

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 20 January 2027

A. Becker
M. Jenkins
NSA-CCSS
19 July 2026

Commercial National Security Algorithm (CNSA) Suite 2.0 Profile for TLS
1.3
draft-becker-cnsa2-tls-profile-05

Abstract

This document defines a base profile of TLS 1.3 which is compliant with the US Commercial National Security Algorithm (CNSA) 2.0 Suite, a cybersecurity advisory published by the United States Government which outlines quantum-resistant cryptographic algorithm policy for US national security applications.

This profile applies to the capabilities, configuration, and operation of all components of US National Security Systems that employ TLS 1.3. It is also appropriate for all other US Government systems that process high-value information.

This memo is not an IETF standard, and has not been shown to have IETF community consensus. This profile is made publicly available for use by developers and operators of these and any other system deployments.

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 20 January 2027.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	3
2. Terminology	3
3. The Commercial National Security Algorithm Suite 2.0	4
4. CNSA 2.0 Suite	4
5. Compliance and Interoperability	5
6. TLS Version	5
"supported_versions" Extension	5
7. Key Exchange Messages	5
7.1. Cipher Suite Negotiation	6
7.2. KEM Requirements	6
7.2.1. "supported_groups" Extension	6
7.2.2. "key_share" Extension	6
8. Authentication Messages	7
8.1. "signature_algorithms" Extension	7
8.2. "signature_algorithms_cert" Extension	7
8.3. CertificateRequest Message	8
8.4. Certificate Selection	8
8.5. CertificateVerify Message	8
9. External Pre-Shared Keys	9
10. Resumption	9
11. Certificate Status	10
12. "early_data" Extension	10
13. DTLS 1.3	10
14. Security Considerations	10
15. IANA Considerations	11
16. Acknowledgements	11
17. References	11
17.1. Normative References	11
17.2. Informative References	13
Authors' Addresses	13

1. Introduction

This document specifies a profile of TLS 1.3 [RFC8446] to comply with the National Security Agency's (NSA) Commercial National Security Algorithm (CNSA) 2.0 Suite [cnsafaq]. This profile applies to the capabilities, configuration, and operation of all components of US National Security Systems (NSS) [SP80059] that employ TLS 1.3. This profile is also appropriate for all other US Government systems that process high-value information, and is made publicly available for use by developers and operators of these and any other system deployments.

This document does not define any cryptographic algorithm, and does not specify how to use any cryptographic algorithm not currently supported by TLS 1.3; instead, it profiles CNSA 2.0-compliant conventions for TLS 1.3, and uses algorithms already specified for use in TLS 1.3 that are also allowed by the CNSA Suite 2.0. This document applies to all CNSA Suite solutions that make use of TLS 1.3.

The reader is assumed to have familiarity with [I-D.ietf-tls-8773bis].

This memo is not an IETF standard, and has not been shown to have IETF community consensus.

This document obsoletes the CSNA 1.0 guidance in [RFC9151] for TLS 1.3. Servers and clients that also support TLS 1.2 (such as, during transition) MUST continue to comply with that guidance.

2. Terminology

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] when, and only when, they appear in all capitals, as shown here. Normative language does not apply beyond the scope of this profile.

This is a profile of TLS 1.3 [RFC8446]. Therefore, the requirements language in this memo may be different than that found in the underlying standards.

All references to "CNSA" in this document refer to CNSA 2.0 [cnsafaq], unless stated otherwise.

3. The Commercial National Security Algorithm Suite 2.0

The CNSA Suite is the set of approved commercial algorithms that can be used by vendors and IT users to meet cybersecurity and interoperability requirements for NSS. The initial suite of CNSA Suite algorithms, Suite B, established a baseline for use of commercial algorithms to protect classified information. The following suite, CNSA 1.0, served as a bridge between the original set and a fully quantum-resistant cryptographic capability. The current suite, CNSA 2.0, seeks to provide fully quantum-resistant protection [cnsafaq].

This profile uses CNSA 2.0 compliant algorithms: ML-DSA-87 [FIPS204] for signing, and ML-KEM-1024 [FIPS203] for key establishment. ML-DSA-87 and ML-KEM-1024 together with SHA-384, SHA-512, AES-256, LMS, and XMSS comprise the CNSA Suite 2.0.

The CNSA Suite was selected by NSA for protection of US national security information based on factors including security, performance, data value and retention requirements, advanced threat actors, and certification process. This suite will not be appropriate for all use cases; data owners are encouraged to perform their own risk assessments and consult with their data protection systems providers.

The NSA is authoring a set of RFCs, including this one, to provide updated guidance for using the CNSA 2.0 Suite in certain IETF protocols. These RFCs can be used in conjunction with other RFCs and cryptographic guidance (e.g., NIST Special Publications) to properly protect Internet traffic and data-at-rest for US Government NSS.

4. CNSA 2.0 Suite

CNSA requires the following algorithms for use in TLS 1.3:

Encryption:

AES [AES] with 256-bit keys, operating in Galois Counter Mode (GCM) [GCM] ({0x13, 0x02} Appendix B.4 [RFC8446])

Key Establishment:

ML-KEM-1024 [FIPS203] ({0x0202} MLKEM1024)

Digital Signature

ML-DSA-87 [FIPS204] ({0x0906} MLDSA87)

The ML-DSA algorithm incorporates an internal hashing function, so there is no need to apply a hashing algorithm before signing. Where an application or implementation makes it more efficient to perform hashing externally, the external- $\hat{H}$ mechanism described in Step 6 of Algorithm 7 of [FIPS204] and Section 8 of [RFC9881] MAY be used. HashML-DSA is not permitted.

CNSA requires the use of SHA-384 [SHS] for the HMAC-based Key Derivation Function (HKDF) in TLS 1.3.

5. Compliance and Interoperability

In some situations, clients or servers configured for CNSA compliance also have to interoperate with non-compliant clients and servers. Compliant clients and servers MUST be configured such that CNSA compliance is preferred even if it is not intended or expected. CNSA-compliant implementations MUST present CNSA compliant algorithms first in the appropriate extensions, and CNSA-compliant algorithms MUST be used when supported by the peer. If any aspect of CNSA compliance is not achieved, the connection is not CNSA compliant. Any connection negotiated to TLS version 1.2 or lower cannot be CNSA compliant.

If interoperability with non-CNSA-compliant clients or servers is not intended, then the session MUST fail if any aspect of CNSA compliance is not achieved.

6. TLS Version

CNSA TLS clients and servers MUST negotiate TLS version 1.3 [RFC8446] when establishing a CNSA-compliant connection. CNSA TLS clients and CNSA TLS servers MUST NOT negotiate TLS versions prior to TLS 1.3 in a CNSA-compliant mode.

"supported_versions" Extension

A CNSA TLS client connecting to a CNSA TLS server MUST include the "supported_versions" extension in the ClientHello (Section 4.2.1 of [RFC8446]) and list TLS 1.3 version value (0x0304) first in the extension.

7. Key Exchange Messages

This section details requirements for CNSA 2.0-compliant algorithm negotiation in TLS 1.3.

7.1. Cipher Suite Negotiation

A CNSA TLS client MUST offer the following cipher suite in the ClientHello:

TLS_AES_256_GCM_SHA384 {0x13, 0x02} (Appendix B.4 [RFC8446])

This CNSA cipher suite MUST be the first (most preferred) cipher suite in the ClientHello message and in the extensions.

A CNSA TLS server MUST select this cipher suite if offered by the client.

Any cipher suite other than TLS_AES_256_GCM_SHA384 offered by the client MUST NOT be accepted by a CNSA TLS server establishing a CNSA-compliant connection.

7.2. KEM Requirements

CNSA 2.0 requires the use of ML-KEM-1024 for key establishment in TLS.

7.2.1. "supported_groups" Extension

A CNSA TLS client MUST offer the following value in named_group_list of the "supported_groups" extension (Section 4.2.7 [RFC8446]) of the ClientHello:

ML-KEM-1024 {0x0202}

ML-KEM-1024 MUST be the first (most preferred) item in named_group_list.

A CNSA TLS server MUST select ML-KEM-1024 if it is included in the "supported_groups" extension sent by a CNSA TLS client in the ClientHello.

Any algorithm other than ML-KEM-1024 offered by the client MUST NOT be accepted by a CNSA TLS server establishing a CNSA-compliant connection.

7.2.2. "key_share" Extension

The "key_share" extension in a CNSA TLS client ClientHello MUST contain the ML-KEM-1024 public key .

The ML-KEM-1024 public key must be the first (most preferred) key in the list of key shares.

A CNSA TLS server MUST select the ML-KEM-1024 key share for key establishment, and the "key_share" extension in a CNSA TLS server ServerHello MUST contain the ML-KEM-1024 ciphertext associated to the public key provided in the ClientHello .

8. Authentication Messages

A CNSA TLS client MUST require the server to authenticate itself. In all cases, a CNSA TLS client MUST authenticate the server using X.509 certificates; PSK-based authentication MAY be used in addition to certificate-based authentication as detailed in Section 9.

A CNSA TLS server MAY also authenticate the client as needed by the specific application. If mutual authentication is desired, X.509 certificates MUST be used in all cases.

8.1. "signature_algorithms" Extension

A CNSA TLS client MUST offer the following value in the "signature_algorithms" extension of the ClientHello:

ML-DSA-87 {0x0906}

The ML-DSA-87 algorithm must be the first (most preferred) value in the extension.

Any signature algorithm values offered other than ML-DSA-87 MUST NOT be accepted by a CNSA TLS server establishing a CNSA-compliant connection.

8.2. "signature_algorithms_cert" Extension

Per [RFC8446], the "signature_algorithms_cert" extension applies to signatures in certificates, while the "signature_algorithms" extension (Section 6.1) applies to signatures in CertificateVerify messages.

The "signature_algorithms_cert" extension is not required for a CNSA-compliant connection, as ML-DSA-87 is the only allowed signature algorithm that can be used for both signatures in the certificate and in the CertificateVerify message under CNSA compliance.

However, if the "signature_algorithms_cert" extension is included, the CNSA TLS client MUST offer the following value first in the "supported_signature_algorithms" field of the extension:

ML-DSA-87 {0x0906}

Any signature algorithm offered other than ML-DSA-87 MUST NOT be accepted by a CNSA TLS server establishing a CNSA-compliant connection.

8.3. CertificateRequest Message

A CNSA TLS server MAY authenticate the client as needed by the specific application.

If the CNSA TLS server requires mutual authentication, it MUST authenticate the client using X.509 certificates.

The "signature_algorithms" extension sent by the CNSA TLS server in the CertificateRequest message MUST include:

ML-DSA-87 {0x0906}

The "signature_algorithms_cert" extension is not required, but if it is included by the CNSA TLS server in the CertificateRequest message, then it MUST include:

ML-DSA-87 {0x0906}

ML-DSA-87 MUST be the first (most preferred) algorithm in the "signature_algorithms" and "signature_algorithms_cert" extensions.

8.4. Certificate Selection

Certificates sent by a CNSA TLS server (and CNSA TLS client, when required) MUST be signed by ML-DSA-87, and MUST be compliant with the CNSA Certificate and Certificate Revocation List Profile [I-D.jenkins-cnsa2-pkix-profile]. If the client cannot validate the server certificate for any reason, it must abort the handshake.

If a CNSA TLS server sends a CertificateRequest message to the client, the client MUST respond with a valid X.509 certificate suitable for authenticating the client. If the CNSA TLS client sends an empty Certificate message, the CNSA TLS server MUST abort the handshake (Section 4.4.2.4 of [RFC8446]). Also, if some aspect of the certificate chain was unacceptable (e.g., it was not signed by a known, trusted CA), the server MUST abort the handshake.

8.5. CertificateVerify Message

A CNSA TLS client or CNSA TLS server MUST use ML-DSA-87 when sending the CertificateVerify message. CNSA TLS clients or servers MUST accept only ML-DSA-87 in the CertificateVerify message when establishing a CNSA-compliant connection.

9. External Pre-Shared Keys

RFC 8773 [I-D.ietf-tls-8773bis] specifies an extension that allows an external PSK to be used for authentication in addition to (and not in lieu of) certificate-based authentication during the initial handshake. The PSK also contributes to the TLS 1.3 key schedule (Section 7.1, [RFC8446]). PSK-only authentication is not compliant beyond resumption as covered in Section 10.

A CNSA TLS client that supports RFC 8773 MUST include the "tls_cert_with_extern_psk" extension (Section 4, [I-D.ietf-tls-8773bis]) in the ClientHello message if it desires an external PSK to be used for authentication with certificate-based authentication. This extension is accompanied by the "key_share", "psk_key_exchange_modes", and "pre_shared_key" extensions defined in (Section 4.2.9 and 4.2.11, respectively, of [RFC8446]).

The "psk_key_exchange_modes" extension MUST NOT include psk_ke mode. The "psk_key_exchange_modes" extension MUST be psk_dhe_ke using ML-KEM-1024.

PSKs shall be at least 256 bits in length and generated from a NIST approved random bit generator that supports 256-bits of entropy [SP800-90c].

If the CNSA TLS server recognizes the "tls_cert_with_extern_psk" extension and possesses at least one of the external PSKs offered by the client in the "pre_shared_key" extension in the ClientHello, then the CNSA TLS server MUST select a CNSA compliant PSK and respond with the "tls_cert_with_extern_psk", "key_share", and "pre_shared_key" extensions in the ServerHello message (Section 4, [I-D.ietf-tls-8773bis]).

10. Resumption

A CNSA TLS server MAY send a CNSA TLS client a NewSessionTicket message to enable resumption. A CNSA TLS client MUST request "psk_dhe_ke" via the "psk_key_exchange_modes" extension in the ClientHello to resume a session. The "supported_groups" and "key_share" extensions MUST comply with those detailed in Section 7.2.1 and Section 7.2.2 of this document, respectively.

11. Certificate Status

A CNSA TLS server (and CNSA TLS client, if client authentication is required) MUST provide certificate status information either via a Certificate Revocation List (CRL) distribution points or by the Online Certificate Status Protocol (OCSP) (with or without stapling). For details on CNSA requirements for these methods, see [I-D.jenkins-cnsa2-pkix-profile].

12. "early_data" Extension

A CNSA TLS client or server MUST NOT include the "early_data" extension.

13. DTLS 1.3

CNSA DTLS clients and servers MUST negotiate DTLS version 1.3, [RFC9147] when establishing a CNSA-compliant connection. DTLS 1.3, 0xfefc, MUST be listed first in the "supported_versions" extension sent by the CNSA DTLS client. CNSA DTLS clients and CNSA DTLS servers MUST NOT negotiate DTLS versions prior to DTLS 1.3 in a CNSA-compliant mode.

A CNSA DTLS client MUST offer the cipher suite TLS_AES_256_GCM_SHA384 {0x13, 0x02} as the first (most preferred) cipher suite in the ClientHello message.

For key establishment, CNSA DTLS clients and servers MUST negotiate use of ML-KEM-1024 {0x0202} .

For authentication, CNSA DTLS clients and servers MUST negotiate use of ML-DSA-87 as the signature algorithm used in the "signature_algorithms" extension (and the "signature_algorithms_cert" extension, if present).

DTLS implementations MUST be consistent with the requirements of TLS defined in this document, except as specified in this section.

14. Security Considerations

Most of the security considerations for this document are described in [RFC8446], [I-D.ietf-tls-8773bis].

As noted in TLS version 1.3 [RFC8446], TLS does not provide inherent replay protections for early data. For this reason, this profile forbids the use of early data.

15. IANA Considerations

This document has no IANA considerations.

16. Acknowledgements

Alison Becker was the primary editor and contributed significantly to this document while she was employed at the National Security Agency's Center for Cybersecurity Standards.

17. References

17.1. Normative References

- [AES] National Institute of Standards and Technology, "Announcing the ADVANCED ENCRYPTION STANDARD (AES)", FIPS 197, DOI 10.6028/NIST.FIPS.197, November 2001, <<https://nvlpubs.nist.gov/nistpubs/fips/NIST.FIPS.197.pdf>>.
- [cnsafaq] National Security Agency, "The Commercial National Security Algorithm Suite 2.0 and Quantum Computing FAQ", December 2024, <https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI_CNSA_2.0_FAQ_.PDF>.
- [FIPS203] National Institute of Standards and Technology (2024), "Module-Lattice-Based Key-Encapsulation Mechanism Standard", (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS), NIST FIPS 203, <<https://doi.org/10.6028/NIST.FIPS.203>>.
- [FIPS204] National Institute of Standards and Technology (2024), "Module-Lattice-Based Digital Signature Standard", (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS), NIST FIPS 204, <<https://doi.org/10.6028/NIST.FIPS.204>>.
- [GCM] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", NIST Special Publication 800-38D, DOI 10.6028/NIST.SP.800-38D, November 2007, <<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf>>.

- [I-D.ietf-tls-8773bis]
Housley, R., "TLS 1.3 Extension for Using Certificates with an External Pre-Shared Key", July 2024,
<<https://datatracker.ietf.org/doc/draft-ietf-tls-8773bis/02/>>.
- [I-D.jenkins-cnsa2-pkix-profile]
Jenkins, M. and A. Becker, "Commercial National Security Algorithm Suite Certificate and Certificate Revocation List Profile", January 2025,
<<https://datatracker.ietf.org/doc/draft-jenkins-cnsa2-pkix-profile/>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC9147] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022,
<<https://www.rfc-editor.org/info/rfc9147>>.
- [RFC9151] Cooley, D., "Commercial National Security Algorithm (CNSA) Suite Profile for TLS and DTLS 1.2 and 1.3", RFC 9151,
DOI 10.17487/RFC9151, April 2022,
<<https://www.rfc-editor.org/info/rfc9151>>.
- [RFC9881] Massimo, J., Kampanakis, P., Turner, S., and B. E. Westerbaan, "Internet X.509 Public Key Infrastructure -- Algorithm Identifiers for the Module-Lattice-Based Digital Signature Algorithm (ML-DSA)", RFC 9881,
DOI 10.17487/RFC9881, October 2025,
<<https://www.rfc-editor.org/info/rfc9881>>.
- [SHS] National Institute of Standards and Technology (NIST),
"Secure Hash Standard (SHS)", DOI 10.6028/NIST.FIPS.180-4,
FIPS PUB 180-4, August 2015,
<<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>>.

17.2. Informative References

[SP800-90c]

National Institute of Standards and Technology,
"Recommendation for Random Bit Generator (RBG)
Constructions.", NIST SP 800-90C 4pd,
<<https://doi.org/10.6028/NIST.SP.800-90C.4pd>>.

[SP80059] National Institute of Standards and Technology, "Guideline
for Identifying an Information System as a National
Security System", Special Publication 59, DOI
10.6028/NIST.SP.800-59, August 2003,
<[https://csrc.nist.gov/publications/detail/sp/800-59/
final](https://csrc.nist.gov/publications/detail/sp/800-59/final)>.

Authors' Addresses

Alison Becker
Email: alison.elaine90@gmail.com

Michael Jenkins
NSA Center for Cybersecurity Standards
Email: mjjenki@cyber.nsa.gov

LAMPS
Internet-Draft
Intended status: Standards Track
Expires: 7 November 2026

D. Van Geest
CryptoNext Security
M. Ounsworth
J. Gray
Entrust
J. Klaussner
Bundesdruckerei GmbH
6 May 2026

Composite ML-KEM for use in Cryptographic Message Syntax (CMS)
draft-ietf-lamps-cms-composite-kem-01

Abstract

Composite ML-KEM defines combinations of ML-KEM with RSA-OAEP, ECDH, X25519, and X448. This document specifies the conventions for using Composite ML-KEM algorithms with the Cryptographic Message Syntax (CMS) using the KEMRecipientInfo structure defined in [RFC 200\234](#) Using Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS) [RFC 200\235](#) (RFC 9629).

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://lamps-wg.github.io/draft-composite-kem/draft-ietf-lamps-pq-composite-kem.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-lamps-cms-composite-kem/>.

Discussion of this document takes place on the LAMPS Working Group mailing list (<mailto:spams@ietf.org>), which is archived at <https://datatracker.ietf.org/wg/lamps/about/>. Subscribe at <https://www.ietf.org/mailman/listinfo/spams/>.

Source for this draft and an issue tracker can be found at <https://github.com/lamps-wg/draft-composite-kem>.

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 7 November 2026.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. ASN.1	3
1.2. Conventions and Terminology	3
1.3. Composite ML-KEM	3
2. Use of Composite ML-KEM in the CMS	4
2.1. RecipientInfo Conventions	4
2.2. Underlying Components	5
2.2.1. Use of the HKDF-Based Key Derivation Function	6
2.3. Certificate Conventions	6
2.4. SMIME Capabilities Attribute Conventions	6
3. Identifiers	7
4. Security Considerations	8
5. IANA Considerations	9
6. References	9
6.1. Normative References	9
6.2. Informative References	11
Appendix A. ASN.1 Module	12
Appendix B. Composite ML-KEM CMS Authenticated-Enveloped-Data Example	13
B.1. Originator CMS Processing	14
B.2. Recipient CMS Processing	20
Appendix C. Acknowledgments	20
Authors' Addresses	20

1. Introduction

[I-D.ietf-lamps-pq-composite-kem] defines a collection of Key Encapsulation Mechanism (KEM) algorithms, referred to as Composite ML-KEM, which combine ML-KEM [FIPS203] with traditional algorithms RSA-OAEP, ECDH, X25519, and X448. [RFC9629] defines the KEMRecipientInfo structure for the use of KEM algorithms for the Cryptographic Message Syntax (CMS) [RFC5652] enveloped-data content type, the CMS authenticated-data content type, and the CMS authenticated-enveloped-data content type. This document acts as a companion to [I-D.ietf-lamps-pq-composite-kem] by providing conventions for using Composite ML-KEM algorithms with the KEMRecipientInfo structure within the CMS.

1.1. ASN.1

CMS values are generated using ASN.1 [X680], using the Basic Encoding Rules (BER) and the Distinguished Encoding Rules (DER) [X690].

1.2. Conventions and Terminology

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] when, and only when, they appear in all capitals, as shown here. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

1.3. Composite ML-KEM

ML-KEM is a lattice-based KEM using Module Learning with Errors as its underlying primitive. It was standardized with three parameter sets: ML-KEM-512, ML-KEM-768, and ML-KEM-1024. Composite ML-KEM pairs ML-KEM-768 or ML-KEM-1024 with RSA-OAEP, ECDH, X25519, or X448 at similar security levels such that the shared secret key from each component algorithm is combined into a single shared secret key. Composite ML-KEM does not provide pairings based on ML-KEM-512.

All KEM algorithms provide three functions: KeyGen(), Encapsulate(), and Decapsulate().

The following summarizes these three functions for Composite ML-KEM:

KeyGen() -> (ek, dk): Generate the public encapsulation key (ek) and a private decapsulation key (dk). Section 3.1 of [I-D.ietf-lamps-pq-composite-kem] specifies the key generation algorithm for Composite ML-KEM.

Encapsulate(ek) -> (c, ss): Given the recipient's public key (ek), produce both a ciphertext (c) to be passed to the recipient and a shared secret (ss) for use by the originator. Section 3.2 of [I-D.ietf-lamps-pq-composite-kem] specifies the encapsulation algorithm for Composite ML-KEM.

Decapsulate(dk, c) -> ss: Given the private key (dk) and the ciphertext (c), produce the shared secret (ss) for the recipient. Section 3.3 of [I-D.ietf-lamps-pq-composite-kem] specifies the decapsulation algorithm for Composite ML-KEM.

2. Use of Composite ML-KEM in the CMS

Composite ML-KEM algorithms MAY be employed for one or more recipients in the CMS enveloped-data content type [RFC5652], the CMS authenticated-data content type [RFC5652], or the CMS authenticated-enveloped-data content type [RFC5083]. In each case, the KEMRecipientInfo [RFC9629] type is used with the Composite ML-KEM algorithm to securely transfer the content-encryption key from the originator to the recipient.

Processing a Composite ML-KEM algorithm with KEMRecipientInfo follows the same steps as Section 2 of [RFC9629]. To support the Composite ML-KEM algorithm, a CMS originator MUST implement the Encapsulate() function and a CMS recipient MUST implement the Decapsulate() function.

2.1. RecipientInfo Conventions

When a Composite ML-KEM algorithm is employed for a recipient, the RecipientInfo alternative for that recipient MUST be OtherRecipientInfo using the KEMRecipientInfo structure as defined in [RFC9629].

The fields of the KEMRecipientInfo have the following meanings:

version

The syntax version number; it MUST be 0.

rid

Identifies the recipient's certificate or public key.

kem

Identifies the KEM algorithm; it MUST contain one of the Composite ML-KEM OIDs in Section 3.

kemct

The ciphertext produced for this recipient.

kdf

Identifies the key derivation algorithm. Note that the Key Derivation Function (KDF) used for CMS RecipientInfo process MAY be different than the KDF used within the Composite ML-KEM algorithm. Implementations MUST support the HMAC-based Key Derivation Function (HKDF) [RFC5869] with SHA-256 [FIPS180], using the id-alg-hkdf-with-sha256 KDF object identifier (OID) [RFC8619]. As specified in [RFC8619], the parameter field MUST be absent when this OID appears within the ASN.1 type AlgorithmIdentifier. Implementations MAY support other KDFs as well.

kekLength

The size of the key-encryption key in octets.

ukm

Optional input to the KDF. The secure use of Composite ML-KEM in CMS does not depend on the use of a ukm value, so this document does not place any requirements on this value. See Section 3 of [RFC9629] for more information about the ukm parameter.

wrap

Identifies a key-encryption algorithm used to encrypt the content-encryption key. Implementations MUST support the AES-Wrap-256 [RFC3394] key-encryption algorithm using the id-aes256-wrap key-encryption algorithm OID [RFC3565]. Implementations MAY support other key-encryption algorithms as well.

Appendix B contains an example of establishing a content-encryption key using Composite ML-KEM in the KEMRecipientInfo type.

2.2. Underlying Components

When Composite ML-KEM is employed in the CMS, the underlying components used within the KEMRecipientInfo structure SHOULD be consistent with a minimum desired security level. Several security levels have been identified [SP.800-57pt1r5].

If underlying components other than those specified in Section 2.1 are used, then the following table gives the minimum requirements on the components used with Composite ML-KEM in the KEMRecipientInfo type in order to satisfy the KDF and key wrapping algorithm requirements from Section 7 of [RFC9629]. The components are chosen based on the ML-KEM variant used within the Composite ML-KEM algorithm.

Security Strength	ML-KEM Variant	KDF Preimage Strength	Symmetric Key-Encryption Strength
192-bit	ML-KEM-768	192-bit	192-bit (*)
256-bit	ML-KEM-1024	256-bit	256-bit

Table 1: Composite ML-KEM KEMRecipientInfo Component Security Levels

(*) In the case of AES Key Wrap, a 256-bit key is typically used because AES-192 is not as commonly deployed.

2.2.1. Use of the HKDF-Based Key Derivation Function

The HKDF function is a composition of the HKDF-Extract and HKDF-Expand functions.

```
HKDF(salt, IKM, info, L)
  = HKDF-Expand(HKDF-Extract(salt, IKM), info, L)
```

When used with KEMRecipientInfo, the salt parameter is unused; that is, it is the zero-length string "". The IKM, info, and L parameters correspond to the same KDF inputs from Section 5 of [RFC9629]. The info parameter is independently generated by the originator and recipient. Implementations MUST confirm that L is consistent with the key size of the key-encryption algorithm.

2.3. Certificate Conventions

[RFC5280] specifies the profile for using X.509 certificates in Internet applications. A recipient static public key is needed for Composite ML-KEM and the originator obtains that public key from the recipient's certificate. The conventions for carrying Composite ML-KEM public keys are specified in [I-D.ietf-lamps-pq-composite-kem].

