Workgroup: ippm
Internet-Draft:
draft-ietf-ippm-ioam-data-integrity-02
Published: 5 July 2022
Intended Status: Standards Track
Expires: 6 January 2023
Authors: F. Brockners S. Bhandari T. Mizrahi J. Iurman
Cisco Thoughtspot Huawei ULiege

Integrity of In-situ OAM Data Fields

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path in the network. IETF protocols require features to ensure their security. This document describes the integrity protection of IOAM-Data-Fields.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction
2. Conventions
3. Threat Analysis
   3.1. Modification: IOAM-Data-Fields
   3.2. Modification: IOAM Option-Type Headers
   3.3. Injection: IOAM-Data-Fields
   3.4. Injection: IOAM Option-Type Headers
   3.5. Management and Exporting
   3.6. Delay
   3.7. Threat Summary
4. Integrity Protected Option-Types
   4.1. Integrity Protected Trace Option-Types
   4.2. Integrity Protected POT Option-Type
   4.3. Integrity Protected E2E Option-Type
5. Methods for space optimized integrity protection
   5.1. Symmetric key based signature
   5.2. Asymmetric key based signature
6. IANA Considerations
   6.1. IOAM Option-Type Registry
   6.2. IOAM Integrity Protection Algorithm Suite Registry
7. Security Considerations
   7.1. Replay protection
8. Acknowledgements
9. References
   9.1. Normative References
   9.2. Informative References
Authors' Addresses

1. Introduction

"In-situ" Operations, Administration, and Maintenance (IOAM) records OAM information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the OAM data is added to the data packets rather than being sent within packets specifically dedicated to OAM. IOAM is to complement mechanisms such as Ping or Traceroute. In terms of "active" or "passive" OAM, "in-situ" OAM can be considered a hybrid OAM type. "In-situ" mechanisms do not require extra packets to be sent. IOAM adds information to the already available data packets and therefore cannot be considered passive. In terms of the classification given in [RFC7799], IOAM could be portrayed as Hybrid Type I. IOAM mechanisms can be leveraged where mechanisms using, e.g., ICMP do not apply or do not offer the desired results, such as proving that a certain traffic flow takes a pre-defined path, SLA verification for the data traffic, detailed statistics on traffic distribution paths in networks that distribute traffic across multiple paths, or scenarios in which probe traffic is potentially handled differently from regular data traffic by the network devices.

[RFC9197] assumes that IOAM is deployed in limited domains, where an operator has means to select, monitor, and control the access to all the networking devices, making the domain a trusted network. As such, IOAM-Data-Fields are carried in clear within packets and there are no protections against any node or middlebox tampering with the data. IOAM-Data-Fields collected in an untrusted or semi-trusted environment require integrity protection to support critical operational decisions.

The following considerations and requirements are to be taken into account in addition to addressing the problem of detectability of any integrity breach of the IOAM-Data-Fields collected:

1. IOAM data is processed by the data plane, hence viability of any method to prove integrity of the IOAM-Data-Fields must be feasible at data plane processing/forwarding rates (IOAM might be applied to all traffic a router forwards).

2. IOAM data is carried within packets. Additional space required to prove integrity of the IOAM-Data-Fields needs to be optimal, i.e. should not exceed the MTU or have adverse effect on packet processing.

3. Replay protection of older IOAM data should be possible. Without replay protection, a rogue node can present the old IOAM data, masking any ongoing network issues/activity and making the IOAM-Data-Fields collection useless.

This document defines the methods to protect the integrity of IOAM-Data-Fields, using the IOAM Option-Types specified in [RFC9197] as an example. The methods similarly apply to other IOAM Option-Types which contain IOAM-Data-Fields.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174].

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

MTU: Maximum Transmit Unit

OAM: Operations, Administration, and Maintenance

POT:
Proof of Transit

E2E: Edge to Edge

3. Threat Analysis

This section presents a threat analysis of integrity-related threats in the context of IOAM. The threats that are discussed are assumed to be independent of the lower layer protocols; it is assumed that threats at other layers are handled by security mechanisms that are deployed at these layers.

This document is focused on integrity protection for IOAM-Data-Fields. Thus the threat analysis includes threats that are related to or result from compromising the integrity of IOAM-Data-Fields. Other security aspects such as confidentiality are not within the scope of this document.

Throughout the analysis there is a distinction between on-path and off-path attackers. As discussed in [RFC9055], on-path attackers are located in a position that allows interception and modification of in-flight protocol packets, whereas off-path attackers can only attack by generating protocol packets.

