ippm
Internet-Draft
Intended status: Standards Track
Expires: April 22, 2022

F. Brockners
Cisco
S. Bhandari
Thoughtspot
T. Mizrahi
Huawei

October 19, 2021

Integrity of In-situ OAM Data Fields
draft-ietf-ippm-ioam-data-integrity-00

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) collects operational and telemetry information in the packet while the packet traverses a path between two points 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

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 April 22, 2022.

Copyright Notice

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

"In-situ" Operations, Administration, and Maintenance (IOAM) collects 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 is being sent within packets specifically dedicated to OAM. IOAM is to complement mechanisms such as Ping, Traceroute, or other active

probing mechanisms. 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 live 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.

The current [I-D.ietf-ippm-ioam-data] 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 tracing data is carried in the packets in clear and there are no protections against any node or middlebox tampering with the data. IOAM tracing data collected in an untrusted or semi-trusted environments requires 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 trace data collected:

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

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

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

This document defines the methods to protect the integrity of IOAM data fields, using the IOAM Trace Option-Types specified in [I-D.ietf-ippm-ioam-data] as example. The methods similarily apply to other IOAM Option-Types such as the DEX Option-Type [I-D.ietf-ippm-ioam-direct-export].

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

SFC: Service Function Chain

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 [I-D.ietf-detnet-security], 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 IOAM data fields in one or more of the IOAM Option-Type headers in order to change or disrupt the

behavior of nodes processing IOAM data fields along the path.

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 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.

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 it 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.

Impact

This attack and it impacts are similar to Section 3.2.

3.5. Replay

Threat

An attacker can replay packets with IOAM data fields. Specifically, an attacker may replay a previously transmitted IOAM Option-Type with a new data packet, thus attaching old IOAM data fields to a fresh user packet. This threat is applicable to both on-path and off-path attackers.

Impact

As with previous threats, this threat may create a false image of a nonexistent failure, or may overload nodes which process IOAM data fields with unnecessary processing.

3.6. 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.

3.7. 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 [I-D.ietf-detnet-security] 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 the scope of 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.

3.8. 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	+	
Replay	+	
Management and Exporting		+
Delay		+

Figure 1: Threat Analysis Summary

4. Methods of providing integrity to IOAM data fields

This section specifies additional IOAM Option-Types to carry data fields to provide for integrity protection. Methods for integrity protection can leverage symmetric or asymmetric key based signatures as described in the sub-sections below.

4.1. Integrity Protected IOAM Option-Types

Each of the IOAM Options defined in [I-D.ietf-ippm-ioam-data] are extended to include Integrity Protected (IP) IOAM Option-Types by allocating the following IOAM Option-Types in the IOAM Option-Type registry.

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

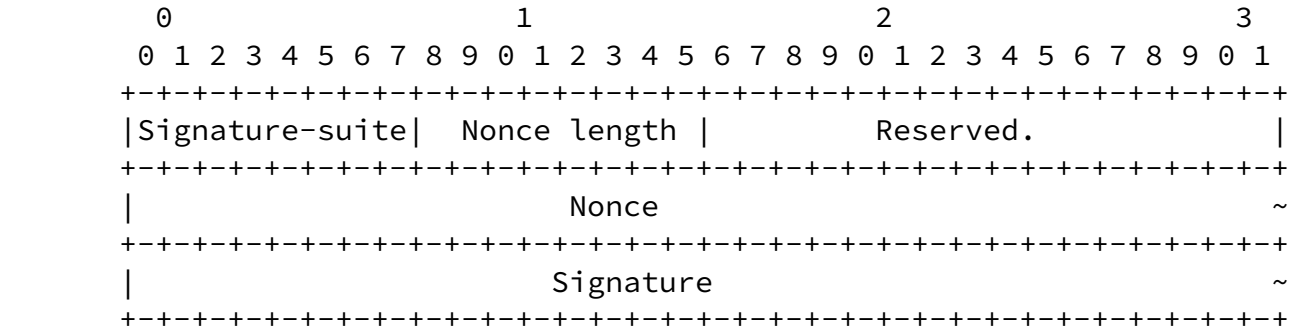
65 IOAM Incremental Trace Integrity Protected Option-Type corresponds to IOAM Incremental Trace Option-Type with integrity protection.

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

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

4.2. Subheader for Integrity Protected IOAM Option-Types

An integrity data sub-header is used in IOAM Integrity Protected Options. It is defined as follows:



Signature-suite: 8-bit unsigned integer. This field defines the algorithms used to compute the digest and the signature over the Option-Type header and data fields excluding the Signature field.