2.4. SMIME Capabilities Attribute Conventions

Section 2.5.2 of [RFC8551] defines the SMIMECapabilities attribute to announce a partial list of algorithms that an S/MIME implementation can support. When constructing a CMS signed-data content type [RFC5652], a compliant implementation MAY include the SMIMECapabilities attribute that announces support for one or more of the Composite ML-KEM algorithm identifiers.

The SMIMECapability SEQUENCE representing the Composite ML-KEM algorithm MUST include one of the Composite ML-KEM OIDs in the capabilityID field. When one of the Composite ML-KEM OIDs appears in the capabilityID field, the parameters MUST NOT be present.

3. Identifiers

All identifiers used to indicate Composite ML-KEM within the CMS are defined in [I-D.ietf-lamps-pq-composite-kem], [RFC8619], and [RFC3565]; they are reproduced here for convenience:

-- Composite ML-KEM OIDs

```
id-MLKEM768-RSA2048-SHA3-256 OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) alg(6) 55 }
```

```
id-MLKEM768-RSA3072-SHA3-256 OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) alg(6) 56 }
```

```
id-MLKEM768-RSA4096-SHA3-256 OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) alg(6) 57 }
```

```
id-MLKEM768-X25519-SHA3-256 OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) alg(6) 58 }
```

```
id-MLKEM768-ECDH-P256-SHA3-256 OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) alg(6) 59 }
```

```
id-MLKEM768-ECDH-P384-SHA3-256 OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) alg(6) 60 }
```

```
id-MLKEM768-ECDH-brainpoolP256r1-SHA3-256 OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) alg(6) 61 }
```

```
id-MLKEM1024-RSA3072-SHA3-256 OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) alg(6) 62 }
```

```
id-MLKEM1024-ECDH-P384-SHA3-256 OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) alg(6) 63 }
```

```
id-MLKEM1024-ECDH-brainpoolP384r1-SHA3-256 OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) alg(6) 64 }

id-MLKEM1024-X448-SHA3-256 OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) alg(6) 65 }

id-MLKEM1024-ECDH-P521-SHA3-256 OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) alg(6) 66 }

-- KEMRecipientInfo.kdf OIDs

id-alg-hkdf-with-sha256 OBJECT IDENTIFIER ::= { iso(1)
    member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
    smime(16) alg(3) 28 }

-- KEMRecipientInfo.wrap OIDs

aes OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16) us(840)
    organization(1) gov(101) csor(3) nistAlgorithms(4) 1 }

id-aes256-wrap OBJECT IDENTIFIER ::= { aes 45 }
```

4. Security Considerations

The Security Considerations sections of [I-D.ietf-lamps-pq-composite-kem] and [RFC9629] apply to this specification as well.

Implementations MUST protect the Composite ML-KEM private key, the key-encryption key, the content-encryption key, message-authentication key, and the content-authenticated-encryption key. Of these keys, all but the private key are ephemeral and MUST be wiped after use. Disclosure of the Composite ML-KEM private key could result in the compromise of all messages protected with that key. Disclosure of the key-encryption key, the content-encryption key, or the content-authenticated-encryption key could result in the compromise of the associated encrypted content. Disclosure of the key-encryption key, the message-authentication key, or the content-authenticated-encryption key could allow modification of the associated authenticated content.

Additional considerations related to key management may be found in [SP.800-57pt1r5].

The generation of private keys relies on random numbers, as does the encapsulation function of Composite ML-KEM. The use of inadequate pseudorandom number generators (PRNGs) to generate these values can result in little or no security. If the random value is weakly chosen, then an attacker may find it much easier to reproduce the PRNG environment that produced the keys or ciphertext, searching the resulting small set of possibilities for a matching public key or ciphertext value, rather than performing a more complex algorithmic attack against Composite ML-KEM.

Composite ML-KEM encapsulation and decapsulation only outputs a shared secret and ciphertext. Implementations MUST NOT use intermediate values directly for any purpose.

Implementations SHOULD NOT reveal information about intermediate values or calculations, whether by timing or other "side channels"; otherwise an opponent may be able to determine information about the keying data and/or the recipient's private key. Although not all intermediate information may be useful to an opponent, it is preferable to conceal as much information as is practical, unless analysis specifically indicates that the information would not be useful to an opponent.

Generally, good cryptographic practice employs a given Composite ML-KEM key pair in only one scheme. This practice avoids the risk that vulnerability in one scheme may compromise the security of the other and may be essential to maintain provable security.

5. IANA Considerations

IANA is requested to allocate a value from the "SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0)" registry for the included ASN.1 module.

* Decimal: IANA Assigned - *Replace TBDMOD*

* Description: Composite-MLKEM-CMS-2026 - id-mod-composite-mlkem-cms-2026

* References: This Document

RFC EDITOR: Please replace TBDCOMPOSITEMOD in the ASN.1 module with with module number assigned to id-mod-composite-mlkem-2025 in [I-D.ietf-lamps-pq-composite-kem].

6. References

6.1. Normative References

- [FIPS180] "Secure hash standard", National Institute of Standards and Technology (U.S.), DOI 10.6028/nist.fips.180-4, 2015, <<https://doi.org/10.6028/nist.fips.180-4>>.
- [I-D.ietf-lamps-pq-composite-kem]
Ounsworth, M., Gray, J., Pala, M., Klaußner, J., and S. Fluhrer, "Composite ML-KEM for use in X.509 Public Key Infrastructure", Work in Progress, Internet-Draft, draft-ietf-lamps-pq-composite-kem-14, 27 March 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-pq-composite-kem-14>>.
- [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/rfc/rfc2119>>.
- [RFC3394] Schaad, J. and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm", RFC 3394, DOI 10.17487/RFC3394, September 2002, <<https://www.rfc-editor.org/rfc/rfc3394>>.
- [RFC3565] Schaad, J., "Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS)", RFC 3565, DOI 10.17487/RFC3565, July 2003, <<https://www.rfc-editor.org/rfc/rfc3565>>.
- [RFC5083] Housley, R., "Cryptographic Message Syntax (CMS) Authenticated-Enveloped-Data Content Type", RFC 5083, DOI 10.17487/RFC5083, November 2007, <<https://www.rfc-editor.org/rfc/rfc5083>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/rfc/rfc5280>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/rfc/rfc5652>>.
- [RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <<https://www.rfc-editor.org/rfc/rfc5869>>.

- [RFC5911] Hoffman, P. and J. Schaad, "New ASN.1 Modules for Cryptographic Message Syntax (CMS) and S/MIME", RFC 5911, DOI 10.17487/RFC5911, June 2010, <<https://www.rfc-editor.org/rfc/rfc5911>>.
- [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/rfc/rfc8174>>.
- [RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <<https://www.rfc-editor.org/rfc/rfc8551>>.
- [RFC8619] Housley, R., "Algorithm Identifiers for the HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 8619, DOI 10.17487/RFC8619, June 2019, <<https://www.rfc-editor.org/rfc/rfc8619>>.
- [RFC9629] Housley, R., Gray, J., and T. Okubo, "Using Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS)", RFC 9629, DOI 10.17487/RFC9629, August 2024, <<https://www.rfc-editor.org/rfc/rfc9629>>.
- [X680] ITU-T, "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T Recommendation X.680, ISO/IEC 8824-1:2021, February 2021, <<https://www.itu.int/rec/T-REC-X.680>>.
- [X690] ITU-T, "Information technology - Abstract Syntax Notation One (ASN.1): ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, ISO/IEC 8825-1:2021, February 2021, <<https://www.itu.int/rec/T-REC-X.690>>.

6.2. Informative References

- [FIPS203] "Module-lattice-based key-encapsulation mechanism standard", National Institute of Standards and Technology (U.S.), DOI 10.6028/nist.fips.203, August 2024, <<https://doi.org/10.6028/nist.fips.203>>.
- [RFC9690] Housley, R. and S. Turner, "Use of the RSA-KEM Algorithm in the Cryptographic Message Syntax (CMS)", RFC 9690, DOI 10.17487/RFC9690, February 2025, <<https://www.rfc-editor.org/rfc/rfc9690>>.

[RFC9936] Prat, J., Ounsworth, M., and D. Van Geest, "Use of ML-KEM in the Cryptographic Message Syntax (CMS)", RFC 9936, DOI 10.17487/RFC9936, March 2026, <<https://www.rfc-editor.org/rfc/rfc9936>>.

[SP.800-57pt1r5] National Institute of Standards and Technology (NIST), "Recommendation for Key Management: Part 1 â\200\223 General", May 2020, <<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf>>.

Appendix A. ASN.1 Module

This appendix includes the ASN.1 module [X680] for Composite ML-KEM. This module imports objects from [RFC5911], [RFC9629], [RFC8619], [I-D.ietf-lamps-pq-composite-kem].

<CODE BEGINS>

Composite-MLKEM-CMS-2026

```
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9)
  id-smime(16) id-mod(0)
  id-mod-composite-mlkem-cms-2026(TBDMOD) }
```

DEFINITIONS IMPLICIT TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS

SMIME-CAPS

```
FROM AlgorithmInformation-2009 -- [RFC5911]
{ iso(1) identified-organization(3) dod(6) internet(1)
  security(5) mechanisms(5) pkix(7) id-mod(0)
  id-mod-algorithmInformation-02(58) }
```

KEM-ALGORITHM

```
FROM KEMAlgorithmInformation-2023 -- [RFC9629]
{ iso(1) identified-organization(3) dod(6) internet(1)
  security(5) mechanisms(5) pkix(7) id-mod(0)
  id-mod-kemAlgorithmInformation-2023(109) }
```

kda-hkdf-with-sha256

```
FROM HKDF-OID-2019 -- [RFC8619]
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
  pkcs-9(9) smime(16) modules(0) id-mod-hkdf-oid-2019(68) }
```

kwa-aes256-wrap

```

FROM CMSAesRsaesOaep-2009 -- [RFC5911]
    { iso(1) member-body(2) us(840) rsadsi(113549)
      pkcs(1) pkcs-9(9) smime(16) modules(0)
      id-mod-cms-aes-02(38) }

kema-MLKEM768-RSA2048-SHA3-256, kema-MLKEM768-RSA3072-SHA3-256,
kema-MLKEM768-RSA4096-SHA3-256, kema-MLKEM768-X25519-SHA3-256,
kema-MLKEM768-ECDH-P256-SHA3-256, kema-MLKEM768-ECDH-P384-SHA3-256,
kema-MLKEM768-ECDH-brainpoolP256r1-SHA3-256,
kema-MLKEM1024-RSA3072-SHA3-256, kema-MLKEM1024-ECDH-P384-SHA3-256,
kema-MLKEM1024-ECDH-brainpoolP384r1-SHA3-256, kema-MLKEM1024-X448,
kema-MLKEM1024-ECDH-P521-SHA3-256
FROM Composite-MLKEM-2025 -- [I-D.ietf-lamps-pq-composite-kem]
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-composite-mlkem-2025(TBDCOMPOSITEMOD) }
;

-- Updates for the SMIME-CAPS Set from RFC 5911

SMimeCapsSet SMIME-CAPS ::=
{ kema-MLKEM768-RSA2048-SHA3-256.&smimeCaps |
  kema-MLKEM768-RSA3072-SHA3-256.&smimeCaps |
  kema-MLKEM768-RSA4096-SHA3-256.&smimeCaps |
  kema-MLKEM768-X25519-SHA3-256.&smimeCaps |
  kema-MLKEM768-ECDH-P256-SHA3-256.&smimeCaps |
  kema-MLKEM768-ECDH-P384-SHA3-256.&smimeCaps |
  kema-MLKEM768-ECDH-brainpoolP256r1-SHA3-256.&smimeCaps |
  kema-MLKEM1024-RSA3072-SHA3-256.&smimeCaps |
  kema-MLKEM1024-ECDH-P384-SHA3-256.&smimeCaps |
  kema-MLKEM1024-ECDH-brainpoolP384r1-SHA3-256.&smimeCaps |
  kema-MLKEM1024-X448.&smimeCaps |
  kema-MLKEM1024-ECDH-P521-SHA3-256.&smimeCaps |
  kda-hkdf-with-sha256.&smimeCaps |
  kwa-aes256-wrap.&smimeCaps,
  ... }

END
<CODE ENDS>

```

Appendix B. Composite ML-KEM CMS Authenticated-Enveloped-Data Example

This example shows the establishment of an AES-256 content-encryption key using:

- * id-MLKEM768-ECDH-P256-SHA3-256;
- * KEMRecipientInfo key derivation using HKDF with SHA-256; and

* KEMRecipientInfo key wrap using AES-256-KEYWRAP.

In real-world use, the originator would encrypt the content-encryption key in a manner that would allow decryption with their own private key as well as the recipient's private key. This is omitted in an attempt to simplify the example.

B.1. Originator CMS Processing

Alice obtains Bob's id-MLKEM768-ECDH-P256-SHA3-256 public key:

```
-----BEGIN PUBLIC KEY-----
MIIE8jAKBggrBgEFBQCgOwOCBOIAjssyhemrdzqTE5MNKmCruyoBancRo6y8A5Aw
uDSZ2AwdyWZCWnG6R2sP0vlXCNg1q7BvHQTF6BObo+tT+ZmJSnsylYaSKuyf8yw3
vxyrlScErsY0r2s1gfprFhYLUQNNQd+YViNTAedvTBeQGJLgJOozEIFKiwhbvcu
ddc2qnVGqNW0SSjF5QoE/lcwgvpYpZoshRePTYhm87hCdQJldwB6PZmgxZqfyeh0
vXsB+KGw+XLJaOgb2vt1+Nmbm5cyUvOgHItYxQIadh08iHEIubYzKdlOcDGIP1qE
lCiMNs1C+TGcl4Ij2SBQgyfLGpw2jpd3xJxyqjFZeeGrgidfRdVRtUyqUbCGHsLH
ITFJz/h9vqe+S5N6vJmHMFCTHtu0CSiz5pK/RepjWKZiWhYdz1k9u8Gwo4q3h8SH
MnQKoSJq3Fx84K1/ghUNxXB//2w+a/ekHuUzK8QbyWIqm2RSHkOj91hApCMG0tMn
IOxoEqm/ewo3xZA5MJUZEdIkmoN+d5fLwVlUOHS0TWVrrCWAvtq4HySFMBU3vGd
qGQpV2ZLd+NgjzmRMwc3unQh4KIXBspNK+lJgyU57QaDWtVq/yoXjhw5YQ1Ymckb
NuIeTjpNDqfNxFsuziVQwZQ6VPm/63tjcxy9uMtxw5wTyqfAkyZVKdTLUdcMsJVF
8XE0r6xtCQRNsmWzqhK1kvOUQcITlZEonyN4uXdfweFwXimg/NSK6QUcjKi6iQeI
GmiY2bKbUIgrIbhyUOZUlad/vrF+dkBzrPei5BanHBLF5hYVoJcQ+hmg3cVsR1Ke
XaduFdh0UXPNPMiTrClLz/FMdfcbzbe2xdSXHQcSdIBKxGagT3p0H/uTWlWwheFr
DJKBDYBaZxVOWRqP16M7QIOT08AqeXTOQMSxAdCAx7NR4SttZbQDrjwj1qCj5LQ9
0TW4vUmNsWkaTeQP0LC6f/J5v1EQeIu0vUuggkqmDdaKPwx35SOixjwEwiweXgil
3bdn68py4/VVYPwXHZaDhakUurZcNEU9lnwkrdG03DmbyBlIrwwIClgMwhmSbSGj
Mql2lCPPE/zBJAvL9lfDP3mpoEMB0iCzm6t2IfAN7NLOWFeBOUomqxcvs0OkdxFG
1Gk0hVEis4LHcje/1rBAX8i2Ige6H0AqY5e9X2OHHjVFNzoM8Mavo5trYEksanbG
aJmpADcX8wxjDPhq+VBethWo5Ik0aLWl6qGY/jAX6R0rpeRob0iSYNnDLFWI38Mf
cZJuHbBaJBe2gwQbTdadu0elr7fMuHd70OB35vgH7qq0lgOg7LcB8rVgXBeHLSJp
AGS2XTYlKFRNieIo5NSRdkXNkIoHyFZWU9dNUwbKD0CYtxxlj3cKp4AFFHdzCTot
bFIfhoQQxIsZhbpOFCZKjtEQ+UVnjQN1y0E9uym1N7ZjW9OkjeCs0YaooyooNpu6
TOMKu3GjVsqmKvgxk3cIztUouTC2+BC0iYANMUxRbZIIqLGWtsa6rBulSqm5Hpu8
R/LBY0gljpyLFdQJiuYRNvOOIUYEO3ubd8fKY8swLGLNi/LoNwtb+HoJ8szpDsXh
TtIalQ0MEsdePvv0LCZ+0N55b1MaFJZieRuCMVPkbeJGWGzb4WVohn4jrsslanc6y
eZxiXOnPS7EDyzLv2PL/zhedCEbC4A==
-----END PUBLIC KEY-----
```

Bob's id-MLKEM768-ECDH-P256-SHA3-256 public key has the following key identifier:

```
14F1D8FED21F6103676C752C97D0949537A96AB2
```

Alice generates a shared secret and ciphertext using Bob's id-MLKEM768-ECDH-P256-SHA3-256 public key:

Shared secret:

9bd51d44ba390ff68176b763cee113007f112aac3c2ddbde916e713762a389c

Ciphertext:

54b75c945ff83194fbf312214749b114bf6838878b49403f5235be774ae6719d
90531da9ab013a0f8a81dbd6510592f5fbcbe3b1bcf393793f517e758bbdd104
67c1a72b14165233e386ea6b027e482824d33244cc7f810bf7d21cd28e007c84
19087698be0dd7934653f35c8b0164c218dcbbcbe28302a813e47a334c9565349
0cf57602c798d63d393e9e6ebc219265aa6562632a618d4c3e4052b20c89f2f0
4ea45554673d1be8911f83ca4a20ce86b59cf0a8c964e5b04707875d56011d48
9291abf2370838dc608fe2e26e617d0ec2c2c643e51caf6716e64c2cc5a185c7
2770ec4185487589156bb47e414180c6cedead8dad07ffa9091441fdaced553
c482bd0c41d5b77649f0ed803d07d29682dc2c41d2047d18fddab96529b9f8a2
f0ddd5e4b302fffd8ccbec03d9917ffa6b8d249117146c58fd89a135a96ece67
0c58a559628a68012d4c9a43e5fffb427f0df30da745bb0e4eb334716bb82175
10467b8b58e1be54b30c857522654cbb5787813cb1b2fb957a725f51612187b5
8bb83353f8e56aa6af89c94edd656d92bf66913022fce8f85730617d8a166bbb
24515559915cc472435950bce8c25219fef507a6527a6a3920c2deada7732902
e2d2ae8c60bda518a65da66f93581b7e37042a10464675ab326f603be14e5b55
25634adeb956512a84fdb809af2f37d20a9fd23f67eca7741b49d5b7d74f947a
8b6dac68888b3e3802994c98cccb68c2a0e7927f15e5e7be1400983e22a4c9ef
2babdf244525adde3b4d91e778e21e2bd32c5221f5433f28390128eabf3b181b
85c2clefd7fae946399bb8fe21de2aed0a3e72cab34d305509547df33a45d17d
013a3ac08f0db69dbff0bae5a4b9c15ee8bdc4e2525b49e6865aeb752051935f
8e2fb5daea912d118671102b8fb4675c3937b389a7c7068cb697b3798fffe9d6
e64156493bb808d067a3eac418a7bf5819e25a740c8349914e5008340c381a07
d870b14b9be9939d306bc702d46a58216df932d2bcb45a3f181bd84f4b9ba992
f3a5f12a5db615ae0598a9c432cf5c0095187a849a93ca0d2d7deedb2db1ced5
3c0d4ae5d2b37eb25e07992452e0188e2d72682f46f0167f0f6a680243eff2c4
b5fcf0235852f99defa4d535a479176fa3a22587511581b3649fe410e302ec1b
9061cb535c492ae57ac126cb49ea47de12a9097c5f8a869d84d4aae903f0bc6f
53a60514cebe02fdc9a204a6be2a66567d589fcbaca36565a3e02d10b16920ea
26b05c50be805d06ece3c9ee7db7eb207c33c1d4ac92ca294f1a0f8fac839fea
0389fae43384a2daa7a4c1f5e92d622f8eb37ac0c55eb9df8d9713de03a0ef8b
2d9306c8530f607d898ba88adb6ad982becd3b05cac32b7dfe767c826e5e8bdf
40120f1a49e84cd11c4c09b07c27591a6032b372a7bd468b09ece07ee6eca6e9
4559f14843dd9a97931e0c06d105aa2237e67a3c128d33ab61dc47986fdafb79
7e60cb44453b4dfbcf7afdf02f686c720fd65d5381827be62f3322acbc37213
0443d6690e3ae3b17f263f9345998ca26317f757b86d0ad41531b114f5d57fa8
2e5023e5176227d087f765e1421cef31be9c315866838017bb2aa58955bf52ce
6e

Alice encodes the CMSORIforKEMOtherInfo:

3010300b060960864801650304012d020120

Alice derives the key-encryption key from the shared secret and CMSORInfo for KEMOtherInfo using HKDF with SHA-256:

```
0555324e4703e672404e2272c555438bd7d04feac1321e05ebae6c0a2336d529
```

Alice randomly generates a 128-bit content-encryption key:

```
3424cef0cbc2b67f58351abd1e87b3507a11c90adef7e1cdab8d20b0331b8105
```

Alice uses AES-256-KEYWRAP to encrypt the content-encryption key with the key-encryption key:

```
5d13ae00d914a29158a8ef32b23ae5f27b8c293387e3be65fc3dfc19b99d8fdc  
552527fd42154b37
```

Alice encrypts the padded content using AES-256-GCM with the content-encryption key and encodes the AuthEnvelopedData (using KEMRecipientInfo) and ContentInfo, and then sends the result to Bob.

The Base64-encoded result is:

-----BEGIN CMS-----

```

MIIFcgYLKozIhvcNAQkQARegggVhMIIFXQIBADGCBQikggUEBgsqhkIG9w0BCRAN
AzCCBPMCAQCAFBTx2P7SH2EDZ2x1LJfQlJU3qWqyMAoGCCsGAQUFBwY7BIIegVS3
XJRf+DGU+/MSIUdJsRS/aDiHi0lAP1I1vndK5nGdkFMdqasBOg+KgdvWUQWS9fvL
47G885N5P1F+dYu90QRnwacrFBZSM+OG6msCfkgoJNMyRMx/gQv30hzSjgB8hBkI
dpi+DdeTRlPzXISBZMIY3Ly+KDAqgT5HozTJVLNJDpV2AseY1j05Pp5uvCGSZap1
YmMqYY1MPkBSsgyJ8vBOPFVUZz0b6JEfg8pKIM6GtZzwqMlk5bBHB4ddVgEdSJKR
q/I3CDjcYI/i4m5hfQ7CwsZD5RyvZxbmTCzFoYXHJ3DsQYVIdYkVa7R+QUGAxs7e
rNja0H/6kJFEH9r0lVPEgr0MQdW3dknw7YA9B9KWgtwsQdIEfRj92r1lKbn4ovDd
leSzAv/9jMvsA9mRf/prjSSRFxRsWP2JoTWpbs5nDFilWWKKaAETtJpD5f/7Qn8N
8w2nRbsOTRm0cWu4IXUQRnuLWOG+VLMMhXUiZUy7V4eBPLGy+5V6cl9RYSgHtYu4
MlP45Wqmr4nJtT1lbZK/ZpEwIvzo+FcwYX2KFmu7JFFVWZFCxHJDWVC86MJSgf71
B6ZSemo5IMLeradzKQLi0q6MYL2lGKZdpm+TWBt+NwQqEEZGdasYb2A74U5bVSVj
St65VlEqhP24Ca8vN9IKn9I/Z+yndBtJlbfXT5R6i22saIiLPjgCmUyYzMtowqDn
kn8V5ee+FACYPiKkye8rq98kRSWt3jtNked44h4r0yxSIfVDPyg5ASjqvzsYG4XC
we/X+ulGOZu4/iHeKu0KPnLKs00wVQlUffm6RdF9ATo6wI8Ntp2/8LrlpLnBXui9
xOJSW0nmhlrrdSBRk1+OL7Xa6pEtEYZxECuPtGdcOTeziafHBoy2l7N5j//p1uZB
Vkk7uAjqZ6PqxBinvlGZ4lp0DINJkU5QCDQMOBoH2HCxS5vpk50wa8cC1GpYIW35
MtK8tFo/GBvYT0ubqZLzpEqXbYVrgWYqcQyz1wAlRh6hJqTyg0tfe7bLbH01TwN
SuXSs36yXgeZJfLgGI4tcmgvRvAWfw9qaAJD7/LEtfzWI1hS+Z3vpNULpHkXb60i
JYdRFYgZzJ/kEOMC7BuQYctTXEkq5XrBJstJ6kfeEqkJfF+Khp2E1KrpA/C8b1Om
BRT0vgL9yaIEpr4qZlZ9WJ/LrKNlZaPgLRcXaSDqJrBcUL6AXQbs48nufbfrIHwz
wdSksopTxoPj6yDn+oDifrkM4Si2qekwfXpLWivjrN6wMVeud+NlxPeA6Dviy2T
BshTD2B9iYuoittq2YK+zTsFysMrff52fIJuXovfQBIPGknoTNECTAmwfCdZGmAy
s3KnvUaLCezgfubspulFWfFIQ92al5MeDabRBaoiN+Z6PBKNM6th3EeYb9r7eX5g
y0RFO037z3r9/gL2hscg/WXVOBgnvmLzMirLw3ITBEPWaq4647F/Jj+TRZmMomMX
91e4bQrUFTGxFPXVf6guUCPlf2In0If3ZeFCHO8xvpwxWGaDgBe7KqWJVb9Szm4w
DQYLKozIhvcNAQkQAxwCASAawCwYJYIZIAWUDBAETBChdE64A2RSikVio7zKyOuXy
e4wpM4fjvmX8PfwZu22P3FULJ/1CFUs3MDoGCSqGSIB3DQEHATAeBglghkgBZQME
AS4wEQQMhrwH+8PcSlmdkPPoAgEQgA0g+U8GH3sF+VsyV//ABBBfKmBjryzwX+Hi
16KLp6sQ

```

-----END CMS-----

This result decodes to:

```

0 1394: SEQUENCE {
4   11: OBJECT IDENTIFIER
      : authEnvelopedData (1 2 840 113549 1 9 16 1 23)
17 1377: [0] {
21 1373: SEQUENCE {
25   1: INTEGER 0
28 1288: SET {
32 1284: [4] {
36   11: OBJECT IDENTIFIER '1 2 840 113549 1 9 16 13 3'
49 1267: SEQUENCE {
53   1: INTEGER 0
56  20: [0]
      : 14 F1 D8 FE D2 1F 61 03 67 6C 75 2C 97 D0 94 95