The analysis also includes the impact of each of the threats. Generally speaking, the impact of a successful attack on an OAM protocol [RFC7276] is a false illusion of nonexistent failures or preventing the detection of actual ones; in both cases, the attack may result in denial of service (DoS). Furthermore, creating the false illusion of a nonexistent issue may trigger unnecessary processing in some of the IOAM nodes along the path, and may cause more IOAM-related data to be exported to the management plane than is conventionally necessary. Beyond these general impacts, threat-specific impacts are discussed in each of the subsections below.

3.1. Modification: IOAM-Data-Fields

Threat

An attacker can maliciously modify the IOAM-Data-Fields of in-transit packets. The modification can either be applied to all packets or selectively applied to a subset of the en route packets. This threat is applicable to on-path attackers.

Impact

By systematically modifying the IOAM-Data-Fields of some or all of the in-transit packets, an attacker can create a false picture of the paths in the network, the existence of faulty nodes and their location, and the network performance.

3.2. Modification: IOAM Option-Type Headers

Threat

An on-path attacker can modify the header in IOAM Option-Types in order to change or disrupt the behavior of nodes processing IOAM-Data-Fields along the path. This threat is not within the scope of this document.

Impact

Changing the header of IOAM Option-Types may have several implications. An attacker can maliciously increase the processing overhead in nodes that process IOAM-Data-Fields and increase the on-the-wire overhead of IOAM-Data-Fields, for example by modifying the IOAM-Trace-Type field in the IOAM Trace Option-Type header. An attacker can also prevent some of the nodes that process IOAM-Data-Fields from incorporating IOAM-Data-Fields, by modifying the RemainingLen field in the IOAM Trace Option-Type header.

3.3. Injection: IOAM-Data-Fields

Threat

An attacker can inject packets with IOAM Option-Types and IOAM-Data-Fields. This threat is applicable to both on-path and off-path attackers.

Impact

This attack and its impacts are similar to Section 3.1.

3.4. Injection: IOAM Option-Type Headers

Threat

An attacker can inject packets with IOAM Option-Type headers, thus manipulating other nodes that process IOAM-Data-Fields in the network. This threat is applicable to both on-path and off-path attackers. This threat is not within the scope of this document.

Impact

This attack and its impacts are similar to Section 3.2.

3.5. Management and Exporting

Threat

Attacks that compromise the integrity of IOAM-Data-Fields can be applied at the management plane, e.g., by manipulating network management packets. Furthermore, the integrity of IOAM-Data-Fields that are exported to a receiving entity can also be compromised. Management plane attacks are not within the scope of this document; the network management protocol is expected to include inherent security capabilities. The integrity of exported data is also not within the scope of this document. It is expected that the specification of the export format will discuss the relevant security aspects.

Impact

Malicious manipulation of the management protocol can cause nodes that process IOAM-Data-Fields to malfunction, to be overloaded, or to incorporate unnecessary IOAM-Data-Fields into user packets. The impact of compromising the integrity of exported IOAM-Data-Fields is similar to the impacts of previous threats that were described in this section.

3.6. Delay

Threat

An on-path attacker may delay some or all of the in-transit packets that include IOAM-Data-Fields in order to create the false illusion of congestion. Delay attacks are well known in the context of deterministic networks [RFC9055] and synchronization [RFC7384], and may be somewhat mitigated in these environments by using redundant paths in a way that is resilient to an attack along one of the paths. This approach does not address the threat in the context of IOAM, as it does not meet the requirement to measure a specific path or to detect a problem along the path. It is noted that this threat is not within the scope of the threats that are mitigated in this document.

Impact

Since IOAM can be applied to a fraction of the traffic, an attacker can detect and delay only the packets that include IOAM-Data-Fields, thus preventing the authenticity of delay and load measurements.

3.7. Threat Summary

Threat	In scope	Out of scope
Modification: IOAM-Data-Fields	+	
Modification: IOAM Option-Type Headers		+
Injection: IOAM-Data-Fields	+	
Injection: IOAM Option-Type Headers		+
Management and Exporting		+
Delay		+

Figure 1: Threat Analysis Summary

4. Integrity Protected Option-Types

This section defines new IOAM Option-Types to be allocated in the IOAM Option-Type Registry. Their purpose is to carry IOAM-Data-Fields with integrity protection. Each of the IOAM Option-Types defined in [RFC9197] is extended as follows:

64 IOAM Pre-allocated Trace Integrity Protected Option-Type: corresponds to the IOAM Pre-allocated Trace Option-Type with integrity protection.

65 IOAM Incremental Trace Integrity Protected Option-Type: corresponds to the IOAM Incremental Trace Option-Type with integrity protection.