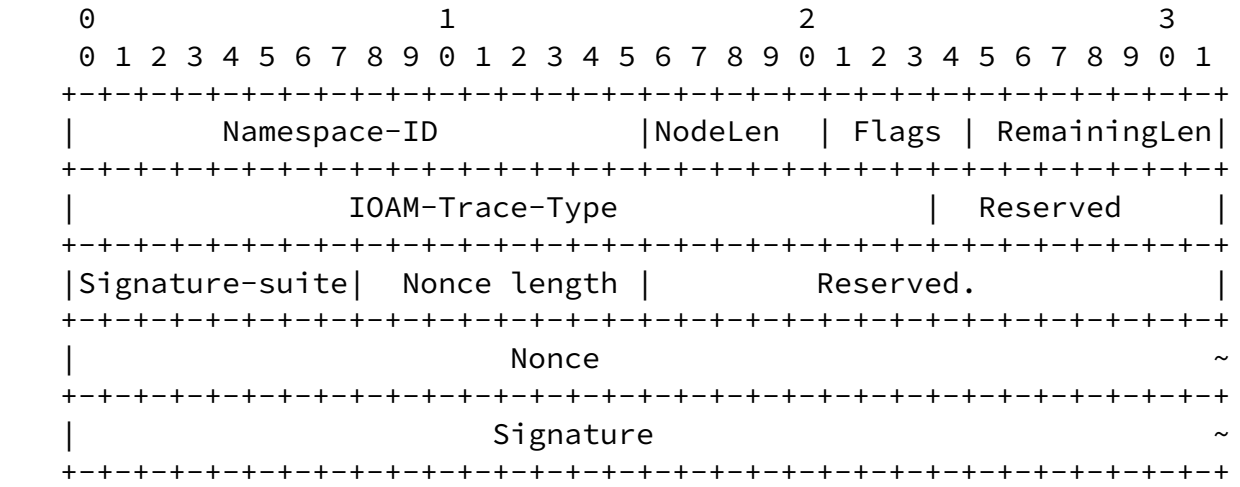
Nonce length: 8-bit unsigned integer. This field specifies the length of the Nonce field in octets.

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

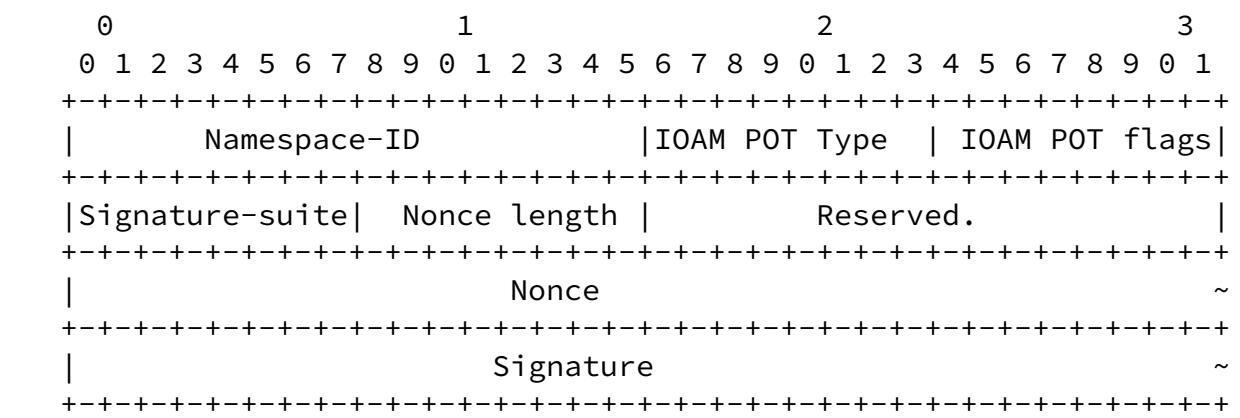
Nonce: Nonce is a variable length field with length specified in Nonce length.

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

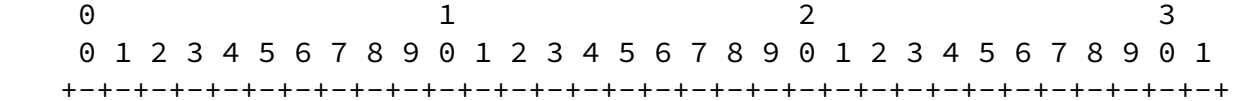
The Integrity sub-header follows the IOAM Option-Type header when the IOAM Option-Type is Integrity Protected Option. Pre-allocated and incremental Trace option headers are as defined in [I-D.ietf-ippm-ioam-data]. When the IOAM Option-Type is set to the IOAM Pre-allocated Trace Integrity Protected Option-Type or IOAM Incremental Trace Integrity Protected Option-Type then the Integrity Protection subheader follows the original IOAM Option Type header: :