```

```
      : 37 A9 6A B2
78    10: SEQUENCE {
80      8: OBJECT IDENTIFIER '1 3 6 1 5 5 7 6 59'
      : }
90 1153: OCTET STRING
      : 54 B7 5C 94 5F F8 31 94 FB F3 12 21 47 49 B1 14
      : BF 68 38 87 8B 49 40 3F 52 35 BE 77 4A E6 71 9D
      : 90 53 1D A9 AB 01 3A 0F 8A 81 DB D6 51 05 92 F5
      : FB CB E3 B1 BC F3 93 79 3F 51 7E 75 8B BD D1 04
      : 67 C1 A7 2B 14 16 52 33 E3 86 EA 6B 02 7E 48 28
      : 24 D3 32 44 CC 7F 81 0B F7 D2 1C D2 8E 00 7C 84
      : 19 08 76 98 BE 0D D7 93 46 53 F3 5C 8B 01 64 C2
      : 18 DC BC BE 28 30 2A 81 3E 47 A3 34 C9 56 53 49
      : 0C F5 76 02 C7 98 D6 3D 39 3E 9E 6E BC 21 92 65
      : AA 65 62 63 2A 61 8D 4C 3E 40 52 B2 0C 89 F2 F0
      : 4E A4 55 54 67 3D 1B E8 91 1F 83 CA 4A 20 CE 86
      : B5 9C F0 A8 C9 64 E5 B0 47 07 87 5D 56 01 1D 48
      : 92 91 AB F2 37 08 38 DC 60 8F E2 E2 6E 61 7D 0E
      : C2 C2 C6 43 E5 1C AF 67 16 E6 4C 2C C5 A1 85 C7
      : 27 70 EC 41 85 48 75 89 15 6B B4 7E 41 41 80 C6
      : CE DE AC D8 DA D0 7F FA 90 91 44 1F DA CE D5 53
      : C4 82 BD 0C 41 D5 B7 76 49 F0 ED 80 3D 07 D2 96
      : 82 DC 2C 41 D2 04 7D 18 FD DA B9 65 29 B9 F8 A2
      : F0 DD D5 E4 B3 02 FF FD 8C CB EC 03 D9 91 7F FA
      : 6B 8D 24 91 17 14 6C 58 FD 89 A1 35 A9 6E CE 67
      : 0C 58 A5 59 62 8A 68 01 2D 4C 9A 43 E5 FF FB 42
      : 7F 0D F3 0D A7 45 BB 0E 4E B3 34 71 6B B8 21 75
      : 10 46 7B 8B 58 E1 BE 54 B3 0C 85 75 22 65 4C BB
      : 57 87 81 3C B1 B2 FB 95 7A 72 5F 51 61 21 87 B5
      : 8B B8 33 53 F8 E5 6A A6 AF 89 C9 4E DD 65 6D 92
      : BF 66 91 30 22 FC E8 F8 57 30 61 7D 8A 16 6B BB
      : 24 51 55 59 91 5C C4 72 43 59 50 BC E8 C2 52 19
      : FE F5 07 A6 52 7A 6A 39 20 C2 DE AD A7 73 29 02
      : E2 D2 AE 8C 60 BD A5 18 A6 5D A6 6F 93 58 1B 7E
      : 37 04 2A 10 46 46 75 AB 32 6F 60 3B E1 4E 5B 55
      : 25 63 4A DE B9 56 51 2A 84 FD B8 09 AF 2F 37 D2
      : 0A 9F D2 3F 67 EC A7 74 1B 49 D5 B7 D7 4F 94 7A
      : 8B 6D AC 68 88 8B 3E 38 02 99 4C 98 CC CB 68 C2
      : A0 E7 92 7F 15 E5 E7 BE 14 00 98 3E 22 A4 C9 EF
      : 2B AB DF 24 45 25 AD DE 3B 4D 91 E7 78 E2 1E 2B
      : D3 2C 52 21 F5 43 3F 28 39 01 28 EA BF 3B 18 1B
      : 85 C2 C1 EF D7 FA E9 46 39 9B B8 FE 21 DE 2A ED
      : 0A 3E 72 CA B3 4D 30 55 09 54 7D F3 3A 45 D1 7D
      : 01 3A 3A C0 8F 0D B6 9D BF F0 BA E5 A4 B9 C1 5E
      : E8 BD C4 E2 52 5B 49 E6 86 5A EB 75 20 51 93 5F
      : 8E 2F B5 DA EA 91 2D 11 86 71 10 2B 8F B4 67 5C
      : 39 37 B3 89 A7 C7 06 8C B6 97 B3 79 8F FF E9 D6
      : E6 41 56 49 3B B8 08 D0 67 A3 EA C4 18 A7 BF 58
```

```

: 19 E2 5A 74 0C 83 49 91 4E 50 08 34 0C 38 1A 07
: D8 70 B1 4B 9B E9 93 9D 30 6B C7 02 D4 6A 58 21
: 6D F9 32 D2 BC B4 5A 3F 18 1B D8 4F 4B 9B A9 92
: F3 A5 F1 2A 5D B6 15 AE 05 98 A9 C4 32 CF 5C 00
: 95 18 7A 84 9A 93 CA 0D 2D 7D EE DB 2D B1 CE D5
: 3C 0D 4A E5 D2 B3 7E B2 5E 07 99 24 52 E0 18 8E
: 2D 72 68 2F 46 F0 16 7F 0F 6A 68 02 43 EF F2 C4
: B5 FC F0 23 58 52 F9 9D EF A4 D5 35 A4 79 17 6F
: A3 A2 25 87 51 15 81 B3 64 9F E4 10 E3 02 EC 1B
: 90 61 CB 53 5C 49 2A E5 7A C1 26 CB 49 EA 47 DE
: 12 A9 09 7C 5F 8A 86 9D 84 D4 AA E9 03 F0 BC 6F
: 53 A6 05 14 CE BE 02 FD C9 A2 04 A6 BE 2A 66 56
: 7D 58 9F CB AC A3 65 65 A3 E0 2D 10 B1 69 20 EA
: 26 B0 5C 50 BE 80 5D 06 EC E3 C9 EE 7D B7 EB 20
: 7C 33 C1 D4 AC 92 CA 29 4F 1A 0F 8F AC 83 9F EA
: 03 89 FA E4 33 84 A2 DA A7 A4 C1 F5 E9 2D 62 2F
: 8E B3 7A C0 C5 5E B9 DF 8D 97 13 DE 03 A0 EF 8B
: 2D 93 06 C8 53 0F 60 7D 89 8B A8 8A DB 6A D9 82
: BE CD 3B 05 CA C3 2B 7D FE 76 7C 82 6E 5E 8B DF
: 40 12 0F 1A 49 E8 4C D1 1C 4C 09 B0 7C 27 59 1A
: 60 32 B3 72 A7 BD 46 8B 09 EC E0 7E E6 EC A6 E9
: 45 59 F1 48 43 DD 9A 97 93 1E 0C 06 D1 05 AA 22
: 37 E6 7A 3C 12 8D 33 AB 61 DC 47 98 6F DA FB 79
: 7E 60 CB 44 45 3B 4D FB CF 7A FD FE 02 F6 86 C7
: 20 FD 65 D5 38 18 27 BE 62 F3 32 2A CB C3 72 13
: 04 43 D6 69 0E 3A E3 B1 7F 26 3F 93 45 99 8C A2
: 63 17 F7 57 B8 6D 0A D4 15 31 B1 14 F5 D5 7F A8
: 2E 50 23 E5 17 62 27 D0 87 F7 65 E1 42 1C EF 31
: BE 9C 31 58 66 83 80 17 BB 2A A5 89 55 BF 52 CE
: 6E
1247 13: SEQUENCE {
1249 11:   OBJECT IDENTIFIER
:       hkdfWithSha256 (1 2 840 113549 1 9 16 3 28)
:   }
1262 1:   INTEGER 32
1265 11: SEQUENCE {
1267 9:   OBJECT IDENTIFIER
:       aes256-wrap (2 16 840 1 101 3 4 1 45)
:   }
1278 40: OCTET STRING
: 5D 13 AE 00 D9 14 A2 91 58 A8 EF 32 B2 3A E5 F2
: 7B 8C 29 33 87 E3 BE 65 FC 3D FC 19 B9 9D 8F DC
: 55 25 27 FD 42 15 4B 37
:   }
:   }
:   }
1320 58: SEQUENCE {
1322 9:   OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)

```

```

1333 30: SEQUENCE {
1335 9: OBJECT IDENTIFIER
      : aes256-GCM (2 16 840 1 101 3 4 1 46)
1346 17: SEQUENCE {
1348 12: OCTET STRING 86 BC 07 FB C3 DC 4A 59 9D 90 F3 E8
1362 1: INTEGER 16
      : }
      : }
1365 13: [0] 20 F9 4F 06 1F 7B 05 F9 5B 32 57 FF C0
      : }
1380 16: OCTET STRING
      : 5F 2A 60 63 AF 2C F0 5F E1 E2 97 A2 8B A7 AB 10
      : }
      : }

```

B.2. Recipient CMS Processing

Bob's id-MLKEM768-ECDH-P256-SHA3-256 private key:

```

-----BEGIN PRIVATE KEY-----
MIGEAgEAMaGCCCsgAQUFBwY7BHOImiTrgVkiRNnS3EmMdxHUrH+EHeFlRSSQaMrG
7NAvf0DDSVt58hvzJz/RuCOE4/8REOs/DZVr5gWO2jOyldClMDECAQEEIM/ctKyh
CrQyBNhBbpwS5ZkEOlmlf14j5NRtwj3obB+oAoGCCqGSM49AWEH
-----END PRIVATE KEY-----

```

Bob decapsulates the ciphertext in the KEMRecipientInfo to get the MLKEM768-ECDH-P256-SHA3-256 shared secret, encodes the CMSORInfo for KEMOtherInfo, derives the key-encryption key from the shared secret and the DER-encoded CMSORInfo for KEMOtherInfo using HKDF with SHA-256, uses AES-256-KEYWRAP to decrypt the content-encryption key with the key-encryption key, and decrypts the encrypted contents with the content-encryption key, revealing the plaintext content:

(Artwork only available as TEST-VECTORS: see <https://www.ietf.org/archive/id/draft-ietf-lamps-cms-composite-kem-01.html>)

Appendix C. Acknowledgments

This document borrows heavily from [RFC9690] and [RFC9936]. Thanks go to the authors of those documents. "Copying always makes things easier and less error prone" - RFC8411.

Authors' Addresses

Daniel Van Geest
CryptoNext Security
â\200\21516, Boulevard Saint-Germain
75007 Paris
France
Email: daniel.vangeest@cryptonext-security.com

Mike Ounsworth
Entrust Limited
2500 Solandt Road â\200\223 Suite 100
Ottawa, Ontario K2K 3G5
Canada
Email: mike.ounsworth@entrust.com

John Gray
Entrust Limited
2500 Solandt Road â\200\223 Suite 100
Ottawa, Ontario K2K 3G5
Canada
Email: john.gray@entrust.com

Jan Klaussner
Bundesdruckerei GmbH
Kommandantenstr. 18
10969 Berlin
Germany
Email: jan.klaussner@bdr.de

Network Working Group
Internet-Draft
Obsoletes: 6831 (if approved)
Intended status: Standards Track
Expires: 26 July 2026

D. Farinacci
lispers.net
D. Meyer
J. Zwiebel
S. Venaas
V. Govindan
Cisco Systems
22 January 2026

The Locator/ID Separation Protocol (LISP) for Multicast Environments
draft-ietf-lisp-rfc6831bis-06

Abstract

This document describes the design for inter-domain multicast overlays using the Locator/ID Separation Protocol (LISP) architecture and protocols. The document specifies how LISP multicast overlays operate over multicast and unicast underlays. The mechanisms in this specification indicate how a signal-based approach using the PIM protocol can be used to program LISP encapsulators with a replication list in a locator-set, where the replication list can be a mix of multicast and unicast locators. This document when approved obsoletes RFC6831

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 26 July 2026.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Requirements Notation	5
3. Definition of Terms	5
4. Basic Overview	8
5. Source Addresses versus Group Addresses	11
6. Locator Reachability Implications on LISP-Multicast	12
7. Multicast Protocol Changes	12
7.1. PIM Join Attributes for LISP	15
8. LISP-Multicast Data-Plane Architecture	15
8.1. ITR Forwarding Procedure	15
8.1.1. Multiple RLOCs for an ITR	16
8.1.2. Multiple ITRs for a LISP Source Site	16
8.2. ETR Forwarding Procedure	16
8.3. Replication Locations	17
9. LISP-Multicast Interworking	17
9.1. LISP and Non-LISP Mixed Sites	18
9.1.1. LISP Source Site to Non-LISP Receiver Sites	19
9.1.2. Non-LISP Source Site to Non-LISP Receiver Sites	21
9.1.3. Non-LISP Source Site to Any Receiver Site	21
9.1.4. Unicast LISP Source Site to Any Receiver Sites	22
9.1.5. LISP Source Site to Any Receiver Sites	22
9.2. LISP Sites with Mixed Address Families	23
9.3. Making a Multicast Interworking Decision	25
10. Considerations When RP Addresses Are Embedded in Group Addresses	25
11. Taking Advantage of Upgrades in the Underlay	26
12. Mtrace Considerations	26
13. Security Considerations	26
14. IANA Considerations	26
15. Acknowledgments	27
16. References	27
16.1. Normative References	27
16.2. Informative References	29
Appendix A. Document Change Log	30
A.1. Changes to draft-ietf-lisp-rfc6831bis-03	30
A.2. Changes to draft-ietf-lisp-rfc6831bis-03	30
A.3. Changes to draft-ietf-lisp-rfc6831bis-02	30

A.4. Changes to draft-ietf-lisp-rfc6831bis-01	30
A.5. Changes to draft-ietf-lisp-rfc6831bis-00	31
A.6. Changes to draft-farinacci-lisp-rfc6831bis-02	31
A.7. Changes to draft-farinacci-lisp-rfc6831bis-01	31
A.8. Changes to draft-farinacci-lisp-rfc6831bis-00	31
Authors' Addresses	32

1. Introduction

The Locator/ID Separation Protocol [RFC9300] architecture provides a mechanism to separate out Identification and Location semantics from the current definition of an IP address. By creating two namespaces, an Endpoint ID (EID) namespace used by sites and a Routing Locator (RLOC) namespace used by underlay routing, the underlay routing infrastructure can scale by doing topological aggregation of routing information.

Since LISP creates a new namespace, a mapping function must exist to map a site's EID-Prefixes to its associated Locators. For unicast packets, both the source address and destination address must be mapped. For multicast packets, only the source address needs to be mapped. The destination group address doesn't need to be mapped because the semantics of an IPv4 or IPv6 group address are logical in nature and not topology dependent. Therefore, this specification focuses on mapping a source EID address of a multicast flow during distribution tree setup and packet delivery.

This specification will address the following scenarios:

1. How a multicast source host in a LISP site sends multicast packets to receivers inside of its site as well as to receivers in other sites that are LISP enabled.
2. How inter-domain (or between LISP sites) multicast distribution trees are built and how forwarding of multicast packets leaving a source site toward receivers sites is performed. While the design principles of how such trees are built and maintained are the same, there are some procedural differences based on whether the underlay is unicast or multicast
3. What protocols are affected and what changes are required to such multicast protocols.
4. How ASM-mode (Any Source Multicast), SSM-mode (Single Source Multicast), and Bidir-mode (Bidirectional Shared Trees) service models will operate.

5. How multicast packet flow will occur for multiple combinations of LISP-enabled and non-LISP-enabled source and receiver sites. For example:
 - a. How multicast packets from a source host in a LISP site are sent to receivers in other sites when they are all non-LISP sites.
 - b. How multicast packets from a source host in a LISP site are sent to receivers in both LISP-enabled sites and non-LISP sites.
 - c. How multicast packets from a source host in a non-LISP site are sent to receivers in other sites when they are all LISP-enabled sites.
 - d. How multicast packets from a source host in a non-LISP site are sent to receivers in both LISP-enabled sites and non-LISP sites.

This specification focuses on what changes are needed to the multicast routing protocols to support LISP-Multicast as well as other protocols used for inter-domain multicast, such as Multiprotocol BGP (MBGP) [RFC4760]. The approach proposed in this specification requires no packet format changes to the protocols and no operational procedural changes to the multicast infrastructure inside of a site when all sources and receivers reside in that site, even when the site is LISP enabled. That is, internal operation of multicast is unchanged, regardless of whether or not the site is LISP enabled or whether or not receivers exist in other sites that are LISP enabled.

Therefore, there are only operational (and not protocol) changes for PIM-ASM [RFC7761], Multicast Source Discovery Protocol (MSDP) [RFC3618], Anycast-RP [RFC4610], and PIM-SSM [RFC4607]. BIDIR-PIM [RFC5015], which typically does not run in an inter-domain environment, is not addressed in depth in this document.

The scope of this document does not cover multicast-based Traffic Engineering (TE) relative to the TE-ITR (TE-based Ingress Tunnel Router) and TE-ETR (TE-based Egress Tunnel Router) descriptions in [RFC9300]. Similarly, specifying the detailed behavior for multicast Proxy-ITRs (mPITRs) (Section 9.1.3), mtrace (Section 12), and locator reachability (Section 6) is outside the scope of this document.

Deployments have successfully demonstrated the successful minimization of multicast state in the underlay. The cost of this optimization is the extra amount of multicast packets that get sent in the same underlay tunnel.

Issues and concerns about the deployment of LISP for Internet traffic are discussed in [RFC9300].

2. Requirements Notation

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] when, and only when, they appear in all capitals, as shown here.

3. Definition of Terms

The terminology in this section is consistent with the definitions in [RFC9300] but is extended specifically to deal with the application of the terminology to multicast routing.

LISP-Multicast: a reference to the design in this specification.

That is, when any site that is participating in multicast communication has been upgraded to be a LISP site, the operation of control-plane and data-plane protocols is considered part of the LISP-Multicast architecture.

Endpoint ID (EID): As defined in [RFC9300], a 32-bit (for IPv4) or 128-bit (for IPv6) value used in the source address field of the first (most inner) LISP header of a multicast packet. The host obtains a destination group address the same way it obtains one today, as it would when it is a non-LISP site. The source EID is obtained via existing mechanisms used to set a host's "local" IP address. An EID is allocated to a host from an EID-Prefix block associated with the site in which the host is located. An EID can be used by a host to refer to another host, as when it joins an SSM (S-EID,G) route using IGMP version 3 [RFC3376] or Multicast Listener Discovery Version 2 [RFC3810] [RFC4604]. LISP uses Provider-Independent (PI) blocks for EIDs; such EIDs MUST NOT be used as LISP RLOCs.

Routing Locator (RLOC): as defined in [RFC9300]

Ingress Tunnel Router (ITR): as defined in [RFC9300]. Additionally, it is a router that accepts an IP multicast packet with a single IP header (more precisely, an IP packet that does not contain a LISP header). The router treats this "inner" IP destination

multicast address opaquely so it doesn't need to perform a map lookup on the group address because it is topologically insignificant. The router then prepends an "outer" IP header with one of its globally routable RLOCs as the source address field. This RLOC is known to other multicast receiver sites that have used the mapping database to join a multicast tree for which the ITR is the root. In general, an ITR receives IP packets from site end-systems on one side and sends LIISP-encapsulated multicast IP packets out all external interfaces that have been joined.

An ITR would receive a multicast packet from a source inside of its site when 1) it is on the path from the multicast source to internally joined receivers, or 2) when it is on the path from the multicast source to externally joined receivers.

Egress Tunnel Router (ETR): as defined in [RFC9300]. Additionally it is a router that is on the path from a multicast source host in another site to a multicast receiver in its own site. An ETR accepts a PIM Join/Prune message from a site-internal PIM router destined for the source's EID in the multicast source site. The ETR maps the source EID in the Join/Prune message to an RLOC address based on the EID-to-RLOC mapping. This sets up the ETR to accept multicast encapsulated packets from the ITR in the source multicast site. A multicast ETR decapsulates multicast encapsulated packets and replicates them on interfaces leading to internal receivers.

xTR: as defined in [RFC9300].

LIISP Header: as defined in [RFC9300]. A LIISP-encapsulated multicast packet will have an "inner" header with the source EID in the source field, an "outer" header with the source RLOC in the source field, and the same globally unique group address in the destination field of both the inner and outer header.

(S,G) State: the formal definition is in the PIM Sparse Mode [RFC7761] specification. For this specification, the term is used generally to refer to multicast state. Based on its topological location, the (S,G) state that resides in routers can be either (S-EID,G) state (at a location where the (S,G) state resides) or (S-RLOC,G) state (in the Internet underlay).

(S-EID,G) State: refers to multicast state in multicast source and receiver sites where S-EID is the IP address of the multicast source host (its EID). An S-EID can appear in an IGMPv3 report, an MSDP SA message or a PIM Join/Prune message that travels inside of a site.

(S-RLOC,G) State: refers to multicast state in the underlay where S is a source locator (the IP address of a multicast ITR) of a site with a multicast source. The (S-RLOC,G) is mapped from the (S-EID,G) entry by doing a mapping database lookup for the EID-Prefix that S-EID maps to. An S-RLOC can appear in a PIM Join/Prune message when it travels from an ETR to an ITR over the Internet underlay.

uLISP Site: a unicast-only LISP site according to [RFC9300] that has not deployed the procedures of this specification and, therefore, for multicast purposes, follows the procedures from Section 9. A uLISP site can be a traditional multicast site.

LISP Site: as defined in [RFC9300], a unicast LISP site (uLISP Site) that is also multicast capable according to the procedures in this specification.

mPETR: this is a multicast proxy-ETR that is responsible for advertising a very coarse EID-Prefix to which non-LISP and uLISP sites can target their (S-EID,G) PIM Join/Prune messages. mPETRs are used so LISP source multicast sites can send multicast packets using source addresses from the EID namespace. mPETRs act as Proxy-ETRs for supporting multicast routing in a LISP infrastructure. It is likely a unicast PITR [RFC6832] hereafter referred to as uPITR in this document and an mPETR will be co-located since the single device advertises a coarse EID-Prefix in the underlying unicast routing system.

Mixed Locator-Sets: this is a Locator-Set for a LISP database mapping entry where the RLOC addresses in the Locator-Set are in both IPv4 and IPv6 format.

Unicast LISP Encapsulated PIM Join/Prune Message: this is a standard PIM Join/Prune message (LISP-encapsulated with destination UDP port 4341) that is sent by ETRs at multicast receiver sites to an ITR at a multicast source site. This message is sent periodically as long as there are interfaces in the OIF-list for the (S-EID,G) entry for which the ETR is joining. A multicast ETR LISP encapsulates PIM Join/Prune messages so PIM signalling is achieved between the ETR and ITR.

OIF-list: this is [RFC7761] notation to describe the outgoing interface list a multicast router stores per multicast routing table entry so it knows on which interfaces to replicate multicast packets.

RPF: Reverse Path Forwarding is a procedure used by multicast

routers as described in [RFC7761]. A router will accept a multicast packet for forwarding if the packet was received on the path that the router would use to forward unicast packets to the multicast packet's source.

4. Basic Overview

LISP, when used for unicast routing, increases the site's ability to control ingress traffic flows. Egress traffic flows are controlled by the IGP in the source site. For multicast, the IGP coupled with PIM can decide which path multicast packets ingress. By using the Traffic Engineering features of LISP [RFC9300], a multicast source site can control the egress of its multicast traffic. By controlling the priorities of Locators from a mapping database entry, a source multicast site can control which way multicast receiver sites join to the source site.

The fundamental multicast forwarding model is to encapsulate a multicast packet into another multicast packet or a unicast packet. An ITR will encapsulate multicast packets received from sources that it serves in a LISP-Multicast header. The destination group address from the inner header is copied to the destination address of the outer header. Alternately, the destination address of the outer header could be the RLOC of the ETR. The inner source address is the EID of the multicast source host and the outer source address is the RLOC of the encapsulating ITR.

The LISP-Multicast architecture will follow this high-level protocol and operational sequence:

1. Receiver hosts in multicast sites will join multicast content the way they do today -- they use IGMP. When they use IGMPv3 where they specify source addresses, they use source EIDs; that is, they join (S-EID,G). If the multicast source is external to this receiver site, the PIM Join/Prune message flows toward the ETRs.
2. The ETR does a mapping database lookup for S-EID. If the mapping is cached from a previous lookup (from either a previous Join/Prune for the source multicast site or a unicast packet that went to the site), it will use the RLOC information from the mapping. The ETR will use the same priority and weighting mechanism as for unicast. So, the source site can decide which way multicast packets egress similar to the way unicast packets flow.

3. When using a multicast underlay, the ETR will build two PIM Join/Prune messages, one that contains an (S-EID,G) entry that is unicast to the ITR that matches the RLOC the ETR selects, and the other that contains an (S-RLOC,G) entry so the underlay network can create multicast state from this ETR to the ITR.
4. When using a unicast underlay, the ETR will build one PIM Join/Prune message, one that contains an (S-EID,G) entry that is unicast to the ITR that matches the RLOC the ETR selects. This PIM Join/ Prune message contains enough information to help the ITR build ingress-replicated unicast trees.
5. When the ITR gets the unicast Join/Prune message (see Section 3 for formal definition), it will process (S-EID,G) entries in the message and propagate them inside of the site where it has explicit routing information for EIDs via the IGP.
6. When using the multicast underlay the ITR receives the (S-RLOC,G) PIM Join/Prune message, it will process it like any other join it would get in today's Internet. The S-RLOC address is the IP address of this ITR.
7. When using the unicast underlay the ITR create an explicitly tracked list of unicast RLOC addresses that have joined the tree. This information is used for ingress-replication.
8. At this point, there is (S-EID,G) state from the joining host in the receiver multicast site to the ETR of the receiver multicast site. There is (S-RLOC,G) state across the underlay network from the ETR of the multicast receiver site to the ITR in the multicast source site and (S-EID,G) state in the source multicast site. Note, the (S-EID,G) state is the same S-EID in each multicast site. As other ETRs join the same multicast tree, they can join through the same ITR (in which case the packet replication is done in the underlay) or a different ITR (in which case the packet replication is done at the source site).
9. When a packet is originated by the multicast host in the source site, the packet will flow to one or more ITRs that will prepend a LIISP header.
10. For the unicast underlay, the ITR makes individual copies for every ETR, the outer destination is set to the S-RLOC of the ETR and the outer source address is set to the S-RLOC of the ITR.

11. For the multicast underlay, the ITR copies the group address to the outer destination address field, the ITR inserts its own locator address in the outer source address field. The ITR will look at its (S-RLOC,G) state, where S-RLOC is its own locator address, and replicate the packet on each interface on which an (S-RLOC,G) join was received. The underlay has (S-RLOC,G) so where fan-out occurs to multiple sites, a underlay router will do packet replication.
12. When either the source site or the underlay replicates the packet, the ETR will receive a LIISP packet with a destination group address. It will decapsulate packets because it has receivers for the group. Otherwise, it would not have received the packets because it would not have joined. The ETR decapsulates and does an (S-EID,G) lookup in its multicast Forwarding Information Base (FIB) to forward packets out one or more interfaces to forward the packet to internal receivers.

This architecture is consistent and scalable with the architecture presented in [RFC9300] where multicast state in the underlay operates on Locators, and multicast state at the sites operates on EIDs.

Alternatively, [RFC9300] also has a mechanism where (S-EID,G) state can reside in the underlay through the use of RPF Vectors [RFC5496] in PIM Join/Prune messages. However, few PIM implementations support RPF Vectors, and LIISP should avoid S-EID state in the underlay. See Section 5 for details.

However, some observations can be made on the algorithm above. The control plane can scale but at the expense of sending data to sites that may have not joined the distribution tree where the encapsulated data is being delivered. For example, one site joins (S-EID1,G), and another site joins (S-EID2,G). Both EIDs are in the same multicast source site. Both multicast receiver sites join to the same ITR with state (S-RLOC,G) where S-RLOC is the RLOC for the ITR. The ITR joins both (S-EID1,G) and (S-EID2,G) inside of the site. The ITR receives (S-RLOC,G) joins and populates the OIF-list state for for the (S-RLOC,G) entry. Since both (S-EID1,G) and (S-EID2, G) map to the one (S-RLOC,G), packets will be delivered by the underlay to both multicast receiver sites even though each have joined a single source-based distribution tree. This behavior is a consequence of the many-to-one mapping between S-EIDs and a S-RLOC.