66 IOAM POT Integrity Protected Option-Type: corresponds to the IOAM POT Option-Type with integrity protection.

67 IOAM E2E Integrity Protected Option-Type: corresponds to the IOAM E2E Option-Type with integrity protection.

The Integrity Protection subheader follows the IOAM Option-Type header when the IOAM Option-Type is an Integrity Protected Option-Type. It is defined as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Signature-suite|  Nonce length |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Nonce                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Signature                           ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Signature-suite: 8-bit unsigned integer. This field defines the algorithms used to compute the digest and the signature over the IOAM-Data-Fields.

Nonce length: 8-bit unsigned integer. This field specifies the length of the Nonce in octets.

Reserved: 16-bit Reserved field. MUST be set to zero upon transmission and ignored upon receipt.

Nonce: Variable length field with length specified in Nonce length.

Signature: Digital signature value generated by the method and algorithm specified by Signature-suite.

4.1. Integrity Protected Trace Option-Types

Both the IOAM Pre-allocated Trace Option-Type header and the IOAM Incremental Trace Option-Type header, as defined in [RFC9197], are followed by the Integrity Protection subheader when the IOAM Option-Type is respectively set to the IOAM Pre-allocated Trace Integrity Protected Option-Type or the IOAM Incremental Trace Integrity Protected Option-Type:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID          | NodeLen | Flags | RemainingLen|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                IOAM-Trace-Type                |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Signature-suite|  Nonce length |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Nonce                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Signature                           ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.2. Integrity Protected POT Option-Type

The IOAM POT Option-Type header, as defined in [RFC9197], is followed by the Integrity Protection subheader when the IOAM Option-Type is set to the IOAM POT Integrity Protected Option-Type:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID          | IOAM-POT-Type | IOAM-POT-Flags|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Signature-suite|  Nonce length |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Nonce                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Signature                           ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.3. Integrity Protected E2E Option-Type

The IOAM E2E Option-Type header, as defined in [RFC9197], is followed by the Integrity Protection subheader when the IOAM Option-Type is set to the IOAM E2E Integrity Protected Option-Type:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID          |         IOAM-E2E-Type         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Signature-suite|  Nonce length |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Nonce                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Signature                           ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5. Methods for space optimized integrity protection

Methods for space optimized integrity protection can leverage symmetric or asymmetric key based signatures, as described in the subsections below. The Signature consumes 32 octets and is carried only once for the entire packet. In case of performance concerns, such method can be applied to a subset of the traffic by using sampling of data to enable IOAM with integrity protection. Both symmetric and asymmetric signature methods work similarly, as follows:

1. The encapsulating node creates a nonce and stores it in the Nonce field of the Integrity Protection subheader. The signature is generated over the Nonce field and the hash of IOAM-Data-Fields it has inserted, i.e., sign(Nonce || hash(IOAM-Data-Fields)). IOAM-Data-Fields supposed to be modified by other IOAM nodes on the path MUST be excluded from the signature (e.g., the POT Cumulative field). The signature is stored in the Signature field of the Integrity Protection subheader. Important note: if all the inserted IOAM-Data-Fields are supposed to be modified by other IOAM nodes on the path, or if there is no IOAM-Data-Field inserted at all, then the encapsulating node MUST NOT use an Integrity Protected Option-Type.

2. A transit node generates a signature over the Signature field and the hash of IOAM-Data-Fields it has inserted, i.e., sign(Signature || hash(IOAM-Data-Fields)). IOAM-Data-Fields modified in-place by the transit node MUST be excluded from the signature (e.g., the POT Cumulative field). The signature is stored in the Signature field of the Integrity Protection subheader. Important note: if the transit node does not insert IOAM-Data-Fields (e.g., it only modifies IOAM-Data-Fields in-place, or does nothing), then the transit node MUST NOT generate a signature and MUST NOT update the Signature field.

3. The decapsulating node (aka the Validator) is responsible for the integrity verification of the IOAM-Data-Fields collected. Serving as the Validator, the decapsulating node MUST NOT generate a signature based on IOAM-Data-Fields it has inserted, if any, and therefore MUST NOT update the Signature field. To validate the IOAM-Data-Fields integrity, the Validator recomputes the signature by iteratively following the same procedure as for the encapsulating and transit nodes, in that order, using their respective keys (see Section 5.1 or Section 5.2 depending on the approach, i.e., symmetric or asymmetric). The recomputed signature is then compared to the Signature field. It is trivial in some cases (e.g., with POT Type-0 or E2E Option-Types), where only the encapsulating node generates a signature, as specified by the method described in this section. For other cases where transit nodes also generate a signature (e.g., with Trace Option-Types), node-ids MUST be included in IOAM-Data-Fields. Details on how the mapping between node-ids and keys is implemented on the Validator are outside the scope of this document.

