
        }
      ]
    }
  }
}
```

Figure 8: JSON-encoded Integrity Protected Pre-allocated Tracing Profile

Appendix E. Example of the Integrity Protected Proof of Transit and Integrity Protected Edge-to-Edge Profiles

A JSON [RFC7951] example of the Integrity Protected Proof of Transit Profile, combined with the Integrity Protected Edge-to-Edge Profile, is depicted in Figure 9. This configuration is received by an IOAM ingress node. This node encapsulates the Integrity Protected Proof of Transit Type 0 in an IPv6 Hop-by-Hop Options header, and also encapsulates the Integrity Protected Edge-to-Edge in an IPv6 Destination Options header. The Edge-to-Edge type indicates the presence of a 64-bit sequence number. The Integrity Protection Method to be used for both is method 0.

```
{
  "ietf-ioam:ioam": {
    "admin-config": {
      "enabled": true
    },
    "profiles": {
      "profile": [
        {
          "profile-name": "ietf-test-profile-pot",
          "protocol-type": "ietf-ioam:ipv6",
          "ietf-ioam-integrity:int-pot-profile": {
            "pot-type": "ietf-ioam:pot-type-0",
            "node-action": "ietf-ioam:action-encapsulate",
            "int-method": "method-0"
          }
        },
        {
          "profile-name": "ietf-test-profile-e2e",
          "protocol-type": "ietf-ioam:ipv6",
          "ietf-ioam-integrity:int-e2e-profile": {
            "node-action": "ietf-ioam:action-encapsulate",
            "e2e-types": {
              "use-namespace": "ietf-ioam:default-namespace",
              "e2e-type": [
                "ietf-ioam:e2e-seq-num-64"
              ]
            },
            "int-method": "method-0"
          }
        }
      ]
    }
  }
}
```

Figure 9: JSON-encoded Integrity Protected Proof of Transit and Integrity Protected Edge-to-Edge Profiles

Authors' Addresses

Justin Iurman
University of Liege
10, Allee de la decouverte (B28)
4000 Sart-Tilman
Belgium
Email: justin.iurman@uliege.be

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing
100095
China
Email: zhoutianran@huawei.com