There is a possible solution to this problem that reduces the number of many-to-one occurrences of (S-EID,G) entries aggregating into a single (S-RLOC,G) entry. If a physical ITR can be assigned multiple RLOC addresses and these addresses are advertised in mapping database entries, then ETRs at receiver sites have more RLOC address options

and therefore can join different (RLOC,G) entries for each (S-EID,G) entry joined at the receiver site. It would not scale to have a one-to-one relationship between the number of S-EID sources at a source site and the number of RLOCs assigned to all ITRs at the site, but "n" can reduce to a smaller number in the "n-to-1" relationship. And in turn, this reduces the opportunity for data packets to be delivered to sites for groups not joined.

5. Source Addresses versus Group Addresses

Multicast group addresses don't have to be associated with either the EID or RLOC namespace. They actually are a namespace of their own that can be treated as logical with relatively opaque allocation. So, by their nature, they don't detract from an incremental deployment of LISP-Multicast.

As for source addresses, as in the unicast LISP scenario, there is a decoupling of identification from location. In a LISP site, packets are originated from hosts using their allocated EIDs. EID addresses are used to identify the host as well as where in the site's topology the host resides but not how and where it is attached to the Internet.

Therefore, when multicast distribution tree state is created anywhere in the network on the path from any multicast receiver to a multicast source, EID state is maintained at the source and receiver multicast sites, and RLOC state is maintained in the underlay. That is, a multicast distribution tree will be represented as a 3-tuple of {(S-EID,G) (S-RLOC,G) (S-EID,G)}, where the first element of the 3-tuple is the state stored in routers from the source to one or more ITRs in the source multicast site; the second element of the 3-tuple is the state stored in routers downstream of the ITR, in the underlay, to all LISP receiver multicast sites; and the third element in the 3-tuple is the state stored in the routers downstream of each ETR, in each receiver multicast site, reaching each receiver. Note that (S-EID,G) is the same in both the source and receiver multicast sites.

The concatenation/mapping from the first element to the second element of the 3-tuples is done by the ITR, and from the second element to the third element is done at the ETRs.

6. Locator Reachability Implications on LISP-Multicast

Multicast state as it is stored in the underlay is either (S,G) state for sources in non-LISP sites or (RLOC,G) state for sources in LISP sites. The underlay routers cannot distinguish one from the other. They don't need to because it is state that uses RPF against the underlay routing tables in the RLOC namespace. The difference is where the root of the distribution tree for a particular source is. In the traditional multicast underlay, the source S is the source host's IP address. For LISP-Multicast, the source S is a single ITR of the multicast source site.

An ITR is selected based on the LISP EID-to-RLOC mapping used when an ETR propagates a PIM Join/Prune message out of a receiver multicast site. The selection is based on the same algorithm an ITR would use to select an ETR when sending a unicast packet to the site. In the unicast case, the ITR can change on a per-packet basis depending on the reachability of the ETR. So, an ITR can change relatively easily using local reachability state. However, in the multicast case, when an ITR becomes unreachable, new distribution tree state must be built because the encapsulating root has changed. This is more significant than an RPF-change event, where any router would typically locally change its RPF-interface for its existing tree state. But when an encapsulating LISP-Multicast ITR goes unreachable, new distribution state must be built and reflect the new encapsulator. Therefore, when an ITR goes unreachable, all ETRs that are currently joined to that ITR will have to trigger a new Join/Prune message for (S-RLOC,G) to the new ITR as well as send a unicast encapsulated Join/Prune message telling the new ITR which (S-EID,G) is being joined.

This issue can be mitigated by using anycast addressing for the ITRs, so the problem does reduce to an RPF change in the underlay, but still requires a unicast encapsulated Join/Prune message to tell the new ITR about (S-EID,G). The problem with this approach is that the ETR really doesn't know when the ITR has changed, so the new anycast ITR will get the (S-EID,G) state only when the ETR sends it the next time during its periodic sending procedures.

7. Multicast Protocol Changes

A number of protocols are used today for inter-domain multicast routing:

IGMPv1-v3, MLDv1-v2: These protocols [RFC3376] [RFC3810] [RFC4604] do not require any changes for LISP-Multicast for two reasons. One is that they are link-local and not used over site boundaries, and the second is that they advertise group addresses that don't need translation. Where source addresses are supplied in IGMPv3

and Multicast Listener Discovery version 2 (MLDv2) messages, they are semantically regarded as EIDs and don't need to be converted to RLOCs until the multicast tree-building protocol, such as PIM, is received by the ETR at the site boundary. Addresses used for IGMP and MLD come out of the source site's allocated addresses, which are therefore from the EID namespace.

MBGP: Even though the Multiprotocol Extensions for BGP-4 (MBGP) [RFC4760] are not part of a multicast routing protocol, they are used to find multicast sources when the unicast BGP peering topology and the multicast MBGP peering topology are not congruent. When MBGP is used in a LISP-Multicast environment, the prefixes that are advertised are from the RLOC namespace. This allows receiver multicast sites to find a path to the source multicast site's ITRs. MBGP peering addresses will be from the RLOC namespace. There are no MBGP changes required to support LISP-Multicast.

MSDP: MSDP [RFC3618] is used to announce active multicast sources to other routing domains (or LISP sites). The announcements come from the PIM Rendezvous Points (RPs) from sites where there are active multicast sources sending to various groups. In the context of LISP-Multicast, the source addresses advertised in MSDP will semantically be from the EID namespace since they describe the identity of a source multicast host. It will be true that the state stored in MSDP caches from underlay routers will be from the EID namespace. An RP address inside of the site will be from the EID namespace so it can be advertised and reached by an internal unicast routing mechanism. However, for MSDP peer-RPF checking to work properly across sites, the RP addresses must be converted or mapped into a routable address that is advertised and maintained in the BGP routing tables in the underlay. MSDP peering addresses can come out of either the EID or a routable address namespace. Also, the choice can be made unilaterally because the ITR at the site will determine which namespace the destination peer address is out of by looking in the mapping database service. There are no MSDP changes required to support LISP-Multicast.

PIM-SSM: In the simplest form of distribution tree building, when PIM operates in SSM mode [RFC4607], a source distribution tree is built and maintained across site boundaries. In this case, there is a small modification to how PIM Join/Prune messages are sent by the LISP-Multicast component. No modifications to any message format, but to support taking a Join/Prune message originated inside of a LISP site with embedded addresses from the EID namespace and converting them to addresses from the RLOC namespace when the Join/Prune message crosses a site boundary. This is similar to the requirements documented in [RFC5135].

BIDIR-PIM: Bidirectional PIM [RFC5015] is typically run inside of a routing domain, but if deployed in an inter-domain environment, one would have to decide if the RP address of the shared tree would be from the EID namespace or the RLOC namespace. If the RP resides in a site-based router, then the RP address is from the EID namespace. If the RP resides in the underlay where RLOC addresses are routed, then the RP address is from the RLOC namespace. This could be easily distinguishable if the EID address were in a well-known address allocation block from the RLOC namespace. Also, when using Embedded-RP for RP determination [RFC3956], the format of the group address could indicate the namespace the RP address is from. However, refer to Section 10 for considerations underlay routers need to make when using Embedded-RP IPv6 group addresses. When using BIDIR-PIM for inter-domain multicast routing, it is recommended to use statically configured RPs. This allows underlay routers to associate a Bidir group's RP address with an ITR's RLOC address, and site routers to associate the Bidir group's RP address as an EID address. With respect to Designated Forwarder (DF) election in BIDIR-PIM, no changes are required since all messaging and addressing is link-local.

PIM-ASM: The ASM mode of PIM [RFC7761], the most popular form of PIM, is deployed in the Internet today by having shared trees within a site and using source trees across sites. By the use of MSDP and PIM-SSM techniques described above, multicast connectivity can occur across LIISP sites. Having said that, that means there are no special actions required for processing (*,G) or (S,G,R) Join/Prune messages since they all operate against the shared tree that is site resident. Just like with ASM, there is no (*,G) in the underlay when LIISP-Multicast is in use. This is also true for the RP-mapping mechanisms Auto-RP and Bootstrap Router (BSR) [RFC5059].

Based on the protocol description above, the conclusion is that there are no protocol message format changes, just a translation function performed at the control plane. This will make for an easier and faster transition for LIISP since fewer components in the network have to change.

It should also be stated just like it is in [RFC9300] that no host changes, whatsoever, are required to have a multicast source host send multicast packets and for a multicast receiver host to receive multicast packets.

7.1. PIM Join Attributes for LISP

There are specific PIM Join/Prune attributes to build multicast distribution trees with either a native Multicast or Unicast underlay. Please refer to [RFC8059] for such attributes.

These attributes are further enhanced in [I-D.ietf-pim-jp-extensions-lisp]

8. LISP-Multicast Data-Plane Architecture

The LISP-Multicast data-plane operation conforms to the operation and packet formats specified in [RFC9300]. However, encapsulating a multicast packet from an ITR is a much simpler process. The process is simply to copy the inner group address to the outer destination address. And to have the ITR use its own IP address (its RLOC) as the source address. The process is simpler for multicast because there is no EID-to-RLOC mapping lookup performed during packet forwarding.

In the decapsulation case, the ETR simply removes the outer header and performs a multicast routing table lookup on the inner header (S-EID,G) addresses. Then, the OIF-list for the (S-EID,G) entry is used to replicate the packet on site-facing interfaces leading to multicast receiver hosts.

There is no Data-Probe logic for ETRs as there can be in the unicast forwarding case.

8.1. ITR Forwarding Procedure

The following procedure is used by an ITR, when it receives a multicast packet from a source inside of its site:

1. A multicast data packet sent by a host in a LISP site will have the source address equal to the host's EID and the destination address equal to the address of the multicast group. It is assumed the group information is obtained by current methods. The same is true for a multicast receiver to obtain the source and group address of a multicast flow.
2. When the ITR receives a multicast packet, it will have both S-EID state and S-RLOC state stored. Since the packet was received on a site-facing interface, the RPF lookup is based on the S-EID state. If the RPF check succeeds, then the OIF-list contains interfaces that are site facing and external facing. For the site-facing interfaces, no LISP header is prepended. For the external-facing interfaces a LISP header is prepended. When the

ITR prepends a LISP header, it uses its own RLOC address as the source address and copies the group address supplied by the IP header that the host built as the outer destination address.

8.1.1. Multiple RLOCs for an ITR

Typically, an ITR will have a single RLOC address, but in some cases there could be multiple RLOC addresses assigned from either the same or different service providers. In this case, when (S-RLOC,G) Join/Prune messages are received for each RLOC, there is a OIF-list merging action that must take place. Therefore, when a packet is received from a site-facing interface that matches on an (S-EID,G) entry, the interfaces of the OIF-list from all (RLOC,G) entries joined to the ITR as well as the site-facing OIF-list joined for (S-EID,G) must be included in packet replication. In addition to replicating for all types of OIF-lists, each OIF-list entry must be tagged with the RLOC address, so encapsulation uses the outer source address for the RLOC joined.

8.1.2. Multiple ITRs for a LISP Source Site

Note that when ETRs from different multicast receiver sites receive (S-EID,G) joins, they may select a different S-RLOC for a multicast source site due to policy (the multicast ITR can return different multicast priority and weight values per ETR Map-Request). In this case, the same (S-EID,G) is being realized by different (S-RLOC,G) state in the underlay. This will not result in duplicate packets because each ITR in the multicast source site will choose their own RLOC for the source address for encapsulated multicast traffic. The RLOC addresses are the ones joined by remote multicast ETRs.

When different (S-EID,G) traffic is combined into a single (RLOC,G) underlay distribution tree, this may cause traffic to go to a receiver multicast site when it does not need to. This happens when one receiver multicast site joins (S1-EID,Gi) through a underlay distribution tree of (RLOC1,Gi) and another multicast receiver site joins (S2-EID,Gi) through the same underlay distribution tree of (RLOC1,Gi). When ETRs decapsulate such traffic, they should know from their local (S-EID,G) state if the packet should be forwarded. If there is no (S-EID,G) state that matches the inner packet header, the packet is discarded.

8.2. ETR Forwarding Procedure

The following procedure is used by an ETR, when it receives a multicast packet from a source outside of its site:

1. When a multicast data packet is received by an ETR on an external-facing interface, it will do an RPF lookup on the S-RLOC state it has stored. If the RPF check succeeds, the interfaces from the OIF-list are used for replication to interfaces that are site facing as well as interfaces that are external facing (this ETR can also be a transit multicast router for receivers outside of its site). When the packet is to be replicated for an external-facing interface, the LISP encapsulation header is not stripped. When the packet is replicated for a site-facing interface, the encapsulation header is stripped.
2. The packet without a LISP header is now forwarded down the (S-EID,G) distribution tree in the receiver multicast site.

8.3. Replication Locations

Multicast packet replication can happen in the following topological locations:

- * In an IGP multicast router inside a site that operates on S-EIDs.
- * In a transit multicast router inside of the underlay that operates on S-RLOCs.
- * At one or more ETR routers depending on the path a Join/Prune message exits a receiver multicast site.
- * At one or more ITR routers in a source multicast site depending on what priorities are returned in a Map-Reply to receiver multicast sites.

In the last case, the source multicast site can do replication rather than having a single exit from the site. But this can occur only when the priorities in the Map-Reply are modified for different receiver multicast sites so that the PIM Join/Prune messages arrive at different ITRs.

This policy technique, also used in [RFC6836] for unicast, is useful for multicast to mitigate the problems of changing distribution tree state as discussed in Section 6.

9. LISP-Multicast Interworking

This section describes the multicast corollary to [RFC6832] regarding the interworking of multicast routing among LISP and non-LISP sites.

9.1. LISP and Non-LISP Mixed Sites

Since multicast communication can involve more than two entities to communicate together, the combinations of interworking scenarios are more involved. However, the state maintained for distribution trees at the sites is the same, regardless of whether or not the site is LISP enabled. So, most of the implications are in the underlay with respect to storing routable EID-Prefixes from either PA or PI blocks.

Before enumerating the multicast interworking scenarios, let's define three deployment states of a site:

- * A non-LISP site that will run PIM-SSM or PIM-ASM with MSDP as it does today. The addresses for the site are globally routable.
- * A site that deploys LISP for unicast routing. The addresses for the site are not globally routable. Let's define the name for this type of site as a uLISP site.
- * A site that deploys LISP for both unicast and multicast routing. The addresses for the site are not globally routable. Let's define the name for this type of site as a LISP-Multicast site.

A LISP site enabled for multicast purposes only will not be considered in this document, but a uLISP site as documented in [RFC6832] will be considered. In this section there is no discussion of how a LISP site sends multicast packets when all receiver sites are LISP-Multicast enabled; that has been discussed in previous sections.

The following scenarios exist to make LISP-Multicast sites interwork with non-LISP-Multicast sites:

1. A LISP site must be able to send multicast packets to receiver sites that are a mix of non-LISP sites and uLISP sites.
2. A non-LISP site must be able to send multicast packets to receiver sites that are a mix of non-LISP sites and uLISP sites.
3. A non-LISP site must be able to send multicast packets to receiver sites that are a mix of LISP sites, uLISP sites, and non-LISP sites.
4. A uLISP site must be able to send multicast packets to receiver sites that are a mix of LISP sites, uLISP sites, and non-LISP sites.

5. A LISP site must be able to send multicast packets to receiver sites which are a mix of LISP sites, uLISP sites, and non-LISP sites.

9.1.1. LISP Source Site to Non-LISP Receiver Sites

In the first scenario, a site is LISP enabled for both unicast and multicast traffic and as such operates on EIDs. Therefore, there is a possibility that the EID-Prefix block is not routable in the underlay. For LISP receiver multicast sites, this isn't a problem, but for non-LISP or uLISP receiver multicast sites, when a PIM Join/Prune message is received by the edge router, it has no route to propagate the Join/Prune message out of the site. This is no different than the unicast case that LISP Network Address Translation (LISP-NAT) in [RFC6832] solves.

LISP-NAT allows a unicast packet that exits a LISP site to get its source address mapped to a globally routable address before the ITR realizes that it should not encapsulate the packet destined to a non-LISP site. For a multicast packet to leave a LISP site, distribution tree state needs to be built so the ITR can know where to send the packet. So, the receiver multicast sites need to know about the multicast source host by its routable address and not its EID address. When this is the case, the routable address is the (S-RLOC,G) state that is stored and maintained in the underlay routers. It is important to note that the routable address for the host cannot be the same as an RLOC for the site because it is desirable for ITRs to process a PIM Join/Prune message that is received from an external-facing interface. If the message will be propagated inside of the site, the site-part of the distribution tree is built.

Using a globally routable source address allows non-LISP and uLISP multicast receivers to join, create, and maintain a multicast distribution tree. However, the LISP-Multicast receiver site will want to perform an EID-to-RLOC mapping table lookup when a PIM Join/Prune message is received on a site-facing interface. It does this because it wants to find an (S-RLOC,G) entry to Join in the underlay. So, there is a conflict of behavior between the two types of sites.

The solution to this problem is the same as when an ITR wants to send a unicast packet to a destination site but needs to determine if the site is LISP enabled or not. When it is not LISP enabled, the ITR does not encapsulate the packet. So, for the multicast case, when the ETR receives a PIM Join/Prune message for (S-EID,G) state, it will do a mapping table lookup on S-EID. In this case, S-EID is not in the mapping database because the source multicast site is using a routable address and not an EID-Prefix address. So, the ETR knows to

simply propagate the PIM Join/Prune message to an external-facing interface without converting the (S-EID,G) because it is an (S,G), where S is routable and reachable via underlay routing tables.

Now that the multicast distribution tree is built and maintained from any non-LISP or uLISP receiver multicast site, the way the packet forwarding model is used can be explained.

Since the ITR in the source multicast site has never received a unicast encapsulated PIM Join/Prune message from any ETR in a receiver multicast site, it knows there are no LISP-Multicast receiver sites. Therefore, there is no need for the ITR to encapsulate data. Since it will know a priori (via configuration) that its site's EIDs are not routable (and not registered to the mapping database system), it assumes that the multicast packets from the source host are sent by a routable address. That is, it is the responsibility of the multicast source host's system administrator to ensure that the source host sends multicast traffic using a routable source address. When this happens, the ITR acts simply as a router and forwards the multicast packet like an ordinary multicast router.

There is an alternative to using a LISP-NAT scheme just as there is an alternative to using unicast [RFC6832] forwarding by employing Proxy Tunnel Routers (PxTRs). This can work the same way for multicast routing as well, but the difference is that non-LISP and uLISP sites will send PIM Join/Prune messages for (S-EID,G) that make their way in the underlay to multicast PxTRs. Let's call this use of a PxTR as a "Multicast Proxy-ETR" (or mPETR). Since the mPETRs advertise very coarse EID-Prefixes, they draw the PIM Join/Prune control traffic making them the target of the distribution tree. To get multicast packets from the LISP source multicast sites, the tree needs to be built on the path from the mPETR to the LISP source multicast site. To make this happen, the mPETR acts as a "Proxy-ETR" (where in unicast it acts as a "Proxy-ITR", or an uPITR [RFC6832]).

The existence of mPETRs in the underlay allows source multicast site ITRs to encapsulate multicast packets according to (S-RLOC,G) state. The (S-RLOC,G) state is built from the mPETRs to the multicast ITRs. The encapsulated multicast packets are decapsulated by mPETRs and then forwarded according to (S-EID,G) state. The (S-EID,G) state is built from the non-LISP and uLISP receiver multicast sites to the mPETRs.

9.1.2. Non-LISP Source Site to Non-LISP Receiver Sites

Clearly non-LISP-Multicast sites can send multicast packets to non-LISP receiver multicast sites. That is what they do today. However, discussion is required to show how non-LISP-Multicast sites send multicast packets to uLISP receiver multicast sites.

Since uLISP receiver multicast sites are not targets of any (S,G) state, they simply send (S,G) PIM Join/Prune messages toward the non-LISP source multicast site. Since the source multicast site in this case has not been upgraded to LISP, all multicast source host addresses are routable. So, this case is simplified to where a uLISP receiver multicast site appears to the source multicast site to be a non-LISP receiver multicast site.

9.1.3. Non-LISP Source Site to Any Receiver Site

When a non-LISP source multicast site has receivers in either a non-LISP/uLISP site or a LISP site, one needs to decide how the LISP receiver multicast site will attach to the distribution tree. It is known from Section 9.1.2 that non-LISP and uLISP receiver multicast sites can join the distribution tree, but a LISP receiver multicast site ETR will need to know if the source address of the multicast source host is routable or not. It has been shown in Section 9.1.1 that an ETR, before it sends a PIM Join/Prune message on an external-facing interface, does an EID-to-RLOC mapping lookup to determine if it should convert the (S,G) state from a PIM Join/Prune message received on a site-facing interface to an (S-RLOC,G). If the lookup fails, or it receives a Negative Map-Reply, the ETR can conclude the source multicast site is a non-LISP site, so it simply forwards the Join/Prune message. (It also doesn't need to send a unicast encapsulated Join/Prune message because there is no ITR in a non-LISP site and there is namespace continuity between the ETR and source.)

For a non-LISP source multicast site, (S-EID,G) state could be limited to the edges of the network with the use of multicast proxy-ITRs (mPITRs). The mPITRs can take native, unencapsulated multicast packets from non-LISP source multicast and uLISP sites and encapsulate them to ETRs in receiver multicast sites or to mPETRs that can decapsulate for non-LISP receiver multicast or uLISP sites. The mPITRs are responsible for sending (S-EID,G) joins to the non-LISP source multicast site. To connect the distribution trees together, multicast ETRs will need to be configured with the mPITR's RLOC addresses so they can send both (S-RLOC,G) joins to build a distribution tree to the mPITR as well as configured for sending unicast joins to mPITRs so they can propagate (S-EID,G) joins into source multicast sites. The use of mPITRs is undergoing more study and is a work in progress.

9.1.4. Unicast LISP Source Site to Any Receiver Sites

In the last section, it was explained how an ETR in a multicast receiver site can determine if a source multicast site is LISP enabled by looking into the mapping database. When the source multicast site is a uLISP site, it is LISP enabled, but the ITR, by definition, is not capable of doing multicast encapsulation. So, for the purposes of multicast routing, the uLISP source multicast site is treated as a non-LISP source multicast site.

Non-LISP receiver multicast sites can join distribution trees to a uLISP source multicast site since the source site behaves, from a forwarding perspective, as a non-LISP source site. This is also the case for a uLISP receiver multicast site since the ETR does not have multicast functionality built-in or enabled.

Special considerations are required for LISP receiver multicast sites; since they think the source multicast site is LISP enabled, the ETR cannot know if the ITR is LISP-Multicast enabled. To solve this problem, each mapping database entry will have a multicast 2-tuple (Mpriority, Mweight) per RLOC [RFC9300]. When the Mpriority is set to 255, the site is considered not multicast capable. So, an ETR in a LISP receiver multicast site can distinguish whether a LISP source multicast site is a LISP-Multicast site or a uLISP site.

9.1.5. LISP Source Site to Any Receiver Sites

When a LISP source multicast site has receivers in LISP, non-LISP, and uLISP receiver multicast sites, it has a conflict about how it sends multicast packets. The ITR can either encapsulate or natively forward multicast packets. Since the receiver multicast sites are heterogeneous in their behavior, one packet-forwarding mechanism cannot satisfy both. However, if a LISP receiver multicast site acts like a uLISP site, then it could receive packets like a non-LISP receiver multicast site, thereby making all receiver multicast sites have homogeneous behavior. However, this poses the following issues:

- * LISP-NAT techniques with routable addresses would be required in all cases.
- * Or, alternatively, mPETR deployment would be required, thus forcing coarse EID-Prefix advertisement in the underlay.

- * But, what is most disturbing is that when all sites that participate are LISP-Multicast sites but a non-LISP or uLISP site joins the distribution tree, then the existing joined LISP receiver multicast sites would have to change their behavior. This would create too much dynamic tree-building churn to be a viable alternative.

So, the solution space options are:

1. Make the LISP ITR in the source multicast site send two packets, one that is encapsulated with (S-RLOC,G) to reach LISP receiver multicast sites and another that is not encapsulated with (S-EID,G) to reach non-LISP and uLISP receiver multicast sites.
2. Make the LISP ITR always encapsulate packets with (S-RLOC,G) to reach LISP-Multicast sites and to reach mPETRs that can decapsulate and forward (S-EID,G) packets to non-LISP and uLISP receiver multicast sites.

9.2. LISP Sites with Mixed Address Families

A LISP database mapping entry that describes the Locator-Set, Mpriority, and Mweight per locator address (RLOC), for an EID-Prefix associated with a site could have RLOC addresses in either IPv4 or IPv6 format. When a mapping entry has a mix of RLOC-formatted addresses, it is an implicit advertisement by the site that it is a dual-stack site. That is, the site can receive IPv4 or IPv6 unicast packets.

To distinguish if the site can receive dual-stack unicast packets as well as dual-stack multicast packets, the Mpriority value setting will be relative to an IPv4 or IPv6 RLOC See [RFC9300] for packet format details.

If one considers the combinations of LISP, non-LISP, and uLISP sites sharing the same distribution tree and considering the capabilities of supporting IPv4, IPv6, or dual-stack, the number of total combinations grows beyond comprehension.

Some of the combinations that can occur are listed below:

1. LISP-Multicast IPv4 Site
2. LISP-Multicast IPv6 Site
3. LISP-Multicast Dual-Stack Site
4. uLISP IPv4 Site

5. uLISP IPv6 Site
6. uLISP Dual-Stack Site
7. non-LISP IPv4 Site
8. non-LISP IPv6 Site
9. non-LISP Dual-Stack Site

The above combinatorial gets even worse when one considers a site using one address family inside of the site and the xTRs using the other address family (as in using IPv4 EIDs with IPv6 RLOCs or IPv6 EIDs with IPv4 RLOCs).

To rationalize this combinatorial nightmare, there are some guidelines that need to be put in place:

- * Each distribution tree shared between sites will either be an IPv4 distribution tree or an IPv6 distribution tree. Therefore, head-end replication can be avoided by building and sending packets on each address-family-based distribution tree. Even though there might be an urge to do multicast packet translation from one address family format to the other, it is a non-viable over-complicated urge. Multicast ITRs will only encapsulate packets where the inner and outer headers are from the same address family.
- * All LISP sites on a multicast distribution tree must share a common address family that is determined by the source site's Locator-Set in its LISP database mapping entry. All receiver multicast sites will use the best RLOC priority controlled by the source multicast site. This is true when the source site is either LISP-Multicast or uLISP enabled. This means that priority-based policy modification is prohibited. When a receiver multicast site ETR receives an (S-EID,G) join, it must select a S-RLOC for the same address family as S-EID.
- * When a multicast Locator-Set has more than one locator, only locators from the same address family MUST be set to the same best priority value. A mixed Locator-Set can exist (for unicast use), but the multicast priorities MUST be the set for the same address family locators.
- * When the source site is not LISP enabled, determining the address family for the flow is up to how receivers find the source and group information for a multicast flow.

9.3. Making a Multicast Interworking Decision

Thus far, Section 9 has shown all combinations of multicast connectivity that could occur. As already concluded, this can be quite complicated, and, if the design is too ambitious, the dynamics of the protocol could cause a lot of instability.

