ippm
Internet-Draft
Intended status: Standards Track
Expires: 17 May 2026

J. Iurman
University of Liege
T. Zhou
Huawei

13 November 2025

A YANG Data Model for In Situ Operations, Administration, and Maintenance (IOAM) Integrity Protected Options
draft-ietf-ippm-ioam-integrity-yang-00

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. I-D.ietf-ippm-ioam-data-integrity (RFC Ed.: to be replaced by RFC YYYY) defines IOAM Options with integrity protection, also called Integrity Protected Options. This document defines a YANG module for the configuration 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 17 May 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 configuration of these Integirty 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 [I-D.ietf-ippm-ioam-data-integrity]
* Integrity Protected Pre-allocated Trace-Option [I-D.ietf-ippm-ioam-data-integrity]
* Integrity Protected Proof of Transit (POT) Option [I-D.ietf-ippm-ioam-data-integrity]
* Integrity Protected Edge-to-Edge (E2E) Option [I-D.ietf-ippm-ioam-data-integrity]

2. Conventions

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 the following figure.

```
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}?
```

This document uses the "ietf-ioam" YANG module [RFC9617] and augments its definition of a profile. The supported profiles are indicated by four defined features, i.e., "int-incremental-trace", "int-preallocated-trace", "int-proof-of-transit", and "int-edge-to-edge" (i.e., "int" prefix for "INTegrity protection"). Although these four new profiles resemble those defined in [RFC9617], they are distinct profiles since they represent different IOAM Option-Type code points.

The YANG data model in this document conforms to the Network Management Datastore Architecture (NMDA) defined in [RFC8342].

3.2. Integrity Protected Pre-allocated Tracing Profile

The "int-preallocated-tracing-profile" parameter contains the detailed information for the pre-allocated tracing data with integrity protection. This information is the same as for the Pre-allocated Tracing Profile; see [RFC9617], Sec. 3.2. This information also includes:

int-method: indicates which Integrity Protection Method is used.

```
+--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?    method-type
```

3.3. Integrity Protected Incremental Tracing Profile

The "int-incremental-tracing-profile" parameter contains the detailed information for the incremental tracing data with integrity protection. This information is the same as for the Incremental Tracing Profile; see [RFC9617], Sec. 3.3. This information also includes:

int-method: indicates which Integrity Protection Method is used.

```
+--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?    method-type
```

3.4. Integrity Protected Proof of Transit Profile

The "int-pot-profile" parameter is intended to contain the detailed information for the proof of transit data with integrity protection. This information is the same as for the Proof of Transit Profile; see [RFC9617], Sec. 3.5. This information also includes:

node-action: imported from the "ietf-ioam" YANG module [RFC9617] with the same definition.
int-method: indicates which Integrity Protection Method is used.

```
+--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?      method-type
```

3.5. Integrity Protected Edge-to-Edge Profile

The "int-e2e-profile" parameter contains the detailed information for the edge-to-edge data with integrity protection. This information is the same as for the Edge-to-Edge Profile; see [RFC9617], Sec. 3.6. This information also includes:

int-method: indicates which Integrity Protection Method is used.

```
+--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?    method-type
```

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@2025-11-13.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)";
  }

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

  contact
    "WG Web:   <https://datatracker.ietf.org/wg/ippm>
     WG List:  <mailto:ippm@ietf.org>
     Author:   Tianran Zhou
               <mailto:zhoutianran@huawei.com>
     Author:   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) 2025 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).

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

  revision 2025-11-13 {
    description
      "Initial revision.";
    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";
      }

      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";
      }

      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";
      }

      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";
      }

      /*
```

```
     * IDENTITIES
     */

    identity method {
      description
        "Base identity to represent the Integrity Protection Method.";
    }

    identity method-0 {
      base method;
      description
        "The Integrity Protection Method 0 uses AES-GMAC with a 12-byte
         Nonce and a 16-byte ICV.";
      reference
        "RFC YYYY: Integrity Protection of In Situ Operations,
         Administration, and Maintenance (IOAM) Data Fields";
    }

    /*
     * TYPE DEFINITIONS
     */

    typedef method-type {
      type identityref {
        base method;
      }
      description
        "It specifies the Integrity Protection Method.";
    }

    /*
     * 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;
```

```
          leaf int-method {
            when "derived-from-or-self(../node-action,
               'ioam:action-encapsulate')";
            type method-type;
            default "method-0";
            description
              "This object indicates the Integrity Protection Method for
               this profile.";
          }
        }

        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;

          leaf int-method {
            when "derived-from-or-self(../node-action,
               'ioam:action-encapsulate')";
            type method-type;
            default "method-0";
            description
              "This object indicates the Integrity Protection Method for
               this profile.";
          }
        }

        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.";
        }

        leaf int-method {
          when "derived-from-or-self(../node-action,
             'ioam:action-encapsulate')";
          type method-type;
          default "method-0";
          description
            "This object indicates the Integrity Protection Method for
             this profile.";
        }
      }