5.1. Symmetric key based signature

This method assumes that symmetric keys have been distributed to the respective nodes as well as the Validator (the Validator receives all the keys). The details of the mechanisms responsible for key distribution are outside the scope of this document.

This method MUST use an algorithm pair defined in Section 6.2 and the approach MUST be symmetric.

5.2. Asymmetric key based signature

This method assumes that asymmetric keys have been generated per IOAM node and the respective nodes can access their keys (the Validator receives all the public keys). The details of the mechanisms responsible for key distribution are outside the scope of this document.

This method MUST use an algorithm pair defined in Section 6.2 and the approach MUST be asymmetric.

6. IANA Considerations

6.1. IOAM Option-Type Registry

This draft defines the following new code points in the IOAM Option-Type Registry:

64 IOAM Pre-allocated Trace Integrity Protected Option-Type

65
IOAM Incremental Trace Integrity Protected Option-Type

66 IOAM POT Integrity Protected Option-Type

67 IOAM E2E Integrity Protected Option-Type

6.2. IOAM Integrity Protection Algorithm Suite Registry

"IOAM Integrity Protection Algorithm Suite Registry" in the "In-Situ OAM (IOAM) Protocol Parameters" group. The one-octet "IOAM Integrity Protection Algorithm Suite Registry" identifiers assigned by IANA identify the digest algorithm and signature algorithm used in the Signature Suite Identifier field. IANA has registered the following algorithm suite identifiers for the digest algorithm and for the signature algorithm.

Algorithm Suite Identifier	Digest Algorithm	Signature Algorithm	Specification Pointer	Approach
0x00	Reserved	Reserved	This document	None
0x01	SHA-256	ECDSA P-256	[SHS] [DSS] [RFC6090] This document	Asymmetric
0x02	SHA-256	AES-256	[AES] [NIST.800-38D] This document	Symmetric
0x03-0xFF	Unassigned	Unassigned		

Figure 2: IOAM Integrity Protection Algorithm Suite Registry

Future assignments are to be made using the Standards Action process defined in [RFC8126]. Assignments consist of the one-octet algorithm suite identifier value and the associated digest algorithm name and signature algorithm name.

7. Security Considerations

This section discusses additional security aspects.

7.1. Replay protection

The nonce makes a signature chain unique but does not necessarily prevent replay attacks. To enable replay protection, the encapsulating node and the Validator MUST use a common, unique nonce.

8. Acknowledgements

The authors would like to thank Santhosh N, Rakesh Kandula, Saiprasad Muchala, Al Morton, Greg Mirsky, Benjamin Kaduk and Martin Duke for their comments and advice.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

[AES] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)", FIPS PUB 197, 2001, <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.

[DSS] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", NIST FIPS Publication 186-4, DOI 10.6028/NIST.FIPS.186-4, 2013, <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>.

[NIST.800-38D] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", NIST Special Publication 800-38D, 2001, <http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf>.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, DOI

10.17487/RFC6090, February 2011, <https://www.rfc-editor.org/info/rfc6090>.

[RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <https://www.rfc-editor.org/info/rfc7276>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <https://www.rfc-editor.org/info/rfc7384>.

[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>.

[RFC9055] Grossman, E., Ed., Mizrahi, T., and A. Hacker, "Deterministic Networking (DetNet) Security Considerations", RFC 9055, DOI 10.17487/RFC9055, June 2021, <https://www.rfc-editor.org/info/rfc9055>.

[RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi, Ed., "Data Fields for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197, May 2022, <https://www.rfc-editor.org/info/rfc9197>.

[SHS] National Institute of Standards and Technology, "Secure Hash Standard (SHS)", NIST FIPS Publication 180-4, DOI 10.6028/NIST.FIPS.180-4, 2015, <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>.

Authors' Addresses

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
40549 DUESSELDORF
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Thoughtspot
3rd Floor, Indiqube Orion, 24th Main Rd, Garden Layout, HSR Layout
Bangalore, KARNATAKA 560 102
India

Email: shwetha.bhandari@thoughtspot.com

Tal Mizrahi
Huawei
8-2 Matam
Haifa 3190501
Israel

Email: tal.mizrahi.phd@gmail.com

Justin Iurman
Universite de Liege
10, Allee de la decouverte (B28)
4000 Sart-Tilman
Belgium

Email: justin.iurman@uliege.be