The trade-off decisions are hard to make, and so the same single solution is desirable to work for both IPv4 and IPv6 multicast. It is imperative to have an incrementally deployable solution for all of IPv4 unicast and multicast and IPv6 unicast and multicast while minimizing (or eliminating) both unicast and multicast EID namespace state.

Therefore, the design decision to go with uPITRs [RFC6832] for unicast routing and mPETRs for multicast routing seems to be the sweet spot in the solution space in order to optimize state requirements and avoid head-end data replication at ITRs.

10. Considerations When RP Addresses Are Embedded in Group Addresses

When ASM and PIM-BIDIR are used in an IPv6 inter-domain environment, a technique exists to embed the unicast address of an RP in an IPv6 group address [RFC3956]. When routers in end sites process a PIM Join/Prune message that contains an Embedded-RP group address, they extract the RP address from the group address and treat it from the EID namespace. However, underlay routers do not have state for the EID namespace and need to extract an RP address from the RLOC namespace.

Therefore, it is the responsibility of ETRs in multicast receiver sites to map the group address into a group address where the Embedded-RP address is from the RLOC namespace. The mapped RP address is obtained from an EID-to-RLOC mapping database lookup. The ETR will also send a unicast (*,G) Join/Prune message to the ITR so the branch of the distribution tree from the source site resident RP to the ITR is created.

This technique is no different than the techniques described in this specification for translating (S,G) state and propagating Join/Prune messages into the underlay. The only difference is that the (*,G) state in Join/Prune messages are mapped because they contain unicast addresses encoded in an Embedded-RP group address.

11. Taking Advantage of Upgrades in the Underlay

If the underlay routers are upgraded to support [RFC5496], then the EID-specific data can be passed through the underlay without, possibly, having to store the state in the underlay.

By doing this, one can eliminate the ETR from unicast encapsulated PIM Join/Prune messages to the source site's ITR.

However, this solution is restricted to a small set of workable cases that would not be good for general use of LISP-Multicast. In addition, due to slow convergence properties, it is not recommended for LISP-Multicast.

12. Mtrace Considerations

Mtrace functionality MUST be consistent with unicast traceroute functionality where all hops from multicast receiver to multicast source are visible.

The design for mtrace for use in LISP-Multicast environments is to be determined but should build upon mtrace version 2 specified in [MTRACE] [RFC8487].

13. Security Considerations

The security concerns for LISP-Multicast are mainly the same as for the base LISP specification [RFC9300] and for multicast in general, including PIM-ASM [RFC7761].

There may be a security concern with respect to unicast PIM messages. When multiple receiver sites are joining an (S-EID1,G) distribution tree that maps to a (RLOC1,G) underlay distribution tree, and a malicious receiver site joins an (S-EID2,G) distribution tree that also maps to the (RLOC1,G) underlay distribution tree, the legitimate sites will receive data from S-EID2 when they did not ask for it.

Deployments have been successfully able to leverage existing security mechanisms deployed for PIM [RFC7761]

14. IANA Considerations

No requests are made to IANA

15. Acknowledgments

The authors would like to gratefully acknowledge the people who have contributed discussion, ideas, and commentary to the making of this proposal and specification. People who provided expert review were Scott Brim, Greg Shepherd, and Dave Oran. Other commentary from discussions at the Summer 2008 IETF in Dublin were Toerless Eckert and IJsbrand Wijnands.

The authors would also like to thank the MBONED working group for constructive and civil verbal feedback when this document was presented at the Fall 2008 IETF in Minneapolis. In particular, good commentary came from Tom Pusateri, Steve Casner, Marshall Eubanks, Dimitri Papadimitriou, Ron Bonica, Lenny Guardino, Alia Atlas, Jesus Arango, and Jari Arkko.

An expert review of this specification was done by Yiqun Cai and Liming Wei. The authors thank them for their detailed comments.

This work originated in the Routing Research Group (RRG) of the IRTF. An individual submission was converted into a LISP working group document.

A very special thanks goes to Mike McBride and Stig Venaas for taking over as joint editors of this document in the later stages of progressing to proposed standard.

The authors also thank Luigi Iannone for his extensive comments to improve the readability of the document.

16. References

16.1. 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>>.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <<https://www.rfc-editor.org/info/rfc3376>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.

- [RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, DOI 10.17487/RFC3956, November 2004, <<https://www.rfc-editor.org/info/rfc3956>>.
- [RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, DOI 10.17487/RFC4604, August 2006, <<https://www.rfc-editor.org/info/rfc4604>>.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <<https://www.rfc-editor.org/info/rfc4607>>.
- [RFC4610] Farinacci, D. and Y. Cai, "Anycast-RP Using Protocol Independent Multicast (PIM)", RFC 4610, DOI 10.17487/RFC4610, August 2006, <<https://www.rfc-editor.org/info/rfc4610>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, DOI 10.17487/RFC5015, October 2007, <<https://www.rfc-editor.org/info/rfc5015>>.
- [RFC5135] Wing, D. and T. Eckert, "IP Multicast Requirements for a Network Address Translator (NAT) and a Network Address Port Translator (NAPT)", BCP 135, RFC 5135, DOI 10.17487/RFC5135, February 2008, <<https://www.rfc-editor.org/info/rfc5135>>.
- [RFC5496] Wijnands, IJ., Boers, A., and E. Rosen, "The Reverse Path Forwarding (RPF) Vector TLV", RFC 5496, DOI 10.17487/RFC5496, March 2009, <<https://www.rfc-editor.org/info/rfc5496>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.

- [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>>.
- [RFC8487] Asaeda, H., Meyer, K., and W. Lee, Ed., "Mtrace Version 2: Traceroute Facility for IP Multicast", RFC 8487, DOI 10.17487/RFC8487, October 2018, <<https://www.rfc-editor.org/info/rfc8487>>.
- [RFC9300] Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, Ed., "The Locator/ID Separation Protocol (LISP)", RFC 9300, DOI 10.17487/RFC9300, October 2022, <<https://www.rfc-editor.org/info/rfc9300>>.

16.2. Informative References

- [I-D.ietf-pim-jp-extensions-lisp] Govindan, V. P. and S. Venaas, "PIM Join/Prune Attributes for LISP Environments using Underlay Multicast", Work in Progress, Internet-Draft, draft-ietf-pim-jp-extensions-lisp-09, 21 February 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-pim-jp-extensions-lisp-09>>.
- [MTRACE] Asaeda, H. and W. Lee, Ed., "Mtrace Version 2: Traceroute Facility for IP Multicast", Work in Progress, October 2012.
- [RFC3618] Fenner, B., Ed. and D. Meyer, Ed., "Multicast Source Discovery Protocol (MSDP)", RFC 3618, DOI 10.17487/RFC3618, October 2003, <<https://www.rfc-editor.org/info/rfc3618>>.
- [RFC5059] Bhaskar, N., Gall, A., Lingard, J., and S. Venaas, "Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)", RFC 5059, DOI 10.17487/RFC5059, January 2008, <<https://www.rfc-editor.org/info/rfc5059>>.
- [RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, DOI 10.17487/RFC6831, January 2013, <<https://www.rfc-editor.org/info/rfc6831>>.
- [RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, DOI 10.17487/RFC6832, January 2013, <<https://www.rfc-editor.org/info/rfc6832>>.

- [RFC6836] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Topology (LIISP+ALT)", RFC 6836, January 2013, <<https://www.rfc-editor.org/rfc/rfc6836>>.
- [RFC8059] Arango, J., Venaas, S., Kouvelas, I., and D. Farinacci, "PIM Join Attributes for Locator/ID Separation Protocol (LIISP) Environments", RFC 8059, DOI 10.17487/RFC8059, January 2017, <<https://www.rfc-editor.org/info/rfc8059>>.

Appendix A. Document Change Log

[RFC Editor: Please delete this section on publication as RFC.]

A.1. Changes to draft-ietf-lisp-rfc6831bis-03

- * Posted Nov 2025.
- * Address one missing comment from Dino

A.2. Changes to draft-ietf-lisp-rfc6831bis-03

- * Posted Nov 2025.
- * Update document for comments from Luigi as prep for wGLC.

A.3. Changes to draft-ietf-lisp-rfc6831bis-02

- * Posted July 2025.
- * Update document timer and references.
- * Update Dave Meyer's email address.

A.4. Changes to draft-ietf-lisp-rfc6831bis-01

- * Posted January 2025 (Mike McBride).
- * Changed all instances of "core" to "underlay".
- * Changed "we see" to "there are".
- * Changed "RFC" to "document".
- * Updated the Requirements Notation.
- * Added a reference to RFC 7761 for the RPF definition.

- * Removed part of a sentence "...the exit is chosen according to routing policies..."
- * Rephrased sentence to: "Multicast state as it is stored in the core is either (S,G) state for sources in non-LISP sites or (RLOC,G) state for sources in LISP sites."
- * Added "...or it receives a Negative Map-Reply".
- * Added a reference to RFC8487 for MTRACE.
- * Moved a few downrefs that needed to be moved to the Informative section.
- * Added Mike and Stig to acknowledgements

A.5. Changes to draft-ietf-lisp-rfc6831bis-00

- * Posted August 2024.
- * Make draft-farinacci-lisp-rfc6831bis-02 working group document draft-ietf-lisp-rfc6831bis-00.

A.6. Changes to draft-farinacci-lisp-rfc6831bis-02

- * Posted May 2024 (Vengada Prasad Govindan).
- * Added reference to RFC8059

A.7. Changes to draft-farinacci-lisp-rfc6831bis-01

- * Posted May 2024 (Vengada Prasad Govindan).
- * Add support for unicast underlay in signal-based approach (RFC6831)

A.8. Changes to draft-farinacci-lisp-rfc6831bis-00

- * Posted May 2024.
- * Starting with [RFC6831] to move it to a bis document for standards track.
- * Changed references to standards track RFCs.
- * Address comments from Prasad Govindan.

Authors' Addresses

Dino Farinacci
lispers.net
Email: farinacci@gmail.com

Dave Meyer
Cisco Systems
Tasman Drive
San Jose, CA
United States of America
Email: dmm613@gmail.com

John Zwiebel
Cisco Systems
Tasman Drive
San Jose, CA
United States of America
Email: jzwiebel@cruzio.com

Stig Venaas
Cisco Systems
Tasman Drive
San Jose, CA
United States of America
Email: stig@cisco.com

Vengada Prasad Govindan
Cisco Systems
Email: venggovi@cisco.com

Network File System Version 4
Internet-Draft
Intended status: Standards Track
Expires: 14 March 2027

T. Haynes
Hammerspace
10 September 2026

Adding an Uncacheable File Data Attribute to NFSv4.2
draft-ietf-nfsv4-uncacheable-files-13

Abstract

Network File System version 4.2 (NFSv4.2) clients commonly perform client-side caching of file data in order to improve performance. On some systems, applications may influence client data caching behavior, but there is no standardized mechanism for a server or administrator to indicate that particular file data should not be cached by clients for reasons of performance or correctness. This document introduces a new file data caching attribute for NFSv4.2. Files marked with this attribute are intended to be accessed with client-side caching of file data suppressed, in order to support workloads that require predictable data visibility. This document extends NFSv4.2.

Note to Readers

Discussion of this draft takes place on the NFSv4 working group mailing list (nfsv4@ietf.org), which is archived at https://mailarchive.ietf.org/arch/search/?email_list=nfsv4. Source code and issues list for this draft can be found at <https://github.com/ietf-wg-nfsv4/uncacheable-files>.

Working Group information can be found at <https://github.com/ietf-wg-nfsv4>.

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 14 March 2027.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Definitions	3
3. Requirements Language	4
4. The Uncacheable File Data Attribute	4
5. Client-Side Caching of File Data	5
5.1. Write-Behind Caching	6
5.2. WRITE Durability	6
5.3. Read Caching	7
5.4. Relationship to Direct I/O	7
6. Setting the Uncacheable File Data Attribute	7
7. Implementation Status	8
8. XDR for Uncacheable Attribute	9
9. Extraction of XDR	9
10. Security Considerations	10
11. IANA Considerations	10
12. References	10
12.1. Normative References	10
12.2. Informative References	11
Acknowledgments	11
Author's Address	12

1. Introduction

Clients of remote filesystems commonly perform client-side caching of file data in order to improve performance. Such caching may include retaining data read from the server to satisfy subsequent READ requests, as well as retaining data written by applications in order to delay or combine WRITE requests before transmitting them to the server. While these techniques are effective for many workloads, they may be unsuitable for workloads that require predictable data

visibility or involve concurrent modification of shared files by multiple clients.

In some cases, Network File System version 4.2 (NFSv4.2) (see [RFC7862]) mechanisms such as file delegations can reduce the impact of concurrent access. However, delegations are not always available or effective, particularly for workloads with frequent concurrent writers or rapidly changing access patterns.

There have been prior efforts to bypass file data caching in order to address these issues. In High-Performance Computing (HPC) workloads, file data caching is often bypassed to improve predictability and to avoid read-modify-write hazards when multiple clients write disjoint byte ranges of the same file.

Applications on some systems can request bypass of the client data cache by opening files with the `O_DIRECT` flag (see [OPEN-O_DIRECT]). However, this approach has limitations, including the requirement that each application be explicitly modified and the lack of a standardized mechanism for communicating this intent between servers and clients.

This document introduces the uncacheable file data attribute to NFSv4.2. This attribute allows a server to indicate that client-side caching of file data for a particular file is unsuitable. When both the client and the server support this attribute, the client is advised to suppress client-side caching of file data for that file, in accordance with the semantics defined in this document. The attribute itself is specified in Section 4 and the client behavior it advises in Section 5.

Using the process described in [RFC8178] Section 6, this document extends NFSv4.2 [RFC7862]; as that section provides, it does not update [RFC7862], which remains a valid description of the base variant of the minor version. The revisions are built on top of the external data representation (XDR) [RFC4506] generated from [RFC7863].

2. Definitions

client-side caching of file data The retention of file data by a client in a local data cache, commonly referred to as the page cache, for the purpose of satisfying subsequent READ requests or delaying transmission of WRITE data to the server.

write-behind caching A form of file data caching in which WRITE data

is retained by the client and transmission of the data to the server is delayed in order to combine multiple WRITE operations or improve efficiency.

direct I/O An access mode in which file data is transferred between application buffers and the underlying storage without populating or consulting the client's file data cache. Direct I/O suppresses both read caching and write-behind caching of file data.

write hole A write hole is an instance of data corruption that arises when multiple clients modify disjoint byte ranges within the same encoded data block without having a consistent view of the existing contents. This can result in stale data overwriting newer updates, particularly in environments that use erasure encoding or striped storage.

revalidation The procedure of [RFC8881] Section 10.3.1 by which a client determines whether data or attributes it holds in a cache are still current: the client fetches the change attribute from the server, compares it with the value it cached, and, if they differ, treats the cached data and attributes as invalid. A client validates data when it fetches it from the server; it revalidates before reusing what it cached.

This document assumes familiarity with the NFSv4 protocol operations, error codes, object types, and attributes as defined in [RFC8881].

3. Requirements Language

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] when, and only when, they appear in all capitals, as shown here.

4. The Uncacheable File Data Attribute

The uncacheable file data attribute, `fattr4_uncacheable_file_data` (see Section 8), is a RECOMMENDED attribute for NFSv4.2, in the attribute-category sense of [RFC8881] Section 5.2 and [RFC7862] Section 12 rather than the BCP 14 sense; a server is not required to support it. It is read-write, applies on a per-file basis, and has a data type of boolean. Table 1 summarizes the attribute using the columns of [RFC7862] Section 12.1, where "R W" indicates that GETATTR may retrieve the attribute and SETATTR may set it.

Name	Id	Data Type	Acc
uncacheable_file_data	87	bool	R W

Table 1: New RECOMMENDED Attribute

Support for the uncacheable file data attribute is specific to the exported filesystem and may differ between filesystems served by the same server. A client can determine whether the attribute is supported for a given file by examining the `supported_attrs` attribute for that file's filesystem or by probing support using the procedures described in [RFC8178].

The uncacheable file data attribute applies only to regular files, that is, objects of type `NF4REG` or `NF4NAMEDATTR` ([RFC7862] Section 6.1). A named attribute is read and written with the same `READ` and `WRITE` operations as a regular file ([RFC8881] Section 5.3), so a client caches its data by the same mechanism and is exposed to the same hazards. A server that receives a `GETATTR` requesting this attribute for an object that is not a regular file **MUST** return `FALSE`: support for an attribute is advertised per file system ([RFC8881] Section 5.8.1.1), so a server that supports this attribute supports it for every object in that file system and owes a value for each ([RFC8881] Section 18.7.3). As with `rawdev` ([RFC8881] Section 5.8.2.31), the value **SHOULD NOT** be considered useful for such an object. A server that receives a `SETATTR` requesting this attribute on an object that is not a regular file **MUST** return `NFS4ERR_WRONG_TYPE` ([RFC8881] Section 15.1.2.9).

5. Client-Side Caching of File Data

The uncacheable file data attribute advises the client to limit the use of client-side caching of file data for a file. This includes both write-behind caching and read caching, which are addressed separately below.

The server is often in a better position than individual clients to determine sharing patterns, access behavior, or correctness requirements associated with a file. By exposing this information via an attribute, the server can advise clients to limit file data caching in a consistent manner.

5.1. Write-Behind Caching

The uncacheable file data attribute inhibits write-behind caching, in which multiple pending WRITES are combined and transmitted to the server at a later time for efficiency.

When honoring the uncacheable file data attribute, clients MUST NOT delay transmission of WRITE data for the purpose of combining multiple WRITE operations or improving efficiency.

When application data spans a data block in a client cache, delayed transmission of WRITE data can result in clients modifying stale data and overwriting updates written by others. Prompt transmission of WRITE data enables the prompt detection of write holes and reduces the risk of data corruption.

5.2. WRITE Durability

The uncacheable file data attribute does not, by itself, dictate the `stable_how` value a client uses on WRITE operations. The protocol-level requirement is the following durability invariant: when the application's write call returns successfully, the WRITE data MUST be durable on the server.

A client honoring the uncacheable file data attribute MAY satisfy this invariant by either:

- * issuing WRITES with `stable_how` of `FILE_SYNC4` or `DATA_SYNC4`, in which case the data is durable on the WRITE response, or
- * issuing WRITES with `stable_how` of `UNSTABLE4` and a COMMIT that completes before the application's write call returns. If the COMMIT response indicates a changed write verifier, the client MUST re-issue the affected WRITES from the application's buffer, which remains available for the duration of the write call.

Clients MUST NOT defer COMMIT past the point at which the application's write call returns, because no client-side copy of the WRITE data is retained beyond that point and the data could not otherwise be re-driven after a server reboot.

The transient retention of WRITE data needed to complete an in-flight UNSTABLE4 and COMMIT exchange is not considered "caching" for the purposes of this attribute. The attribute concerns the long-lived retention of file data for the purpose of satisfying future READs or combining future WRITES.

5.3. Read Caching

The uncacheable file data attribute may also influence the use of read caching. Retaining cached READ data while other clients concurrently modify disjoint byte ranges of the same file can result in read-modify-write operations based on stale data.

Clients SHOULD ensure that cached file data is not reused without first revalidating it.

At a minimum, clients MUST revalidate metadata necessary to ensure correctness of cached file data, including the change attribute and file size. These attributes provide the primary mechanism for detecting modification of file contents. Meeting this MUST requirement satisfies the general SHOULD obligation above.

Clients MAY revalidate additional attributes (e.g., modification time or change time) as required by their local semantics or application requirements.

Failure to perform such revalidation can result in the client presenting stale or inconsistent file state (e.g., incorrect size or timestamps) to the application.

Suppressing read caching in addition to suppressing write-behind caching can further reduce the risk of stale-data overwrite in multi-writer workloads. However, in some cases read caching may remain appropriate when another NFSv4.2 mechanism ensures a consistent view of the file, such as a delegation.

5.4. Relationship to Direct I/O

While similar in intent to `O_DIRECT` (`[OPEN-O_DIRECT]`) and `forcedirectio` (`[SOLARIS-FORCEDIRECTIO]`), the uncacheable file data attribute operates at the protocol level and is advisory. Clients retain flexibility in how they satisfy the requirements described above.

6. Setting the Uncacheable File Data Attribute

In some deployments, applications or administrative tools may request that the uncacheable file data attribute be set on a file in order to influence client behavior. For example, applications that require predictable data visibility or that would otherwise rely on mechanisms such as `O_DIRECT` may use this attribute as a protocol-visible hint to the server.

However, the setting of this attribute is subject to server policy. The server is responsible for determining whether a request to set or clear the attribute is permitted. This may depend on factors such as administrative configuration, export policy, or access control mechanisms.

Requests that are not permitted MUST be rejected using existing NFSv4 error codes (e.g., NFS4ERR_INVAL or NFS4ERR_PERM).

This document does not require a server to implement any particular policy, nor any particular means of configuring one. A server that always permits, or always refuses, requests to set or clear the attribute conforms to this document; what the protocol requires is only the error returned when a request is refused.

One possible deployment model is for a server or administrator to configure a mount (see [MOUNT]) option such that newly created files under a given export are marked as uncacheable file data. In such a configuration, a client may request setting of the attribute at file creation time, via the `createattrs` of `OPEN`.

This approach is conceptually similar in intent to the Solaris `forcedirectio` mount option (see [SOLARIS-FORCEDIRECTIO]), but differs in scope and visibility in that it allows `DIRECT-I/O`-like behavior to be applied without requiring changes to individual applications.

Unlike local mechanisms such as `forcedirectio`, the NFSv4.2 attribute is visible to all clients accessing the file and is intended to convey server-side knowledge or policy in a distributed environment.

Changes to the uncacheable file data attribute while a file is actively in use may not be immediately reflected in client behavior. A client that has already opened a file MAY continue to operate based on its existing caching behavior and is not required to immediately alter its behavior in response to a change in the attribute.

Clients are expected to observe attribute changes through normal NFSv4 mechanisms (e.g., `GETATTR` or revalidation) and apply updated behavior as appropriate for subsequent operations.

7. Implementation Status

Note to RFC Editor: please remove this section prior to publication.

There is a prototype Hammerspace server which implements the uncacheable file data attribute and a prototype Linux client which treats the attribute as an indication to use `O_DIRECT`-like behavior for file access.

For the prototype, all files created under the mount point have the `fattr4_uncacheable_file_data` set to be true.

Experience with the prototype indicates that the uncacheable file data attribute can provide many of the practical benefits of `O_DIRECT` without requiring application modification. For applications that issue well-formed I/O requests, this approach has been observed to improve performance in many cases, while also reducing memory pressure and CPU utilization in the NFS client.

8. XDR for Uncacheable Attribute

```
///  
/// typedef bool                fattr4_uncacheable_file_data;  
///  
/// const FATTR4_UNCACHEABLE_FILE_DATA    = 87;  
///
```

9. Extraction of XDR

This document contains the external data representation (XDR) [RFC4506] description of the uncacheable file attribute. The XDR description is presented in a manner that facilitates easy extraction into a ready-to-compile format. To extract the machine-readable XDR description, use the following shell script:

```
<CODE BEGINS>  
#!/bin/sh  
grep '^ *///' $* | sed 's?^ */// ??' | sed 's?^ *///$??'  
<CODE ENDS>
```

For example, if the script is named `'extract.sh'` and this document is named `'spec.txt'`, execute the following command:

```
<CODE BEGINS>  
sh extract.sh < spec.txt > uncacheable_prot.x  
<CODE ENDS>
```

This script removes leading blank spaces and the sentinel sequence `'///'` from each line. XDR descriptions with the sentinel sequence are embedded throughout the document.

Note that the XDR code contained in this document depends on types from the NFSv4.2 `nfs4_prot.x` file (generated from [RFC7863]). This includes both nfs types that end with a 4, such as `offset4`, `length4`, etc., as well as more generic types such as `uint32_t` and `uint64_t`.

While the XDR can be appended to that from [RFC7863], the code snippets should be placed in their appropriate sections within the existing XDR.

10. Security Considerations

The uncacheable file data attribute does not introduce new authentication or authorization mechanisms and does not alter existing NFSv4.2 access control semantics. All operations that set or clear the attribute are subject to existing access control and server policy.

In particular, a server **MUST** enforce appropriate authorization checks for SETATTR operations that modify the `fattnr4_uncacheable_file_data` attribute. The ability to set or clear the attribute may be restricted based on administrative configuration, export policy, or other server-defined criteria.

Because the attribute is visible to and may affect the behavior of multiple clients, servers **SHOULD** consider the implications of allowing unprivileged users to modify it. Inappropriate use of the attribute could impact performance or data access patterns for other clients accessing the same file.

The uncacheable file data attribute is advisory and does not provide a security boundary. Clients **MUST NOT** rely on the presence or absence of this attribute to make access control decisions.

Use of this attribute does not replace or modify existing cache consistency mechanisms or data integrity protections provided by NFSv4.2.

11. IANA Considerations

This document has no IANA actions.

12. References

12.1. 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/rfc/rfc2119>>.
- [RFC4506] Eisler, M., Ed., "XDR: External Data Representation Standard", STD 67, RFC 4506, DOI 10.17487/RFC4506, May 2006, <<https://www.rfc-editor.org/rfc/rfc4506>>.

- [RFC7862] Haynes, T., "Network File System (NFS) Version 4 Minor Version 2 Protocol", RFC 7862, DOI 10.17487/RFC7862, November 2016, <<https://www.rfc-editor.org/rfc/rfc7862>>.
- [RFC7863] Haynes, T., "Network File System (NFS) Version 4 Minor Version 2 External Data Representation Standard (XDR) Description", RFC 7863, DOI 10.17487/RFC7863, November 2016, <<https://www.rfc-editor.org/rfc/rfc7863>>.
- [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/rfc/rfc8174>>.
- [RFC8178] Noveck, D., "Rules for NFSv4 Extensions and Minor Versions", RFC 8178, DOI 10.17487/RFC8178, July 2017, <<https://www.rfc-editor.org/rfc/rfc8178>>.
- [RFC8881] Noveck, D., Ed. and C. Lever, "Network File System (NFS) Version 4 Minor Version 1 Protocol", RFC 8881, DOI 10.17487/RFC8881, August 2020, <<https://www.rfc-editor.org/rfc/rfc8881>>.

12.2. Informative References

- [MOUNT] Linux man-pages project, "mount(2) - mount filesystem", Linux Programmer's Manual, 2024, <<https://man7.org/linux/man-pages/man2/mount.2.html>>.
- [OPEN-O_DIRECT] Linux man-pages project, "open(2) - Linux system call for opening files (O_DIRECT)", 2024, <<https://man7.org/linux/man-pages/man2/open.2.html>>.
- [SOLARIS-FORCEDIRECTIO] Oracle Solaris Documentation, "mount -o forcedirectio - Solaris forcedirectio mount option", Solaris Administration Guide, 2023, <<https://docs.oracle.com/en/operating-systems/solaris/oracle-solaris/11.4/manage-nfs/mount-options-for-nfs-file-systems.html>>.

Acknowledgments

Trond Myklebust, Mike Snitzer, Jon Flynn, Keith Mannthey, and Thomas Haynes all worked on the prototype at Hammerspace.

Rick Macklem, Chuck Lever, Dave Noveck, Barry Leiba, Vijay Gurbani, Claudio Allocchio, and Jamie Koehl reviewed the document.

Chris Inacio, Chuck Lever, Brian Pawlowski, and Gorrry Fairhurst
helped guide this process.