      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;

        leaf int-method {
          when "derived-from-or-self(../node-action,
             'ioam:action-encapsulate')";
          type method-type;
          default "method-0";
          description
            "This object indicates the Integrity Protection Method for
             this profile.";
        }
      }
    }
  }
```

<CODE ENDS>

5. Security Considerations

This section is modeled after the template described in Section 3.7.1 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 protocols have to use a secure transport layer (e.g., SSH [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and 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: Please refer to the Security Considerations of [RFC9617].

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: Please refer to the Security Considerations of [RFC9617].

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

RFC Ed.: In this section and in Section 4, please replace all occurrences of 'RFC XXXX' with the actual RFC number. Also in Section 4 and in the Abstract, please replace all occurrences of 'RFC YYYY' with the actual RFC number of [I-D.ietf-ippm-ioam-data-integrity] (and remove this note).

IANA is requested to assign a new URI from the "IETF XML Registry" [RFC3688]. The following URI is suggested:

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

This document also requests a new YANG module name in the "YANG Module Names" registry [RFC6020] with the following suggestion:

Name: ietf-ioam-integrity
Namespace: urn:ietf:params:xml:ns:yang:ietf-ioam-integrity
Prefix: ioam-int
Reference: RFC XXXX

7. Acknowledgements

The authors would like to thank Alex Huang Feng and Will Hawkins 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>.

[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>.

[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>.

[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>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[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>.

[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>.

[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. An Example of the Integrity Protected Incremental Tracing Profile

An example of the Integrity Protected Incremental Tracing Profile is depicted in the following figure. 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.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     message-id="101">
  <edit-config>
    <target>
      <candidate/>
    </target>
    <config>
      <ioam xmlns="urn:ietf:params:xml:ns:yang:ietf-ioam">
        <admin-config>
          <enabled>true</enabled>
        </admin-config>
        <profiles>
          <profile>
            <profile-name>ietf-test-profile</profile-name>
            <protocol-type>ipv6</protocol-type>
            <int-incremental-tracing-profile
             xmlns="urn:ietf:params:xml:ns:yang:ietf-ioam-integrity">
              <node-action>action-encapsulate</node-action>
              <trace-types>
                <use-namespace>default-namespace</use-namespace>
                <trace-type>trace-transit-delay</trace-type>
              </trace-types>
              <max-length>512</max-length>
              <int-method>method-0</int-method>
            </int-incremental-tracing-profile>
          </profile>
        </profiles>
      </ioam>
    </config>
  </edit-config>
</rpc>
```

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

An example of the Integrity Protected Pre-allocated Tracing Profile is depicted in the following figure. 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.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     message-id="101">
  <edit-config>
    <target>
      <candidate/>
    </target>
    <config>
      <ioam xmlns="urn:ietf:params:xml:ns:yang:ietf-ioam">
        <admin-config>
          <enabled>true</enabled>
        </admin-config>
        <profiles>
          <profile>
            <profile-name>ietf-test-profile</profile-name>
            <filter>
              <filter-type>acl-filter</filter-type>
              <ace-name>test-acl</ace-name>
            </filter>
            <protocol-type>nsh</protocol-type>
            <int-preallocated-tracing-profile
             xmlns="urn:ietf:params:xml:ns:yang:ietf-ioam-integrity">
              <node-action>action-encapsulate</node-action>
              <trace-types>
                <use-namespace>default-namespace</use-namespace>
                <trace-type>trace-namespace-data</trace-type>
              </trace-types>
              <max-length>512</max-length>
              <int-method>method-0</int-method>
            </int-preallocated-tracing-profile>
          </profile>
        </profiles>
      </ioam>
    </config>
  </edit-config>
</rpc>
```

Appendix C. An Example of the Integrity Protected Proof of Transit Profile

An example of the Integrity Protected Proof of Transit Profile is depicted in the following figure. This configuration is received by an IOAM ingress node. This node encapsulates the IOAM data (POT Type 0) in the IPv6 Hop-by-Hop option header. The Integrity Protection Method to be used is method 0.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     message-id="101">
  <edit-config>
    <target>
      <candidate/>
    </target>
    <config>
      <ioam xmlns="urn:ietf:params:xml:ns:yang:ietf-ioam">
        <admin-config>
          <enabled>true</enabled>
        </admin-config>
        <profiles>
          <profile>
            <profile-name>ietf-test-profile</profile-name>
            <protocol-type>ipv6</protocol-type>
            <int-pot-profile
             xmlns="urn:ietf:params:xml:ns:yang:ietf-ioam-integrity">
              <pot-type>pot-type-0</pot-type>
              <node-action>action-encapsulate</node-action>
              <int-method>method-0</int-method>
            </int-pot-profile>
          </profile>
        </profiles>
      </ioam>
    </config>
  </edit-config>
</rpc>
```

Appendix D. An Example of the Integrity Protected Edge-to-Edge Profile

An example of the Integrity Protected Edge-to-Edge Profile is depicted in the following figure. This configuration is received by an IOAM ingress node. This node encapsulates the IOAM data in the IPv6 Destination option header. The Integrity Protection Method to be used is method 0. The Edge-to-Edge type indicates the presence of a 64-bit sequence number.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     message-id="101">
  <edit-config>
    <target>
      <candidate/>
    </target>
    <config>
      <ioam xmlns="urn:ietf:params:xml:ns:yang:ietf-ioam">
        <admin-config>
          <enabled>true</enabled>
        </admin-config>
        <profiles>
          <profile>
            <profile-name>ietf-test-profile</profile-name>
            <protocol-type>ipv6</protocol-type>
            <int-e2e-profile
             xmlns="urn:ietf:params:xml:ns:yang:ietf-ioam-integrity">
              <node-action>action-encapsulate</node-action>
              <e2e-types>
                <use-namespace>default-namespace</use-namespace>
                <e2e-type>e2e-seq-num-64</e2e-type>
              </e2e-types>
              <int-method>method-0</int-method>
            </int-e2e-profile>
          </profile>
        </profiles>
      </ioam>
    </config>
  </edit-config>
</rpc>
```

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