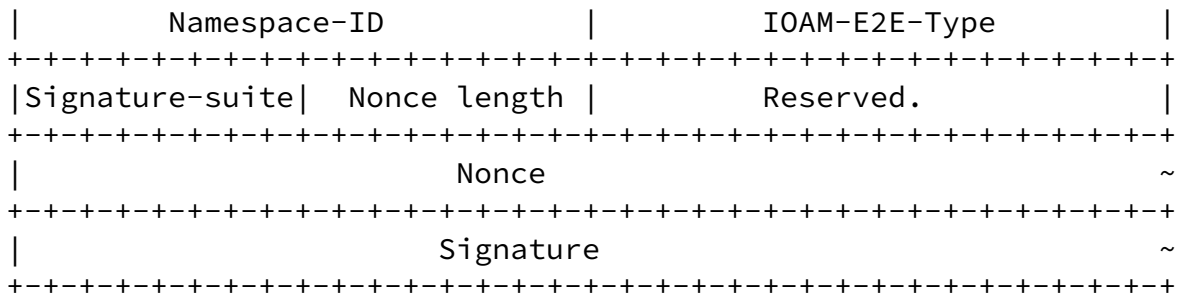


IOAM POT option header as defined in [I-D.ietf-ippm-ioam-data] is followed by Integrity Protection subheader when IOAM Option Type is set to IOAM POT Integrity Protected Option-Type:



IOAM E2E option header as defined in [I-D.ietf-ippm-ioam-data] is followed by Integrity Protection subheader when IOAM Option Type is set to IOAM E2E Integrity Protected Option-Type:





4.3. Space optimized symmetric key based signing of IOAM data

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 of how keys are distributed are outside the scope of this document. The "Signature" field is populated as follows:

1. The first node creates a nonce and signature over the hash of IOAM Option excluding the Signature field, the nonce and its symmetric key. The nonce is included as a field in Integrity Protection sub-header of the corresponding IOAM Option. The resulting signature is included in the corresponding Signature field.

2. Transit nodes will update the Signature field by creating a signature of data where the data is [Signature || hash(node_data_list[x])] with its symmetric key in case of IOAM Trace Integrity Protected Options. Transit nodes updating IOAM POT Option will update the Signature field by creating a signature of data where the data is [Signature || hash(IOAM POT OPTION excluding Signature field)] with its symmetric key in case of IOAM POT Integrity Protected Option.

3. The Validator will iteratively recreate the Signature over the IOAM Option fields collected and matches the Signature field to validate the data integrity.

This method uses the following algorithms:

1. The algorithm to calculate the signature using symmetric key MUST be Advanced Encryption Standard (AES) AES-256. [AES]

[NIST.800-38D].

2. The digest/hash algorithm used MUST be SHA-256 [SHS].

4.3.1. Overhead consideration

The Signature would consume 32 bytes with AES-256. With this method the Signature is carried only once for the entire packet. As there are dedicated options for carrying IOAM data with integrity protection, in case of performance concerns in calculating signature and validating it, these options can be used for a subset of the packets by using sampling of data to enable IOAM with integrity protection.

4.4. Space optimized asymmetric key based signing of trace data

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 of how keys are generated and shared are outside the scope of this document. The " Signature" field is populated as follows:

1. The first node creates a nonce and signs over the hash of IOAM Option it populates excluding the Signature field in the option, the nonce and its private key. The resulting signature is included in the Signature field.

2. Transit nodes will update the Signature field by creating a signature of data where the data is [Signature || hash(node_data_list[x])] with its private key in case of IOAM Trace Integrity Protected Options. Transit nodes updating IOAM POT Option will update the Signature field by creating a signature of data where the data is [Signature || hash(IOAM POT OPTION excluding Signature field)] with its private key in case of IOAM POT Integrity Protected Option.

3. The Validator will iteratively recreate the Signature over the IOAM Option fields collected and matches the Signature field to validate the data integrity using public keys of the IOAM nodes.

This method uses the following algorithms:

1. The signature algorithm used MUST be the Elliptic Curve Digital Signature Algorithm (ECDSA) with curve P-256 [RFC6090] [DSS].

2. The digest/hash algorithm used MUST be SHA-256 [SHS].

4.4.1. Overhead consideration

The Signature consumes 32 bytes based on the SHA-256 ECDSA P-256 algorithm employed. With this method the Signature is only carried once for the entire packet. As there are dedicated options for carrying IOAM data with integrity protection, in case of performance concerns in calculating signature and validating it, these options can be used for a subset of the packets by using sampling of data to enable IOAM with integrity protection.

5. IANA Considerations

5.1. IOAM Option-Type Registry

The following code points are defined in this draft in "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

5.2. 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.

IOAM Integrity Protection Algorithm Suite Registry

Algorithm Suite Identifier	Digest Algorithm	Signature Algorithm	Specification Pointer
0x0	Reserved	Reserved	This document
0x1	SHA-256	ECDSA P-256	[SHS] [DSS] [RFC6090] This document
0x2	SHA-256	AES-256	[AES] [NIST.800-38D] This document
0xEF-0xFF	Unassigned	Unassigned	

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.

6. Security Considerations

This section will be completed in a future revision of this document.

7. 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.

8. References

8.1. Normative References

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

[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>.

8.2. 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>.

[I-D.ietf-detnet-security] Grossman, E., Mizrahi, T., and A. J. Hacker, "Deterministic Networking (DetNet) Security Considerations", draft-ietf-detnet-security-16 (work in progress), March 2021.

[I-D.ietf-ippm-ioam-data] Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-15 (work in progress), October 2021.

[I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-07 (work in progress), October 2021.

[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>.

[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
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
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