Author's Address

Thomas Haynes
Hammerspace
Email: loghyr@gmail.com

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 24 January 2027

H. Zheng, Ed.
Huawei Technologies Co., Ltd.
Y. Lee
CRU
R. Casellas
CTTC
D. Ceccarelli
Cisco
July 2026

PCEP Extension for Flexible Grid Networks
draft-ietf-pce-flexible-grid-16

Abstract

This document provides the Path Computation Element Communication Protocol (PCEP) extensions for the support of Routing and Spectrum Assignment (RSA) in Flexible Grid networks.

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 2 January 2027.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Terminology	4
3. Requirements Language	4
4. Spectrum Assignment (SA) Object	4
4.1. Frequency Slot Selection TLV	7
4.2. Frequency Slot Restriction TLV	8
5. Encoding of an RSA Path Reply	10
5.1. Error Indicator	11
5.2. NO-PATH Indicator	11
6. Manageability Considerations	12
6.1. Control of Function and Policy	12
6.2. Information and Data Models	12
6.3. Verifying Correct Operation	12
6.4. Requirements on Other Protocols and Functional Components	13
6.5. Impact on Network Operation	13
7. Implementation Status	13
7.1. CTTC	13
8. Security Considerations	14
9. IANA Considerations	15
9.1. New PCEP Object	15
9.2. SA Object Flag Field	15
9.3. New PCEP TLVs	15
9.4. Frequency Slot Assignment Method Values	16
9.5. Frequency Slot Restriction TLV Action Values	16
9.6. Spectrum Allocation TLV Flag Field	17
9.7. New No-Path Reasons	17
9.8. New Error-Types and Error-Values	17
9.9. New Error-Values for Existing Error Type (24)	18
10. Acknowledgements	18
11. Contributors' Address	18
12. References	18
12.1. Normative References	18
12.2. Informative References	20
Authors' Addresses	21

1. Introduction

[RFC4655] defines a Path Computation Element (PCE)-based path computation architecture and explains how a Path Computation Element (PCE) can compute Label Switched Paths (LSP) in Multiprotocol Label Switching Traffic Engineering (MPLS-TE) and Generalized MPLS (GMPLS) networks at the request of Path Computation Clients (PCCs). A PCC is said to be any network component that makes such a request and can be, for instance, an Optical Switching Element within a Wavelength Division Multiplexing (WDM) network. The PCE, itself, can be located anywhere within the network, and can be within an optical switching element, a Network Management System (NMS) or Operational Support System (OSS), or can be an independent network server.

The PCE communications Protocol (PCEP) is the communication protocol used between a PCC and a PCE, and can also be used between cooperating PCEs. [RFC4657] sets out the common protocol requirements for PCEP. Additional application-specific requirements for PCEP are deferred to separate documents.

[RFC8780] provides the PCEP extensions for the support of Routing and Wavelength Assignment (RWA) in Wavelength Switched Optical Networks (WSO) based on the requirements specified in [RFC6163] and [RFC7449].

To allow efficient allocation of optical spectral bandwidth for systems that have high bit-rates, the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) has extended its Recommendations [ITU-T_G.694.1] to include an enhanced Dense Wavelength Division Multiplexing (DWDM) grid by defining a set of nominal central frequencies, channel spacings, and the concept of the "frequency slot". In such an environment, a data-plane connection is switched based on allocated, variable-sized frequency ranges within the optical spectrum, creating what is known as a flexible grid (flexi-grid). [RFC7698] provides Framework and Requirements for GMPLS-Based Control of Flexi-Grid Dense Wavelength Division Multiplexing (DWDM) Networks.

The term "Routing and Spectrum Assignment" (RSA) is introduced in [RFC7698] to refer to the process that determines a route and frequency slot for an LSP. Hence, when a path is computed, the spectrum assignment process determines the central frequency and slot width. The term "Spectrum Switched Optical Networks" is also introduced in [RFC7698] to refer to a flexi-grid enabled DWDM network, which can be controlled by a GMPLS or PCE control plane.

This document provides PCEP extensions to support RSA in Flexi-grid networks. The extensions in this document apply to point-to-point LSPs; support for point-to-multipoint (P2MP) LSPs is out of scope for this document and is left for future work.

Figure 1 shows one typical PCE-based implementation, which is referred to as the Combined Routing and Spectrum Assignment (RSA) [RFC7698]. With this architecture, the two processes of routing and spectrum assignment are accessed via a single PCE. This architecture is the base architecture from which the PCEP extensions are specified in this document.

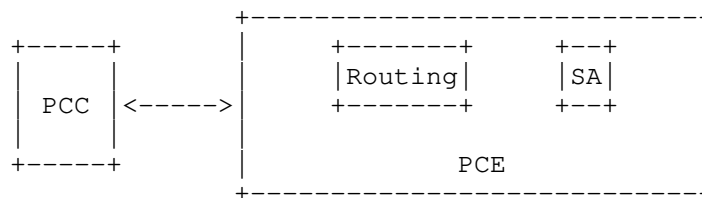


Figure 1: Combined Routing and Spectrum Assignment Architecture

2. Terminology

This document uses the terminology defined in [RFC4655], [RFC5440], and [RFC7698].

3. Requirements Language

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] when, and only when, they appear in all capitals, as shown here.

4. Spectrum Assignment (SA) Object

This document aligns with GMPLS extensions for PCEP [RFC8779] for generic properties such as label, label-set and label assignment, noting that frequency is a type of label. Frequency restrictions and constraints are also formulated in terms of labels per [RFC7579].

The SA Object enables a PCC to request spectrum assignment from the PCE and to express preferences such as assignment mode and frequency-slot restrictions. It thus allows the PCE to perform the combined Routing and Spectrum Assignment (RSA) according to the PCC's policies.

Spectrum allocation can be performed by the PCE by different means:

- a. By means of Explicit Label Control (ELC) where the PCE allocates which label to use for each interface/node along the path.
- b. By means of a Label Set where the PCE provides a range of potential frequency slots to allocate by each node along the path.

Option b. allows distributed spectrum allocation (performed during signaling) to complete spectrum assignment. When a range of potential frequency slots is given for allocation, a PC Request SHOULD convey the heuristic or mechanism to be used for the allocation.

The format Routing Backus-Naur Form (RBNF) [RFC5511] of a PCReq message per [RFC5440] after incorporating the Spectrum Assignment (SA) Object is as follows:

```
<PCReq Message> ::= <Common Header>
                        [<svec-list>]
                        <request-list>

Where:
<request-list> ::= <request> [<request-list>]
<request> ::= <RP>
                <GENERALIZED ENDPOINTS>
                [<SA>]
                [other optional objects...]
```

If the SA Object is present in the PCReq message, it MUST be encoded after the GENERALIZED ENDPOINTS Object.

This document specifies the use of the SA Object in the PCReq message. The use of the SA Object in other PCEP messages, such as the LSP Initiate Request (PCInitiate) message defined in [RFC8281], is out of scope for this document and is left for future work.

The SA Object-Class is TBD1 (to be assigned by IANA). The SA Object-Type is 1.

This document does not define a new PCEP capability advertisement for the SA Object. If a PCE receives an RSA request and the PCE is not capable of RSA computation, it reports the condition using the error procedure defined in Section 5.1. This error-based approach avoids adding a new OPEN Object capability for this extension; operators can use the PCE Discovery mechanisms described in Section 6.4 to advertise Flexi-Grid RSA path computation capabilities to PCCs.

The format of the Spectrum Assignment (SA) Object body is as shown in Figure 2.

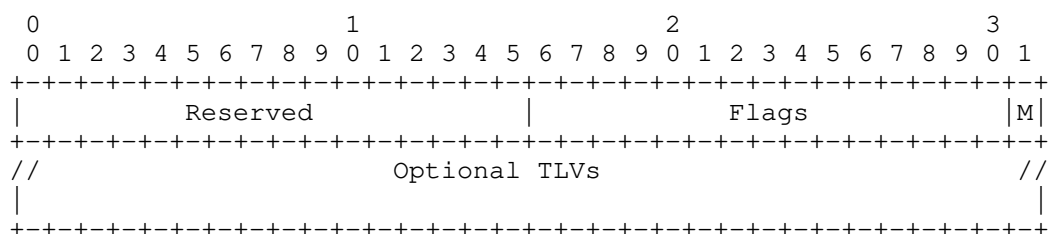


Figure 2: SA Object

Reserved (16 bits): the Reserved field MUST be set to zero by the sender and ignored by the receiver.

Flags (16 bits): bits 0-14 are unassigned. They MUST be set to zero by the sender and ignored by the receiver.

One flag bit is allocated as follows:

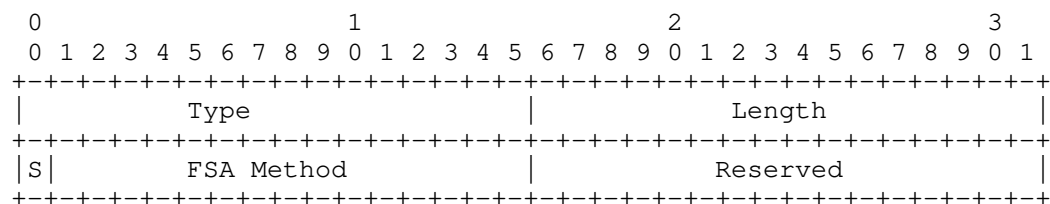
M (Mode - bit 15): M bit is used to indicate the mode of spectrum assignment. When the M bit is set to 1, this indicates that the spectrum assigned by the PCE MUST be explicit. That is, the selected way to convey the allocated spectrum is by means of Explicit Label Control (ELC) [RFC3472] for each hop of a computed LSP. Otherwise, when the M bit is set to 0, the spectrum assigned by the PCE does not need to be explicit (i.e., it can be suggested in the form of Label Set Objects in the corresponding response, to allow distributed SA). In such case, the PCE MUST return a Label Set Field as described in Section 2.6 of [RFC7579] in the response. See Section 5 of this document for the encoding discussion of a Label Set Field in a PCRep message.

IANA maintains the "SA Object Flag Field" registry requested in Section 9.2.

4.1. Frequency Slot Selection TLV

The Frequency Slot Selection TLV is used to indicate the frequency slot selection constraint in regard to the order of frequency slot assignment to be returned by the PCE. This TLV is only applied when the M bit is set in the SA Object specified in Section 4. This TLV SHOULD NOT be present and MUST be ignored when the M bit is set to 0.

The Frequency Slot Selection TLV format is defined as:



Type (16 bits): The type of the TLV (TBD2).

Length (16 bits): The length of the Value portion of the TLV in bytes. The Type and Length fields and any padding are not included in the Length field.

S (Symmetry - 1 bit): This flag is only meaningful when the request is for a bidirectional LSP, i.e., when the B (Bi-directional) flag is set in the RP Object (see Section 7.4.1 of [RFC5440]). 0 denotes requiring the same frequency slot in both directions; 1 denotes that different spectrum in the two directions is allowed.

Frequency Slot Assignment (FSA) Method (15 bits):

- * 0: unspecified (any); This does not constrain the SA method used by a PCC. This value is implied when the Frequency Slot Selection TLV is absent.
- * 1: First-Fit. All the feasible frequency slots are numbered (based on 'n' parameter), and this SA method chooses the available frequency slot with the lowest index, where 'n' is the parameter in $f = 193.1 \text{ THz} + n \times 0.00625 \text{ THz}$ where 193.1THz is the ITU-T 'anchor frequency' and 'n' is a positive integer including 0 [RFC7698].
- * 2: Random. This SA method chooses a feasible frequency slot value of 'n' randomly.

- * 3-32767: Unassigned.
- IANA maintains the "Frequency Slot Assignment Method Values" registry requested in Section 9.4.
- Reserved (16 bits): the Reserved field MUST be set to zero by the sender and ignored by the receiver.
- If a PCE does not support the attribute(s), its behavior is specified below:
- * S bit clear not supported: a PathErr MUST be generated with the Error Code "Routing Problem" (24) with error sub-code "Unsupported Frequency Slot Selection Symmetry value" (TBD3).
 - * FSA method not supported: a PathErr MUST be generated with the Error Code "Routing Problem" (24) with error sub-code "Unsupported Frequency Slot Assignment value" (TBD4).

4.2. Frequency Slot Restriction TLV

For any request that contains a frequency slot assignment, the requester (PCC) MUST be able to specify a restriction on the frequency slots to be used. This restriction is to be interpreted by the PCE as a constraint on the tuning ability of the origination laser transmitter or on any other maintenance related constraints.

The Frequency Slot Restriction TLV type is TBD5 (to be assigned by IANA). This TLV MAY appear more than once to be able to specify multiple restrictions. The TLV data is defined as shown in Figure 3.

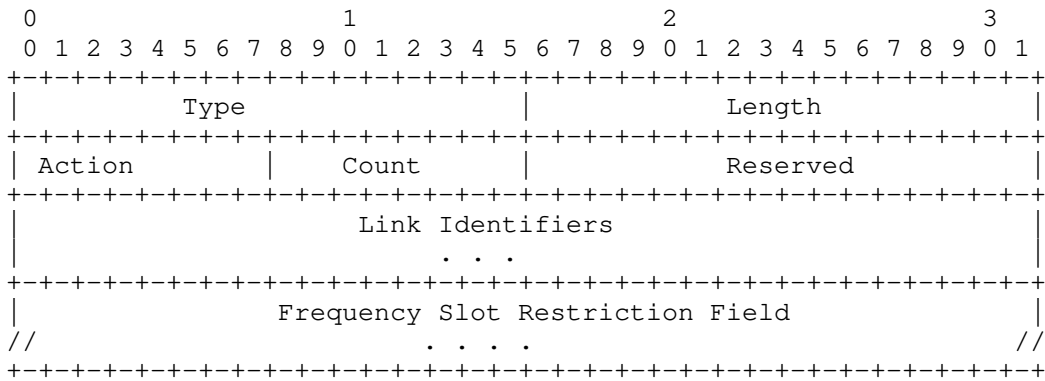


Figure 3: Frequency Slot Restriction TLV Encoding

The fields in the TLV are as follows:

* Type (16 bits): The type of the TLV (TBD5).

Length (16 bits): The length of the Value portion of the TLV in bytes. The Type and Length fields and any padding are not included in the Length field.

Action: 8 bits.

- 0 - Inclusive List indicates that one or more link identifiers are included in the Link Set. Each identifies a separate link that is part of the set.
- 1 - Inclusive Range indicates that the Link Set defines a range of links. It contains two link identifiers. The first identifier indicates the start of the range (inclusive). The second identifier indicates the end of the range (inclusive). All links with numeric values between the bounds are considered to be part of the set. A value of zero in either position indicates that there is no bound on the corresponding portion of the range. Note that the Action field can be set to 0 when unnumbered link identifier is used.

* Count: The number of the link identifiers (8 bits).

* Reserved: Reserved for future use (16 bits). It MUST be set to zero by the sender and ignored by the receiver.

* Link Identifiers: Identifies each link ID for which restriction is applied. The length is dependent on the link format and the Count field. See Section 4.3.1 in [RFC8780] for Link Identifier encoding.

IANA maintains the "Frequency Slot Restriction TLV Action Values" registry requested in Section 9.5.

A PCC MAY add a frequency slot restriction that applies to all links by setting the Count field to zero and specifying just a set of frequency slots.

All link identifiers in the same list MUST be of the same type.

The Frequency Slot Restriction Field of the Frequency Slot Restriction TLV is encoded as defined in Section 4.2 of [RFC8363].

5. Encoding of an RSA Path Reply

This section provides the encoding of an RSA Path Reply, in the PCRep/PCUpd message, for frequency slot allocation as discussed in Section 4. The Spectrum Allocation TLV type is TBD6 (to be assigned by IANA). The Spectrum Allocation TLV uses the standard PCEP TLV format defined in Section 7.1 of [RFC5440]. The TLV is defined as shown in Figure 4.

[RFC7570] describes how an attribute TLV ([RFC5420]) can be carried in an ERO as a TLV within an LSP Attribute Subobject to provide a per-hop description of an LSP attribute. The Spectrum Assignment TLV can be carried in the LSP Attribute Subobject to indicate the spectrum to be assigned on the identified link.

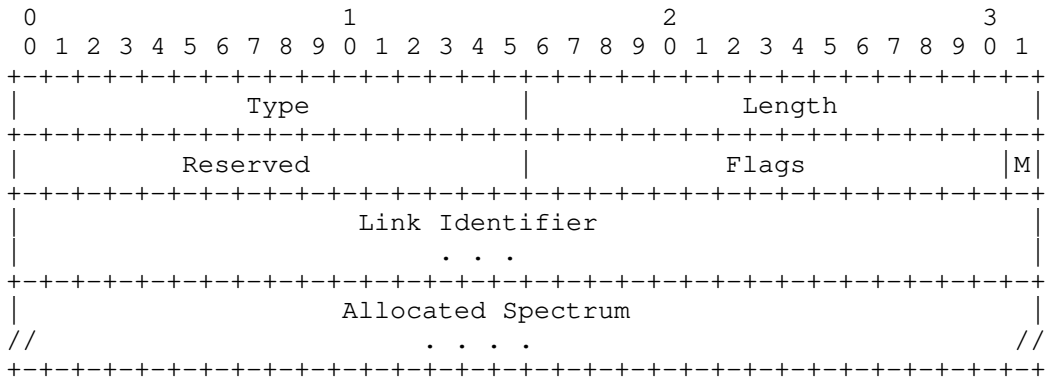


Figure 4: Spectrum Allocation TLV Encoding

- * Type (16 bits): The type of the TLV (TBD6).
- * Length (16 bits): The length of the Value portion of the TLV in bytes. The Type and Length fields and any padding are not included in the Length field.
- * Reserved (16 bits): The Reserved field MUST be set to zero by the sender and ignored by the receiver.
- * Flags (16 bits): bits 0-14 are unassigned. They MUST be set to zero by the sender and ignored by the receiver.
- * M (Mode - bit 15): 1 bit
 - 1 indicates the allocation is under Explicit Label Control.

- 0 indicates the allocation is expressed in Label Sets.
- * Link Identifier: Identifies the link ID for which the spectrum allocation is applied. See Section 4.3.1 in [RFC8780] for Link Identifier encoding.
- * Allocated Spectrum (variable): Indicates the spectrum allocated to the link identifier. The field carries one or more Flexi-Grid Labels encoded as defined in Section 4.1 of [RFC7699]. A single allocated slot is encoded as one 64-bit Flexi-Grid Label; multiple allocated slots are encoded as the compound label described in Section 4.3 of [RFC7699].

IANA maintains the "Spectrum Allocation TLV Flag Field" registry requested in Section 9.6.

5.1. Error Indicator

To indicate errors associated with the RSA request, a new Error-Type (TBD7, Flexi-Grid RSA Error) and subsequent error-value are defined for inclusion in the PCEP-ERROR Object:

- * Error-Type=TBD7 (Flexi-Grid RSA Error); Error-value=1: if a PCE receives an RSA request and the PCE is not capable of RSA computation, the PCE MUST send a PCErr message with a PCEP-ERROR Object (Error-Type=TBD7) and an Error-value (Error-value=1). The PCE stops processing the request. The corresponding RSA computation MUST be cancelled at the PCC.

Note that generic resource exhaustion at the PCE (e.g., insufficient memory) is not signaled via this Error-Type; the PCEP overload mechanism (see Section 7.15 of [RFC5440]) applies.

5.2. NO-PATH Indicator

To communicate the reasons for not being able to find RSA for the path request, the NO-PATH Object can be used in the corresponding response. The format of the NO-PATH Object body is defined in [RFC5440]. The object MAY contain a NO-PATH-VECTOR TLV to provide additional information about why a path computation has failed.

One new bit flag is defined to be carried in the Flags field in the NO-PATH-VECTOR TLV carried in the NO-PATH Object.

- * Bit TBD8: When set, the PCE indicates no feasible path was found that meets all the optical spectrum constraints associated with the path computation request.

This flag is specific to spectrum-related path computation failures; other path-computation failure reasons continue to use the existing PCEP NO-PATH mechanisms.

6. Manageability Considerations

Manageability of flexi-grid Routing and Spectrum Assignment (RSA) with PCE includes the following considerations:

6.1. Control of Function and Policy

In addition to the parameters already listed in Section 8.1 of [RFC5440], a PCEP implementation SHOULD allow configuring the following PCEP session parameters on a PCC:

- * The ability to send a Flexi-Grid RSA request.

In addition to the parameters already listed in Section 8.1 of [RFC5440], a PCEP implementation SHOULD allow configuring the following PCEP session parameters on a PCE:

- * The support for Flexi-Grid RSA.
- * A set of Flexi-Grid RSA specific policies (authorized sender, request rate limiter, etc).

These parameters can be configured as default parameters for any PCEP session the PCEP speaker participates in, or can apply to a specific session with a given PCEP peer or a specific group of sessions with a specific group of PCEP peers.

6.2. Information and Data Models

Extensions to the PCEP YANG module can be defined to cover the Flexi-Grid RSA information introduced in this document. Liveness Detection and Monitoring Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in Section 8.3 of [RFC5440].

6.3. Verifying Correct Operation

Mechanisms defined in this document do not imply any new verification requirements in addition to those already listed in section 8.4 of [RFC5440].

6.4. Requirements on Other Protocols and Functional Components

The PCE Discovery mechanisms ([RFC5088] and [RFC5089]) can be used to advertise Flexi-Grid RSA path computation capabilities to PCCs.

6.5. Impact on Network Operation

Mechanisms defined in this document do not imply any new network operation requirements in addition to those already listed in Section 8.6 of [RFC5440].

7. Implementation Status

[NOTE TO RFC EDITOR: This whole section and the reference to [RFC7942] are to be removed before publication as an RFC.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

7.1. CTTC

CTTC has implemented several aspects of the extensions defined by earlier revisions of this document in a research Path Computation Element and SDN controller. The PCC and PCE are built from the same codebase, and the implementation uses private codepoints for the new object and TLVs pending IANA allocation. Details are as follows:

Organization: Centre Tecnologic de Telecomunicacions de Catalunya (CTTC)

Implementation: Research PCE and SDN controller.

Description: The Spectrum Assignment (SA) Object is implemented using a private Object-Class codepoint and is encoded after the ENDPOINTS Object. The M (Mode) flag is supported for spectrum assignment by means of Explicit Label Control (ELC), which is also the default behavior when the SA Object is absent; the spectrum assignment algorithms in use allocate ELC, and requests for non-explicit assignment raise an error. The Frequency Slot Selection TLV is implemented, including the S (Symmetry) bit (the default for bidirectional LSPs) and the First-Fit (default) and Random assignment methods. The Frequency Slot Restriction TLV is partially implemented: the TLV is parsed in full, and restrictions that apply to all links (Count field set to zero) are supported, in practice as a single frequency slot that applies end to end for fixed transceivers. The new error codes are not implemented; generic PCEP errors (e.g., NO-PATH) are used instead.

Maturity Level: Research prototype used in research projects; not aimed at interoperability or product release. Interoperability has been tested only between the PCC and PCE built from the same codebase.

Coverage: Partial.

Contact: ramon.casellas@cttc.es

8. Security Considerations

This document has no requirement for a change to the security models within PCEP. The security considerations described in [RFC5440] and the use of TLS to provide a secure transport for PCEP as described in [RFC8253] apply to the extensions defined in this document. The additional information distributed in order to address the RSA problem represents a disclosure of network capabilities that an operator might wish to keep private. Operators are encouraged to secure this information.

9. IANA Considerations

This document requests the following IANA actions.

9.1. New PCEP Object

IANA is requested to allocate a new object class from the "PCEP Objects" registry (<http://www.iana.org/assignments/pcep/pcep.xhtml#pcep-objects>):

Object-Class Value	Name	Object-Type	Reference
TBD1	SA	0: Reserved	
		1: Spectrum Assignment	[This.I-D]
		2-15: Unassigned	

9.2. SA Object Flag Field

IANA is requested to create the "SA Object Flag Field" registry within the "Path Computation Element Protocol (PCEP) Numbers" registry. New values are to be assigned by IETF Review [RFC8126]. Bits are numbered from bit 0 as the most significant bit.

Bit	Description	Reference
0-14	Unassigned	
15	Spectrum Assignment Mode	[This.I-D]

9.3. New PCEP TLVs

IANA is requested to allocate three new TLV types, the Frequency Slot Selection TLV, the Frequency Slot Restriction TLV, the Spectrum Allocation TLV. All these TLVs are from the "PCEP TLV Type Indicators" registry (<http://www.iana.org/assignments/pcep/pcep.xhtml#pcep-tlv-type-indicators>).

Value	Description	Reference
TBD2	Frequency Slot Selection	[This.I-D]
TBD5	Frequency Slot Restriction	[This.I-D]
TBD6	Spectrum Allocation	[This.I-D]

9.4. Frequency Slot Assignment Method Values

IANA is requested to create the "Frequency Slot Assignment Method Values" registry within the "Path Computation Element Protocol (PCEP) Numbers" registry. New values are to be assigned by IETF Review [RFC8126].

Value	Description	Reference
0	Unspecified	[This.I-D]
1	First-Fit	[This.I-D]
2	Random	[This.I-D]
3-32767	Unassigned	

9.5. Frequency Slot Restriction TLV Action Values

IANA is requested to create the "Frequency Slot Restriction TLV Action Values" registry within the "Path Computation Element Protocol (PCEP) Numbers" registry. New values are to be assigned by IETF Review [RFC8126].

Value	Description	Reference
0	Inclusive List	[This.I-D]
1	Inclusive Range	[This.I-D]
2-255	Unassigned	

9.6. Spectrum Allocation TLV Flag Field

IANA is requested to create the "Spectrum Allocation TLV Flag Field" registry within the "Path Computation Element Protocol (PCEP) Numbers" registry. New values are to be assigned by IETF Review [RFC8126]. Bits are numbered from bit 0 as the most significant bit.

Bit	Description	Reference
0-14	Unassigned	
15	Spectrum Allocation Mode	[This.I-D]

9.7. New No-Path Reasons

IANA is requested to allocate a new bit flag from the "PCEP NO-PATH-VECTOR TLV Flag Field" registry (<http://www.iana.org/assignments/pcep/pcep.xhtml#no-path-vector-tlv>).

Bit	Description	Reference
TBD8	No spectrum constraints met	[This.I-D]

9.8. New Error-Types and Error-Values

IANA is requested to allocate a new Error-Type and Error-value from the "PCEP-ERROR Object Error Types and Values" registry (<http://www.iana.org/assignments/pcep/pcep.xhtml#pcep-error-object>).

Error-Type	Meaning	Error-value	Reference
TBD7	Flexi-Grid RSA Error	1: RSA computation not supported	[This.I-D]

9.9. New Error-Values for Existing Error Type (24)

IANA is requested to allocate two new PathErr values for the Existing Error Type (24):

Error-Type	Meaning	Error-value	Reference
24	Routing Problem	TBD3: Unsupported Frequency Slot Selection Symmetry value	[This.I-D]
		TBD4: Unsupported Frequency Slot Assignment value	[This.I-D]

10. Acknowledgements

Thanks to Francesco Lazzeri for the technical contribution, and Quan Xiong, Dhruv Dhody and Adrian Farrel for useful comments. Thanks to Ketan Talaulikar for his detailed AD review and helpful suggestions.

11. Contributors' Address

Ricard Vilalta
CTTC
Spain

Email: ricard.vilalta@cttc.es

12. References

12.1. 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>>.
- [RFC3472] Ashwood-Smith, P., Ed. and L. Berger, Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions", RFC 3472, DOI 10.17487/RFC3472, February 2003, <<https://www.rfc-editor.org/info/rfc3472>>.
- [RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, DOI 10.17487/RFC5088, January 2008, <<https://www.rfc-editor.org/info/rfc5088>>.
- [RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, DOI 10.17487/RFC5089, January 2008, <<https://www.rfc-editor.org/info/rfc5089>>.
- [RFC5420] Farrel, A., Ed., Papadimitriou, D., Vasseur, JP., and A. Ayyangar, "Encoding of Attributes for MPLS LSP Establishment Using Resource Reservation Protocol Traffic Engineering (RSVP-TE)", RFC 5420, DOI 10.17487/RFC5420, February 2009, <<https://www.rfc-editor.org/info/rfc5420>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, DOI 10.17487/RFC5511, April 2009, <<https://www.rfc-editor.org/info/rfc5511>>.
- [RFC7570] Margaria, C., Ed., Martinelli, G., Balls, S., and B. Wright, "Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)", RFC 7570, DOI 10.17487/RFC7570, July 2015, <<https://www.rfc-editor.org/info/rfc7570>>.
- [RFC7579] Bernstein, G., Ed., Lee, Y., Ed., Li, D., Imajuku, W., and J. Han, "General Network Element Constraint Encoding for GMPLS-Controlled Networks", RFC 7579, DOI 10.17487/RFC7579, June 2015, <<https://www.rfc-editor.org/info/rfc7579>>.

- [RFC7699] Farrel, A., King, D., Li, Y., and F. Zhang, "Generalized Labels for the Flexi-Grid in Lambda Switch Capable (LSC) Label Switching Routers", RFC 7699, DOI 10.17487/RFC7699, November 2015, <<https://www.rfc-editor.org/info/rfc7699>>.
- [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>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.
- [RFC8363] Zhang, X., Zheng, H., Casellas, R., Gonzalez de Dios, O., and D. Ceccarelli, "GMPLS OSPF-TE Extensions in Support of Flexi-Grid Dense Wavelength Division Multiplexing (DWDM) Networks", RFC 8363, DOI 10.17487/RFC8363, May 2018, <<https://www.rfc-editor.org/info/rfc8363>>.
- [RFC8779] Margaria, C., Ed., Gonzalez de Dios, O., Ed., and F. Zhang, Ed., "Path Computation Element Communication Protocol (PCEP) Extensions for GMPLS", RFC 8779, DOI 10.17487/RFC8779, July 2020, <<https://www.rfc-editor.org/info/rfc8779>>.
- [RFC8780] Lee, Y., Ed. and R. Casellas, Ed., "The Path Computation Element Communication Protocol (PCEP) Extension for Wavelength Switched Optical Network (WSO) Routing and Wavelength Assignment (RWA)", RFC 8780, DOI 10.17487/RFC8780, July 2020, <<https://www.rfc-editor.org/info/rfc8780>>.

12.2. Informative References

- [ITU-T_G.694.1] ITU-T, "SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS; Digital networks; Spectral grids for WDM applications: DWDM frequency grid", ITU-T Rec. G.694.1, October 2020, <<https://www.itu.int/rec/T-REC-G.694.1>>.

- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC4657] Ash, J., Ed. and J.L. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, DOI 10.17487/RFC4657, September 2006, <<https://www.rfc-editor.org/info/rfc4657>>.
- [RFC6163] Lee, Y., Ed., Bernstein, G., Ed., and W. Imajuku, "Framework for GMPLS and Path Computation Element (PCE) Control of Wavelength Switched Optical Networks (WSOs)", RFC 6163, DOI 10.17487/RFC6163, April 2011, <<https://www.rfc-editor.org/info/rfc6163>>.
- [RFC7449] Lee, Y., Ed., Bernstein, G., Ed., Martensson, J., Takeda, T., Tsuritani, T., and O. Gonzalez de Dios, "Path Computation Element Communication Protocol (PCEP) Requirements for Wavelength Switched Optical Network (WSO) Routing and Wavelength Assignment", RFC 7449, DOI 10.17487/RFC7449, February 2015, <<https://www.rfc-editor.org/info/rfc7449>>.
- [RFC7698] Gonzalez de Dios, O., Ed., Casellas, R., Ed., Zhang, F., Fu, X., Ceccarelli, D., and I. Hussain, "Framework and Requirements for GMPLS-Based Control of Flexi-Grid Dense Wavelength Division Multiplexing (DWDM) Networks", RFC 7698, DOI 10.17487/RFC7698, November 2015, <<https://www.rfc-editor.org/info/rfc7698>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.

Authors' Addresses

Haomian Zheng (editor)
Huawei Technologies Co., Ltd.
Email: zhenghaomian@huawei.com

Young Lee
CRU
Email: younglee.tx@gmail.com

Ramon Casellas
CTTC
Email: ramon.casellas@cttc.es

Daniele Ceccarelli
Cisco
Email: dceccare@cisco.com

Internet Research Task Force
Internet-Draft
Intended status: Informational
Expires: 7 January 2027

J. FranÃ§ois
University of Luxembourg and Inria
A. Clemm
Independent
D. Papadimitriou
3NLab Belgium Research Center
S. Fernandes
Canada Post
S. Schneider
Digital Railway (DSD) at Deutsche Bahn
6 July 2026

Research Challenges in Coupling Artificial Intelligence and Network
Management
draft-irtf-nmrg-ai-challenges-06

Abstract

This document is intended to introduce the challenges to overcome when Network Management (NM) problems may require coupling with Artificial Intelligence (AI) solutions. On the one hand, many difficult NM problems still lack good solutions, or existing approaches come with significant limitations. Artificial Intelligence may help produce novel solutions to those problems. On the other hand, due to the high computational costs of AI solutions and stringent data privacy constraints, the distributed execution of AI workloads has become paramount. Consequently, networks must be operated efficiently to sustain these distributed processing requirements.

To identify the right set of challenges, the document defines a method based on the evolution and nature of NM problems. This will be done in parallel with advances and the nature of existing solutions in AI in order to highlight where AI and NM have already been coupled together or could benefit from a closer integration. So, the method aims at evaluating the gap between NM problems and AI solutions. Challenges are derived accordingly, assuming that solving these challenges will help to reduce the gap between NM and AI.

This document is a product of the Network Management Research Group (NMRG) of the Internet Research Task Force (IRTF). This document reflects the consensus of the research group. It is not a candidate for any level of Internet Standard and is published for informational purposes.

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 7 January 2027.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	3
2. Acronyms	5
3. Difficult problems in network management	6
4. High-level challenges in adopting AI in NM	10
5. AI techniques and network management	13
5.1. Problem type and mapping	13
5.1.1. Sub-challenge: Suitable Approach for Given Input . .	13
5.1.2. Sub-challenge: Suitable Approach for Desired Output	14
5.1.3. Sub-challenge: Tailoring the AI Approach to the Given Problem	15
5.2. Performance of produced models	16
5.3. Lightweight AI	18
5.4. Distributed AI	20
5.4.1. Network management for efficient distributed AI . . .	20
5.4.2. Distributed AI for network management	21
5.5. AI for planning of actions	21
6. Network data as input for ML algorithms	23

6.1. Data for AI-based NM solutions	24
6.2. Data collection	25
6.3. Usable data	26
7. Acceptability of AI	28
7.1. Explainability of Network-AI products	28
7.2. AI-based products and algorithms in production systems .	29
7.3. AI with humans in the loop	31
8. Conclusions and Future Research Directions	32
9. Security considerations	33
9.1. AI-based security solutions	33
9.2. Security of AI	34
9.3. Relevance of AI-based outputs	35
10. IANA Considerations	35
11. References	35
11.1. Normative References	35
11.2. Informative References	36
Acknowledgments	48
Authors' Addresses	48

1. Introduction

The functional scope of Network Management (NM) is very large, ranging from monitoring to accounting, from network provisioning to service diagnostics, from usage accounting to security. The taxonomy defined in [Hoo18] extends the traditional Fault, Configuration, Accounting, Performance, Security (FCAPS) domains by considering additional functional areas but above all by promoting additional views. For instance, network management approaches can be classified according to the technologies, methods or paradigms they will rely on. Methods include common approaches as for example mathematical optimization or queuing theory but also techniques which have been widely applied in last decades like game theory, data analysis, data mining and machine learning. In management paradigms, autonomic and cognitive management are listed. As highlighted by this taxonomy, the definition of automated and more intelligent techniques has been promoted to support efficient network management operations.

Research in NM and networking has been very active in applied ML [Bou18] and has enabled progress in areas such as network security [Ke23], Software-Defined Networks (SDN) [Ham21], and vehicular networks [Tan21]. This broad adoption has led to a strong interlinkage between networking and AI, laying part of the foundation for future systems such as 6G or B5G networks that are expected to be AI-native [Sha25].

However, for maintaining network operational in pre-defined safety bounds, NM still heavily relies on established procedures. Even after several cycles of adding automation, these procedures are still

mostly fixed and set offline in the sense that the exact control loops and all possible scenarios are defined in advance. They are so mostly deterministic by nature or at least with sufficient safety margin. Obviously, there have been a lot of propositions to make network smarter or more intelligent with the use of Machine Learning (ML) but without large adoption for running real networks because it changes the paradigms towards stochastic methods.

ML includes regression analysis, statistical learning (SVM and variants), deep learning (ANN and variants), Reinforcement Learning (RL), Large Language Models (LLMs), etc. It is a sub-area of Artificial Intelligence (AI) that concentrates the focus nowadays but AI encompasses other areas including knowledge representation, inductive logic programming, inference rule engine or by extension the techniques that allow one to observe and perform actions on a system.

It is thus legitimate to question whether ML or AI in general could be helpful for NM in practical deployment. This question is closely tied to the problems NM aims to address. Independently of NM, ML-based solutions were introduced to solve problems in an approximate way when they are very complex in nature, i.e., when finding an optimal solution is not possible in polynomial time. This is the case for NP-hard problems. In those cases, solutions typically rely on heuristics that may not yield optimal results, or algorithms that run into issues with scalability and the ability to produce timely results due to the exponential search space. In NM, those problems exist: allocation of resources in case of Service Function Chaining (SFC) or network slicing among others are recent examples which have gained interest in our community with SDN. Many proposed approaches consist of modelling the optimization problem as a MILP (Mixed-Integer Linear Programming) and solve it by the means of heuristics to reach a satisfactory trade-off between solution quality (gap to the optimal solution) – computation time and model size/dimensionality. Hence, ML is recognized to be well adapted to progress on this type of problem [Kaf19].

However, not all computational problems in network management are NP-hard. Due to real-time constraints, some involve very short control loops that require both rapid decisions and the ability to rapidly adapt to new situations and different contexts. So, even in that case, time is critical, and approximate solutions are usually more acceptable. Again, it is where AI can be beneficial. Actually, expert systems are AI systems [Ste92] but this kind of rule-based systems are not designed to scale with the volume and heterogeneity of data we can collect in a network today. In contrast, ML is more efficient to automatically learn abstract representations of the rules, which can be eventually updated.

However, not all computational problems in NM are NP-hard. Due to real-time constraints, some involve very short control loops that require both rapid decisions and the ability to quickly adapt to new situations and different contexts. In such cases, time is critical, and approximate solutions are often acceptable. This is another domain where AI can be beneficial. Expert systems are AI systems [Ste92], but rule-based systems are not designed to scale with the volume and heterogeneity of data that can be collected in a network today. In contrast, ML is better suited to automatically learn abstract rule structures that can be updated over time.

Another common problem in NM is classification. For instance, classifying network flows is helpful for security purposes to detect attack flows and to differentiate QoS among flows (e.g., real-time streams that require prioritization). ML-based classification algorithms have been widely used with satisfactory results when properly applied and have been adopted in commercial products. There are many algorithms including decision trees, Support Vector Machine (SVM) or (deep) Neural Networks (NNs) which have been proven to be effective in many areas and notably for image and natural language processing.

Finally, many problems still involve humans in the loop: from support issues such as handling trouble tickets to planning activities for the rollout of new services. This creates operational bottlenecks and is often expensive and error-prone. These tasks could be either automated or guided by an AI system to avoid individual human bias. It is worth noting that ML models trained on human-generated data can also indirectly suffer from collective human bias. Indeed, the balance between human resources and problem complexity is increasingly skewed and will continue to worsen due to growing network size, device heterogeneity, service diversity, and more. As a result, human-based procedures tend to be either overly simplistic for the problems they address or excessively time-consuming. Notable examples are in security, where network operators must defend against potential unknown threats.

All these problems are exacerbated by increasingly complex networks operating across many dimensions (users, devices, services, connections, etc.). Therefore, AI is expected to enable or simplify solving these problems in real networks in the near future [czb20] [Yan20], as they require unprecedented levels of performance in terms of throughput, latency, mobility, security, and more.

2. Acronyms

* AI: Artificial Intelligence

- * CNN: Convolutional Neural Network
- * FL: Federated Learning
- * GAN: Generative Adversarial Network
- * GNN: Graph Neural Network
- * IBN: Intent-Based Networking
- * LLM: Large Language Model
- * LSTM: Long Short-Term Memory
- * MAE: Mean Absolute Error
- * ML: Machine Learning
- * MILP: Mixed-Integer Linear Programming
- * MLP: Multilayer Perceptron
- * MSE: Mean Squared Error
- * NM: Network Management
- * RL: Reinforcement Learning
- * SDN: Software-Defined Networks
- * SFC: Service Function Chaining
- * SVM: Support Vector Machine
- * VNF: Virtual Network Function

3. Difficult problems in network management

As mentioned in the introduction, problems to be tackled in NM tend to be complex and exhibit characteristics that make them good candidates for AI-based solutions. We summarize these characteristics with short names to improve readability and to help the reader track how they recur throughout the document:

- * C1: Fundamental Algorithmic Complexity -- Managing a very large solution space that combinatorially explodes with the scale and topological footprint of the network domain, making it computationally impractical to explore and evaluate every possible state space (such as in NP-hard problems).
- * C2: Uncertainty and Unpredictability -- Coping with highly dynamic, non-deterministic variables along multiple operational dimensions, including shifting user traffic patterns, lack of visibility into end-to-end network states, and myriads of multi-domain provider interdependencies that cannot be modeled or known in advance.
- * C3: Real-Time Constraints -- Computing analytical solutions, issuing telemetry verdicts, and executing closed-loop control actions within strictly bounded or deterministic timeframes, especially where the context changes dynamically and decisions must be implemented rapidly to remain useful.
- * C4: Data Dependency -- Processing and analyzing massive volumes of network telemetry to derive accurate management decisions, which requires dealing with compound data-plane challenges ranging from data heterogeneity and incomplete or missing datasets to high data velocity.
- * C5: Integration with Existing Practice -- Interoperating seamlessly with pre-existing automated procedures, legacy deterministic control loops, heterogeneous operator workflows, and established human-in-the-loop administrative structures.
- * C6: Economics -- Ensuring cost-effectiveness across limited physical and operational resources (including available link bandwidth, node CPU, and human overhead), particularly when model training or inference processing is distributed at the network edge or embedded within data-plane elements.

Characteristic Tag & Short Name	Relevant Document Sections / Gaps
C1: Fundamental Algorithmic Complexity	Section 5.1 (Problem Mapping); Section 5.5 (Action Planning); Section 5.4 (Distributed AI)
C2: Uncertainty and Unpredictability	Section 5.5 (Action Planning); Section 5.2 (Performance of produced models); Section 9.2 (Security of AI); Section 9.3 (Relevance of AI-based outputs)
C3: Real-Time Constraints and Determinism	Section 5.3 (Lightweight AI); Section 7.2 (Production Systems); Section 5.2 (Performance of produced models)
C4: Data Dependency and Heterogeneity	Section 6.1 (Network data as input for ML algorithms); Section 6.2 (Data collection); Section 6.3 (Usable data)
C5: Integration with Existing Practice	Section 7.3 (AI with humans in the loop); Section 5.4.1 (NM for AI); Section 7.1 (Explainability)
C6: Economics and Computational Footprint	Section 5.3 (Lightweight AI); Section 7.2 (Production Systems)

Table 1: Problem characteristics

Many decision/optimization problems are affected by multiple criteria. Below is a non-exhaustive list of complex NM problems for which AI and/or non-AI-based approaches have been proposed:

- * Computation of optimal paths: Packet forwarding is not always based on traditional least-cost routing protocols, but on computation of paths that are optimized for certain criteria - for example, to meet certain level objectives, to result in greater resilience to balance utilization, to optimize energy usage, etc. Many of those solutions can be found in SDN, where a controller or path computation element computes paths that are subsequently provisioned across the network. However, such solutions generally do not scale to millions of paths (C1) and cannot be recomputed in sub-second time scales (C3) to take into account dynamically changing network conditions (C2). To compute those paths,

operations research techniques have been extensively used in literature along with AI methods [Lop20]. Mobility and dynamicity are two conditions that make the problem of computing optimal paths even harder. Hence, adaptive routing based on RL has been proposed as it allows an agent to promptly react to topological changes [Sin22]. As such, this problem can be considered as close to big data problems with some of the different Vs: volume, velocity, variety, value\200;

- * **Classification of network traffic:** Without loss of generality, a common objective of network monitoring for operators is to know the type of traffic going through their networks (web, streaming, gaming, VoIP). Such a task analyses data (C4) which can vary over time (C2) except in very particular scenarios like industrial isolated networks. However, the output of the classification technique is time-constrained only in specific cases where fast decisions must be made, for example to reroute traffic. Simple identification based on IANA-assigned TCP/UDP port numbers were sufficient in the past. However, with applications using dynamic port numbers, signature techniques can be used to match packet payload [Sen04]. To handle applications now encapsulated in encrypted web or VPN traffic, various ML-based approaches have been thus adopted [Bri19][Naj24][Wan24][Ake24] including LLMs [Gin24].
- * **Network diagnostics:** Disruptions of networking services can have many causes and thus can rely on analysing many sources of data (C4). Identifying the root causes is of high importance, so that repair actions can address them versus just working around the symptoms. Such repair actions may involve human actions (C5). Further complicating the matter are scenarios in which disruptions are not complete but involve only a degradation of service level, and where disruptions are intermittent, not reproducible, and hard to predict. Artificial intelligence techniques can offer promising solutions. Especially, anticipation of faults is of paramount importance and will lead to the development of predictive maintenance in future networks[Mut24].
- * **Network observability:** obtaining deeper insight into network status can rely on monitoring techniques to gather data from various sources. A major issue is to aggregate all these data in a valuable format [Zha21]. When it is not directly used to automate some actions, the aggregation of the data needs to be presented in an interpretable manner to human operators. In this area, visualisation techniques are helpful and also rely on AI techniques to provide the best outputs by reducing the number of dimensions (C4) and adapting the visualisation of data for human-operated processes (C5) [Ami24].

- * **Intent-Based Networking (IBN):** Roughly speaking, IBN refers to the ability to manage networks by articulating desired outcomes without the need to specify a course of [RFC9315]. The ability to determine such courses of actions, in particular with multiple interdependencies, conflicting goals, large scale, and highly complex and dynamic environments is a huge and largely unsolved challenge (C1, C2, C3). As an illustration, a major problem with intent is to interpret them correctly knowing that different intent formats have been proposed including natural language. Without good interpretation of the intent, i.e. the expected outcomes, the derived actions will not be adequate. In case the intent is correctly interpreted, a major problem is to find concrete solutions to realize the intents which implicitly requires optimizing the actions to be taken. Artificial Intelligence techniques can be of help here in multiple ways, from accurately classifying dynamic context to determine matching actions to reframing the expression of intent as a game that can be played (and won) using artificially intelligent techniques. As an example, LossLeaP even goes further by trying to predict the gap between a targeted objective and the predicted impact of the intent realization [Col22].
- * **VNF (Virtual Network Function) placement and SFC (Service Function Chain) design:** VNFs need to be placed on physical resources and SFCs designed in an optimized manner to minimize the use of networking resources and energy (C1,C6). As it is known to be a NP-hard problem, many heuristic- or machine-learning-based approaches have been proposed. The VNF paradigm actually emerges alongside 5G networking and orchestration methods [Att23].
- * **Smart admission control to avoid congestion and oversubscription of network resources:** Admission control needs to be set up to ensure service levels are optimized in a manner that is fair and aligned with application needs, congestion avoided, or its effects mitigated (C6). This field of research has notably been extended to the context of network slicing [Vin21][Sul23].

4. High-level challenges in adopting AI in NM

As shown in the previous section, AI techniques are good candidates for difficult NM problems. There have been many proposed solutions, but most remain at the prototype level or have been evaluated only through simulation and/or emulation. It is thus questionable why the NM community conducts extensive research in this direction but has not widely adopted those solutions to operate real networks. There are several obstacles.

First, AI advances have been historically driven by the image/video, natural language and signal processing communities as well as robotics for many decades. As a result, the most impressive applications are in this area including recently the generalization of home assistants, chatbots or the large progress in autonomous vehicles. However, the network experts have been focused on building the Internet, in particular designing protocols to make the world interconnected and with always better performance and services. This trend continues today with the 5G networks in deployment and beyond 5G under definition. Hence, AI was not the primary focus even if increased network automation calls for AI and ML solutions. However, AI is now considered as a core enabler for the future 6G networks which are sometimes qualified as AI-native networks [Sha25].

While we can see major contributions in AI-based solutions for networking over more than two decades, only a fraction of the community was concerned by AI at that time. Progress as a whole, from a community perspective, was therefore limited and compensated by relying on the development of AI in the communities as mentioned earlier. Even if our problems share some commonalities, for example on the volume of data to analyse, there are differences: data types are completely different, networks are by nature heavily distributed, etc. If problems are different, they should require distinct solutions or at least in-depth adaptation. In a nutshell, network-tailored AI has been overlooked, leading to a first set of challenges described in Section 5.

Second, many AI techniques require sufficiently representative data. For example, (deep) learning techniques mostly rely on having vectors of (real) numbers as input which fits naturally with numerical metrics (packet/byte counts, latency, delays, etc) but needs some adjustment for categorical (IP addresses, port numbers, etc) or topological features. Conversions are usually applied using common techniques like one-hot encoding or by coarse-grained representations [Scoll]. However, more advanced techniques can be proposed to embed representation of network entities rather than pure encoding as illustrated in [Rin17][Evr19][Sol20].

Besides, AI techniques that involve analysis of networking data can also lead to the extraction of sensitive and personally identifiable information, raising potential privacy concerns and concerns regarding the potential for abuse. Actually, this is a common and known problem that applies to many application domains [Liu22]. For example, AI techniques used to analyse encrypted network traffic with the legitimate goal to protect the network from intrusions and illegitimate attack traffic could be used to infer information about network usage and interactions of network users [Hoa21]. Intelligent data analysis and the need to maintain privacy are in many ways

contradictory in nature, resulting in an arms race. Similarly, training ML solutions on real network data is often preferable over using less-realistic synthetic data sets [Liu22b]. However, network data may contain private or sensitive data, the sharing of which may be problematic from a privacy standpoint and even result in legal exposure. The challenge concerns thus how to allow AI techniques to perform legitimate network management functions and provide network owners with operational insights into what is going on in their networks, while prohibiting their potential for abuse for other (illegitimate) purposes. Challenges related to network data as input to ML algorithms is detailed in Section 6.

Finally, networks are already operated thanks to (semi-)automated procedures involving many resources which are synchronized with management or orchestration tools. Adding AI therefore requires seamless integration within pre-existing processes. Although the goal of these procedures might be solely to provide relevant information to operators through alerts or dashboards in case of monitoring applications, this can be defined to trigger actions on the different resources, which can be local or remote. The use of AI or any other approaches to derive NM actions adds further constraints on them, especially regarding time constraints [Liu21] and synchronization to maintain a coherence over a distributed system.

A related challenge concerns the fact that to be deployed, a solution needs to provide a technical solution but also be acceptable to users – in this case, network administrators and operators. With automated solutions concerns that users want to feel “in control” and able to

understand what is going on, even more so if ultimately those users are held accountable for whether or not the network is running smoothly. To mitigate those concerns, aspects such as the ability to explain actions that are taken – or about to be taken – by AI systems become important [Sen24].

Beyond reasons of making users more comfortable, there are potentially also legal or regulatory ramifications to ensure that actions taken are properly understood. For example, agencies such as the FCC may impose fines on network operators when services such as E911 experience outages. In investigating causes for such outages, the underlying behaviour of the systems has to be properly understood, and even more so the reasons for actions that fall under the realm of network operations. All these aspects about integration and acceptability of the integration of AI in NM processes is detailed in Section 7.

5. AI techniques and network management

5.1. Problem type and mapping

An increasing number of different AI techniques have been proposed and applied successfully to a growing variety of different problems in different domains, including network management [Mus18], [Xie18]. Some of the most recently proposed AI approaches are clearly advancements of older approaches, which they supersede. Many other AI approaches are not predecessors or successors but simply complementary because they are useful for different problems or optimize different metrics. In fact, different AI approaches are useful for different kinds of problem inputs (e.g., tabular data vs. text vs. images vs. time series) and also for different kinds of desired outputs (e.g., a predicted value, a classification, or an action). Similarly, there may be trade-offs between multiple approaches that take the same kind of inputs and desired outputs (e.g., in terms of desired objective, computation complexity, constraints).

Overall, it is a key challenge of using AI techniques for network management to properly understand and map which kind of problems with which inputs, outputs, and objectives are best solved with which kind of AI (or non-AI) approaches. Given the wealth of existing and newly released AI approaches, this is far from a trivial task.

5.1.1. Sub-challenge: Suitable Approach for Given Input

Different problems in network management come with widely different problem parameters. For example, security-related problems may have large amounts of textual or encrypted data as input, whereas forecasting problems have historical time series data as input. They also vary in the amount of available data.

Both the type and amount of data influences the selection of an appropriate AI technique. On one hand, in scenarios with small dimensional data, classical machine learning techniques (e.g., SVM, tree-based approaches, etc.) are often sufficient and even superior to NNs [Grel9]. On the other hand, NNs have the advantage of learning complex models from large amounts of data without requiring feature engineering. Here, different neural network architectures are useful for different kinds of problems. The traditional and simplest architecture are (fully connected) Multi-Layer Perceptrons (MLPs), which are useful for structured, tabular data. For images, videos, or other high-dimensional data with correlation between adjacent features, convolutional neural networks (CNNs) are useful.

Recurrent neural networks (RNNs), especially the Long Short-Term Memory (LSTM) architecture, and attention-based neural networks

(transformers) are great for sequential data like time series or text. This evolution leads to the era of LLMs also impacting research in networking [Hua25][Wu24].

It is worth noting that Graph Neural Networks (GNNs) can incorporate and consider the graph-structured input, which is very useful in network management [Jie22], e.g., to represent the network topology.

The aforementioned rough guidelines can help identify a suitable AI approach or NN architecture. Still, best results are often achieved with a sophisticated combination of different approaches. For example, multiple elements can be combined into one architecture [Ham23], e.g., with both CNNs and LSTMs, and multiple separate AI approaches can be used as an ensemble to combine their strengths [Das23]. Here, simplifying the mapping from problem type and input to suitable AI approaches and architectures is clearly an open challenge. Future work should address this challenge by providing both clearer guidelines and striving for more general AI approaches that can easily be applied to a large variety of different problem inputs.

5.1.2. Sub-challenge: Suitable Approach for Desired Output

Similar to the challenge of identifying suitable AI approaches for a given problem input, the desired output for a given problem also affects which AI approach should be chosen. Here, the format of the desired output (single value, class, action, etc.), the frequency of these outputs and their meaning should be considered.

Again, there are rough guidelines for identifying a group of suitable AI approaches. For example, if a single numerical value is required (e.g., the amount of resources to allocate to a service instance), then a supervised regression approaches should be considered as a first candidate option as it is lightweight for this type of task. In case of classification (e.g., of malware or another security issue [Abdl0]) instead of predicting a value is desired, supervised methods can be used if labelled training data is available. There are also cases where a single class of training data is available, as for example in the context of anomaly detection where the model is fitted to normal data. In that case, one-class supervised techniques can be considered as a good candidate. Alternatively, unsupervised machine learning can help to cluster given data into separate groups, which can be useful to analyse networking data, e.g., for better understanding different types of traffic or user segments. Furthermore, the quality of the data [Liu22b]â\200~directly impacts on the robustness of a ML model with the risk of biased models due to overfitting. As highlighted with these few examples, finding a suitable approach to a problem depends on many factors including the type of

problem to handle but also other contextual elements such as the availability and the quality of data. To help in building AI-based solutions, pipeline generators have merged with automated capabilities, paving the way to the field of AutoML [Urb23].

In addition to these classical supervised and unsupervised methods, Reinforcement Learning (RL) approaches allows active, sequential decisions rather than simple predictions or classification. This is often useful in network management, e.g., to actively control service scaling and placement [Sah23]. RL agents autonomously select suitable actions in a given environment and are especially useful for self-learning network management. In addition to model-free RL, model-based planning approaches (e.g., Monte Carlo Tree Search) also allow choosing suitable actions but require full knowledge of the environment dynamics. In contrast, model-free RL is ideal for scenarios with unknown environment dynamics, which is often the case in network management.

Like the previous sub-challenge, these are just rough guidelines that can help to select a suitable group of AI approaches. Identifying the most suitable approach within the group, e.g., the best out of the many existing reinforcement learning approaches, is still challenging. And, as before, different approaches could be combined to enable even more effective network management (e.g., heuristics + RL, LSTMs + RL, etc.). Here, further research can simplify the mapping from desired problem output to choosing or designing a suitable AI approach.

5.1.3. Sub-challenge: Tailoring the AI Approach to the Given Problem

After addressing the two aforementioned sub-challenges, one may have selected a useful kind of AI approach for the given input and output of a network management problem. For example, one may select regression and supervised learning to forecast upcoming network traffic or select reinforcement learning to continuously control network and service coordination (scaling, placement, etc.). However, even within each of these fields (regression, reinforcement learning, etc.), there are many possible algorithms and hyper-parameters to consider. Selecting a suitable algorithm and parametrizing it with the right hyper-parameters is crucial to tailor the AI approach to the given network management problem.

For example, there are many different regression techniques (classical linear, polynomial regression, lasso/ridge regression, support vector regression, regression trees, neural networks, etc.), each with different benefits and drawbacks and each with its own set of hyper-parameters. Choosing a suitable technique depends on the amount and structure of the input data as well as on the desired

output. It also depends on the available amount of compute resources and compute time until a prediction is required. If resources and time are not a limiting factor, many hyper-parameters can be tuned automatically.

This sub-challenge holds for all fields of AI: supervised learning (regression and classification), self-supervised learning, unsupervised learning, and reinforcement learning, each are broad and rapidly growing fields. Selecting suitable algorithms and hyper-parameters to tailor AI approaches to the network management problem is both an opportunity and a challenge. Here, future work should further explore these trade-offs and provide clearer guidelines on how to navigate these trade-offs for different network management tasks. As already mentioned, the AutoML field of research provides solutions to better customize ML algorithms and pipeline. However, such kind of optimization should be optimized according to domain-specific metrics rather than pure-AI metrics only. For instance, the integration of network-specific knowledge can be done through human feedback [Arz21].

5.2. Performance of produced models

From a general point of view, any AI technique will produce results with a certain level of quality. This leads to two inherent questions: (1) what is the definition of the performance in a context of a NM application? (2) How to measure it? (3) How to ensure the quality of produced results by AI is aligned with NM objectives? (4) How to maintain or improve the quality of produced results?

Many metrics have been already defined to evaluate the performance of an AI-based technique according to its NM-level objectives. For example, QoS metrics (throughput, latency) can serve to measure the performance of a routing algorithm along with the computational complexity (memory consumption, size of routing tables). The question is to model and measure these two antagonist types of metrics. Number of true/false positives/negatives are the most basic metrics for network attack detection functions. Although the first two questions are thus already answered even if improvement can be done, question (3) refers to the integration of metrics into AI algorithms. Its objective is to obtain the best results which need to be quantified with these metrics. Depending on the type of algorithm, these metrics are either evaluated in an online manner with a feedback loop (for example with reinforcement learning) or in batch to optimize a model based on a particular context (for example described by a dataset for machine learning).

The problem is twofold. First, the performance can be measured through multiple metrics of different types (numerical or ordinal, for example), and some can be constrained by fixed boundaries (such as maximum latency), making their joint use challenging when creating an AI model to resolve an NM problem. Second, the scale of the metrics differs in importance or impact and can eventually vary across domains. It can be hard to precisely assess what a good or bad value is (as it might depend on multiple other ones) and it is even more difficult to integrate this in an AI technique, especially for learning algorithms to adjust their models based on performance. Indeed, many learning algorithms run through multiple iterations and rely on internal metrics, Mean Absolute Error (MAE) or Mean Squared Error (MSE) for neural network, Gini index or entropy for decision trees, distance to a hyperplane for SVMs, etc) which are not strongly correlated to the final operational metrics of the NM application. AI-internal metrics such as the loss do not match well the metrics related to the final NM objectives, thus the significance and impact of the AI errors cannot be easily translated into the NM domain.

For instance, a decision tree algorithm for classification purposes aims to create branches with a maximum of data from the same classes and to avoid mixing classes. It is done thanks to a criterion like the entropy index, but this kind of index does not assume any difference between mixing classes A and B or A and C. Assuming an operational point of view in which A and B mixing in the predictions is not critical, the algorithm should have preferred to mix A and B rather than A and C, even if in the first case it will produce more errors. Therefore, the internal functioning of the AI algorithms should be refined, specifically by defining a particular criterion to replace entropy as a quality measure when separating two branches. It assumes that the final NM objectives are integrated at this stage.

Another concrete example is traffic predictors, which aim to forecast traffic demands. They produce an output that is not necessarily simple to interpret and use by, for example, capacity allocation strategies/policies. A traditional traffic prediction that tries to minimize (perfectly symmetric) MAE/MSE treats positive and negative errors identically, hence is agnostic to the diverse meaning (and costs) of under- and over-provisioning. And, such a prediction does not provide any information on, for example, how to dimension resources/capacity to accommodate future demand while avoiding all under-provisioning (which causes service disruption) while minimizing over-provisioning (i.e., wasting resources). In other words, it forces the operator to guess the over-provisioning by taking uninformed safety margins. A more sensible approach is to forecast directly the needed capacity, rather than the traffic [Beg19].

While this is just an example, the high-level challenge is devising forecasting models that minimize the correct objective/loss function for the specific NM task at hand (instead of generic MAE/MSE). In this way, the prediction phase becomes an integral part of NM, and not just a limited and difficult-to-use input to it. In ML terms, this maps to solving the loss-metric mismatch in the context of anticipatory NM [Hua19].

Another issue for statistical learning (from examples/observations) is mainly about extracting an estimator from a finite set of input-output samples drawn from an unknown probability distribution that should be descriptive enough for unseen/new input data. In this context, online monitoring and error control of the quality/properties of these point estimators (bias, variance, mean squared error, etc.) is critical for dynamic/uncertain network environments. Similar reasoning/challenge applies to interval estimates, i.e., confidence intervals (frequentist) and credible intervals (Bayesian).

Finally, question (4) refers to the ability of an AI solution to remain efficient and eventually improve over time [Ans26]. This requires dynamic methods capable to adapt to a changing environment. As already highlighted, the models can be dynamically adjusted based on the errors they produced. In the context of ML, the models can be also updated based on new data, either through a complete re-learning phase, fine-tuning or transfer-learning. This assumes to collect and ingest continuously new data. However, as highlighted in [Sha21b], this type of ML, qualified as online or incremental, raises several challenges when applied to traffic analysis. For example, there is a set of related challenges related to select or discard some data over a time horizon and to label data in real-time. Other challenges are more generic to this ML research area such as class imbalance or concept drift.

5.3. Lightweight AI

Network management and operations often need to be performed under strict time constraints, i.e., at line rate, particularly in autonomic or self-driven networks. Locating NM functions as close as possible to where forwarding is achieved is an interesting option to avoid additional delays when these operations are performed remotely, for example in a centralized controller. Besides, forwarding devices may offer available resources to supplement or replace edge resources. For AI coupled with network management, AI tasks can be offloaded in network devices or more generally embedded within the network. Obviously, time-critical tasks are the best candidates to be offloaded within the network. Costly learning tasks should be processed in high-end servers but created models can be deployed, configured, modified and tuned in switches.

Recent advances in network programmability facilitate the programming of specific tasks at the data-plane level. P4 [Bos14] is widely used today for many tasks including firewalling [Dat18] or bandwidth management [Che19]. P4 is generally agnostic to specific hardware. It is based on the RMT (Reconfigurable Match Table) architectural model [Bos13] that is generally accepted to be generic enough to represent limited but essential switch architecture components and functionalities. The RMT model allows reconfiguring match-action tables where actions can be usual ones (rewrite some headers, forward, drop...). Actions are thus applied on the packets when they are forwarded. Actions can also be more complex programs with some safeguards: no loops, resistivity, etc. The impact on the program development is huge. For example, real number operations are not available by default while they are widely used in many AI algorithms.

In a nutshell, the first challenge to overcome in embedding AI in a network is the hardware's capacity to support AI operations (architectural limitation). Considering software equipment such as a virtual switch simplifies the problem but does not entirely resolve it, as even in that case, strict line-rate requirements limit the type of programs that can be executed. For example, BPF (Berkeley Packet Filter) [McC93] programs provide greater control over packet processing in OVS [Cha18] but still have some limitations, as the execution time of these programs is bounded by nature to ensure their termination, an essential requirement assuming the run-to-completion model which permits high throughput.

The second challenge of network-embedded AI in the network is allocating enough resources for AI tasks while minimizing impact on other network device tasks such as forwarding, monitoring, and filtering. Approximation and/or optimization of AI tasks are potential directions to help in this area. For instance, many network monitoring proposals rely on sketches with well-tuned implementations for the data-plane [Liu16][Yan18]. However, no general optimized AI-programmable abstraction exists to fit all cases, and proposals are mostly use-case centric. There have been many proposed approaches to develop specific P4 programs for many NM tasks, including involving ML. For each, this requires a specific adaptation [Hau23] with a few attempts to propose generic programs to be reusable or composed as a kind of libraries such as [Zha24] leveraging quantization, [Jos21][Jos22] relying on pre-computed lookup tables of real-value functions or [Swa23] proposing function templates for common operations. Besides, distributed processing is a common technique to distribute the load of a single task between multiple entities. AI task decomposition between network elements, edge servers, or controllers has also been proposed [Gup18].

5.4. Distributed AI

Distributed AI assumes different related tasks and components to be distributed across computational and possibly heterogeneous resources. For example, with advances in transfer and Federated Learning (FL), models can be learned, partially shared and combined or data can be also shared to either improve a local or global model. By nature, a network and a networked system is distributed and is thus well adapted to any distributed application. This is exacerbated with the deployment of fog infrastructure mixing network and computational resources. Hence, network management can directly benefit to the distributed network structure to solve its own particular problems, but any other type of AI-based distributed applications also assumes communication technologies to enable interactions between the different entities. This leads to the two sub-challenges described hereafter.

5.4.1. Network management for efficient distributed AI

Distributed AI relies on exchanging information between different entities and comes with various requirements in terms of volume, frequency, security, etc. This can be mapped to network requirements such as latency, bandwidth or confidentiality. Therefore, the network needs to provide adequate resources to support the proper execution of the AI distributed application. While this is true for any distributed application, the nature of the problem intended to be solved by an AI application and how this would be solved should be considered [Lin21]. For example, in the context of optimizing FL [Che22], local models can be shared to create a global model. In case of failure of network links or too high latency, some local models might not be appropriately integrated into the global model with possible impact on AI performance. Depending on the nature of the latter, it might be better to guarantee high-performance communications with a few nodes or to ensure connectivity between all of them even with lower network performance. Mapping is thus necessary between the network management plane and the distributed AI applications which leads to a set of questions to be addressed about interfaces, data and information models or protocols. While the network can be adapted or eventually adapt itself to the AI distributed applications, AI applications could also adapt themselves to the underlying network conditions [Raj24]. It paves the way to research on methods to support AI-aware NM or network-aware AI applications or a combination of both.

5.4.2. Distributed AI for network management

For network management applications relying on distributed AI, challenges from Section 5.4.1 are still valid. Furthermore, network management problems also consider network-specific elements like traffic to be analyzed or configuration to be set on distributed network equipment. Co-locating AI processing and these elements (fully or partially) may help increase performance. For example, pre-calculation on traffic data can be offloaded on network routers before being further processed in high-end servers in a data-center. In general, distributed AI-based network management decisions could be made at different nodes in the network based on locally available information [Sch21]. Hence, deployment of AI-based solutions for network management can also consider various network attributes like network topology, routing policies, or network device capability. In that case, management of computational and network resources is even more coupled than in Section 5.4.1, since the network is both part of the AI pipeline resources and the managed object through AI.

A primary application for distributed AI is for management problems that have a local scope. One example concerns problems that can be addressed at the edge, involving tasks and control loops that monitor and apply local optimizations to the edge in isolation from activities conducted by other instances across the network. However, distributed AI can involve techniques in which multiple entities collaborate to solve a global problem. Such solutions lend themselves to problems in which centralized solutions face certain foundational challenges such as security, privacy, and trust: The need to maintain complete state in a centralized solution may not be practical in some cases due to concerns such as privacy and trust among multiple subdomains, which may not want to share all their data even if they would be willing to collaborate on a problem). Other foundational challenges concern issues related to timeliness, in which distributed solutions may have inherent advantages over centralized solutions as they avoid delays caused by the need to communicate updates globally and across long distances.

5.5. AI for planning of actions

Many tasks in network management revolve around the planning of actions with the purpose of optimizing a network and facilitating the delivery of communication services. For example, communication paths need to be planned and set up in ways that minimize wasted network resources (to optimize cost) while facilitating high network utilization (avoiding bottlenecks and the formation of congestion hotspots) and ensuring resiliency (by making sure that backup paths are not congruent with primary paths). Other examples were mentioned in Section 3.

The promise of central control is that decisions can be optimized when made with complete knowledge of relevant context, as opposed to distributed control, which needs to rely on local decisions made with incomplete knowledge while incurring higher overhead to replicate relevant state across multiple systems. However, as the scale of networks and interconnected systems continues to grow, so does the size of the planning task. Many problems are NP-hard. As a result, solutions typically need to rely on heuristics and algorithms that often result in suboptimal outcomes and that are challenging to deploy in a scalable manner [Ahm21].

The emergence of Intent-Based Networking (IBN) emphasizes the need for automated planning even further. IBN should allow users (network operators, not end users of communication services) to articulate desired outcomes without the need to specify how to achieve those outcomes. An Intent-Based System is responsible for translating the intent into courses of action that achieve the desired outcomes and that continue to maintain the outcomes over time [RFC9315]. How the necessary courses of action are derived and what planning needs to take place is left open, but that is where the real challenge lies. Solutions that rely on clever algorithms devised by human developers face the same challenges as any other network management tasks.

These properties (problems with a clearly defined need, whose solution is faced with exploding search spaces and that today rely on algorithms and heuristics that in many cases result only suboptimal outcomes and significant limitations in scale) make automated planning of actions an ideal candidate for the application of AI-based solutions [Abd24].

A much-publicized leap in AI has been the development of AlphaGo [Sil16]. Instead of using AI to merely solve classification problems, AlphaGo has been successful in automatically deriving winning strategies for board games, specifically the game of Go which features a prohibitively large search space that was long thought to put the ability to play Go at a world class level beyond the reach of problems that AI could solve. Among the remarkable aspects of Alpha Go is that it is able to identify winning strategies completely on its own, without needing those strategies to be taught or learned through observations assuming the system is aware of rules.

The challenge for AI in network management is hence, where is the equivalent of an Alpha Go that can be applied to network management (and networking) problems? Specifically, better solutions are needed for solutions that automatically derive plans and courses of actions for network optimization and similar NP-hard problems, such as provided today with only limited effectiveness by controllers and management applications.

Although AI-based solutions for the automated planning of actions, including the automated identification of courses of action, have not been explored as much as classification [Sal20] problems, the quest for autonomous networks in the last decade and the advent of 5G and B5G have led to a rapid increase in proposed solutions, such as within the context of Zero Touch Management [Cor22].

Also, the evaluation of AI algorithms to derive courses of actions is complex. Contrary to game playing, solutions need to be applied in the real world, where actions have real effects and consequences. Different orientations can be envisioned. First, incremental application of AI decisions with small steps can allow careful observation and detection of unexpected effects. This can be complemented with rollback techniques. Second, verification techniques can be leveraged to verify that decisions made by AI are maintained within safety bounds [Xin24]. Third, sandbox environments can be used but they should be representative of the real world. After progress in simulation and emulation, recent research advances lead to the definition of digital twins, which implies tight coupling between a real system and its digital twin to ensure parallel but synchronized execution [Wu21]. Alternatively, transfer learning techniques are another promising area to capitalize on ML models applicable to real world systems, for instance, to learn an intrusion detector that can be instantiated in multiple environments [Sha24][Ans22]. Generally, it is also an open problem to make the use of AI more acceptable, as highlighted in the dedicated section.

6. Network data as input for ML algorithms

Many applications of AI take data as input. The quality of the outputs of ML-based techniques is highly dependent on the quality and quantity of data used for learning and also during inference. For example, as modern network infrastructures move toward higher speed and scale, they aim to support increasingly more demanding services with strict performance guarantees. These often require resource reconfigurations at runtime in response to emerging network events so that they can ensure reliable delivery at the expected performance level. Timely observation and detection of events is also of paramount importance for security purposes and can allow faster execution of remedy actions, thus leading to reduced service downtime.

Thus, the challenge of data management is multifaceted, as detailed in the next subsections.

6.1. Data for AI-based NM solutions

Assuming a network management application, the first problem to address is to define the data to be collected, which will be appropriate to obtain accurate results. This data selection can require defining problem-specific data or features (feature engineering). Furthermore, machine learning algorithms only work as desired when data to be analyzed respects given properties. Many methods rely on vector-based distances, which assume that the data encoded into the vector respects the underlying distance semantic. Taking the first n bytes of a packet as vectors and computing distances accordingly is possible, but does not embed the semantic of the information carried in the headers. A solution to circumvent the problem of network traffic representation is to transform traffic data into a specific format that can be easily handled by NN architectures, for instance by creating images analyzed by convolutional neural networks [Sha21]. Data can also be easily represented as graphs, because topologies or communication graphs would require less adaptation to be given as inputs to GNNs [Jie22]. As an example, graph-based representations are considered practically efficient due to their intrinsic structural similarities with network data, as illustrated in [Bar23] for attack detection.

Since data to handle can also be in a schema-free or eventually text-based format, one example could be the automated annotation of management intents provided in unstructured textual format (policies descriptions, specifications) to extract management entities and operations. For that purpose, suitable annotation models need to be built using existing NER (Named Entity Recognition) techniques usually applied for NLP. However, this shall be carefully crafted or specialized for network management (intent) language which indirectly relates back to the challenges of AI techniques for NM specified earlier. Today, with the progress on LLMs, different proposed approaches have emerged as for example in [Mek24][Fua24][Dze23]. Secondly, similar to the problem of mapping AI algorithms with NM problems in section Section 5.1, data to be collected also depends on the NM problem to be solved. The mapping between the data sources and the problem is not straightforward as all dependencies or correlations are not known in advance and some might be discovered by the AI algorithms themselves [For23]. In addition, the types of data to collect can vary over time to maintain the performance of an AI-based application or to adapt it to a new context when learned models are updated dynamically.

Thirdly, the behaviour of any network is not just derived from the events that can be directly observed, such as network traffic overload, but also from events occurring outside the environment of the network. The information provided by the detectors of such kinds

of events, e.g. a natural incident (earthquake, storm), can be used to determine the adaptation of the network to avoid potential problems derived from such events. Those can be provided by big data sources as well as sensors of many kinds. The challenge related to this task is to process large amounts of data and associate it with the effects that those events have on the network. It is hard to determine the static and dynamic relation between the data provided by external sources and the specific implications it may have in networks. For instance, the effect of a flash crowd detected in an external source may require adapting the network service configuration to support is related workload. The objective is to complement a control-loop, as shown in [Mar18], by including the specific AI engines into the decision components as well as the processes that close the loop, so the AI engine can receive feedback from the network in order to improve its own behaviour.

6.2. Data collection

Once defined, the second problem to address is the collection of data. Monitoring frameworks have been developed for many years such as IPFIX [RFC7011] and more recently with SDN-based monitoring solutions [Yul4][Ngu20]. However, going towards more AI for actions in network management supposes also to retrieve more than traffic related information. Actually, configuration information such as topologies, routing tables or security policies have been proven to be relevant in specific scenarios. As a result, many different technologies can be used to retrieve meaningful data. To support improved QoE, monitoring of the application layer is helpful but far from being easy with the heterogeneity of end-user applications and the wide use of encrypted channels. Monitoring techniques need to be reinvented through the definition of new techniques to extract knowledge from raw measurement [Bril9] or by involving end-users with crowd-sourcing [Hir15] and distributed monitoring. Also, the data-mesh concept [Mac22] proposes to classify data into three categories: source-aligned, aggregate and consumer-aligned. Source-aligned data are those related to the same operational domain, and it is important to correlate or aggregate them with higher planes: management-, control- and forwarding plane. An issue is the difference, not only in the nature of data, but in their volumes and their variety. Some may change rapidly over time (for example network traffic) while other may be quite stable (device state).

The collecting process requirements depend on the kind of processing. We can distinguish two major classes: batch/offline vs real-time/online processing. In particular, real-time monitoring tools are key in enabling dynamic resource management functions to operate on short reconfiguration cycles. However, maintaining an accurate view of the network state requires a vast amount of information to be collected

and processed. While efficient mechanisms that extract raw measurement data at line rate have been recently developed, the processing of collected data is still a costly operation. This involves potentially sampling, evaluating and aggregating a vast amount of state information as a response to a diverse set of monitoring queries, before generating accurate reports. One difficult problem resides also in the availability of data as real-time data from different sources to be aggregated may not arrive at the same time requiring so some buffering techniques. Machine learning methods, e.g. based on regression, can be used to intelligently filter the raw measurements and thus reduce the volume of data to process. For example, in [Tan20] the authors proposed an approach in which the classifiers derived for this purpose (according to measurements on traffic properties) can achieve a threefold improvement in the query processing capability. A residual question is the storage of raw measurements. In fact, predicting the lifetime of data is challenging because their analysis may not be planned and triggered by a particular event (for example, an anomaly or attack). As a result, the provisioning of storage capacity can be hard.

In parallel to the continuously increasing dynamic behavior of networks and complexity of traffic, there is a trend towards more user traffic processing customization [RFC8986][Li19]. As a result, fine grained information about network element states is expected and new proposed approaches have emerged to collect on-path data or in-band network telemetry information [Tan20b]. These new approaches have been designed by introducing much flexibility and customization and could be helpful to be used in conjunction with AI applications. However, the seamless coupling of telemetry processes with packet forwarding requires careful definition of solutions to limit the overhead and the impact of the throughput while providing the necessary level of details. This shares commonalities with the lightweight AI challenge.

6.3. Usable data

Although all agree on the necessity to have more shared datasets, it is quite uncommon in practice. Data contains private or sensitive information and may not be shared because of the criticality of data (which can be used by ill-intentioned adversaries) or due to laws or regulations, even within the same company. To solve this issue, anonymization techniques [Dij19] can be enhanced to optimize the trade-off between valuable data and sensitive information (potential) leakage or reconstruction. Whatever the final user of data, regulations and laws impose rules on data management with potentially costly impact if they are not respected voluntarily or not. Defining a new monitoring framework should always consider security and privacy aspects, for example to let any user/customer or access/

remove its own data with General Data Protection Regulation (GDPR) in EU. The challenge resides here in the capacity of qualifying what is critical or private information and the capacity for an adversary to reconstruct it from other sources of data. Hence AI/ML based solutions will require more data but also more administrative, legal and ethical procedures. Those can last long and so slow down the deployment of a new solution. In addition, this requires interaction with experts from different domains (e.g. AI engineer and a lawyer) for example to ensure by-design privacy in traffic analysis techniques [Wan22]. The integration of these non-technical constraints should be considered when defining new data to be collected or a new technique to collect data. However, knowing the final use of data is most of the time necessary for ethical and legal assessment which assumes that those considerations should be integrated from the early design of new AI-based solutions.

For supervised or semi-supervised training, having a labelled dataset is a prerequisite. It constitutes a major challenge as well. While probes exist to collect real network data, those data are typically unlabelled. This limits application of ML to unsupervised learning tasks (learning from data). Because manual labelling is a tedious task, one option is to leverage AI to guide humans. This may also support a better generalization of a learned model. Indeed, an underlying challenge is the genericity or coverage of the datasets. Labels encode values of an objective function, the challenge posed by the design of such tools is tremendous since for involving a M:N relationship: 1 data type may be associated to M objective function values and N data types may be associated to 1 objective function. As a result, most datasets used for research encode a single label for a particular application like attack label for datasets to be used in the context of intrusion detection or application type for network traffic used for classification where the value of a single dataset could be capitalized in several applications. More generally, in the context of intrusion detection, using raw data rather than pre-processed data as it is common in open dataset has been demonstrated to be inefficient [Dub24].

Again, researchers need empirical (or at least realistic) datasets to validate their solutions. Unfortunately, as highlighted above, having such data from real deployments for various reasons (business secrets, privacy concerns, concerns that vulnerabilities are revealed by accident, raw unlabelled data, etc.) is tough. Even if such a dataset is available it might not be enough to convincingly validate a new algorithm. Instead of falling back to artificial testbed experiments or simulation, it would be useful to have the capability to generate datasets with characteristics that are not 100% identical but like the characteristics of one or more real datasets. Such synthetic networks can be used to validate new management algorithms,

intrusion detection systems, etc. The usage of AI, Generative Adversarial Networks (GANs), in this area [Hui22] is not yet widespread and there are still many concerns that deter researchers, e.g. the fear of leaking sensitive information from the original dataset into the synthetic dataset. Furthermore, a major underlying challenge is to generate data realistic enough. This requires formalizing and integrating network-specific constraints, such as protocol specification, when generating data [Jia24] or using a real testbed guided by AI to create challenging scenarios for validation purposes [Ans25].

7. Acceptability of AI

Networks are critical infrastructures. On one hand, they should be operated without interruption and must be interoperable. Networks, except in a lab, are not isolated, which slow down innovation in general. For example, changing Internet routing protocols need to be accepted by multiple entities such as operators of interconnected networks. The same applies for protocols. Even if there have been several versions of major protocols in use like TCP or DNS, there are still some security issues which cannot be patched with 100% guarantee. On the other hand, results provided by AI solutions are uncertain by nature. The same technique applied in different environments can produce different results. AI techniques need some effort (time and human) to be properly configured or to be stabilized. For instance, RL needs several iterations before being able to produce acceptable results. These properties of AI techniques are thus a bit antagonist with the criticality of network infrastructures. With that in mind, acceptability of AI by network operators is clearly an obstacle for its larger adoption.

7.1. Explainability of Network-AI products

A common issue across many ML applications is their lack to provide human-understandable reasoning processes. This means that, after training, the knowledge acquired by ML models is unintelligible to humans. As a result, offering hard guarantees on performance is a very challenging issue. In addition, complex ML models like neural networks -that often have more than hundreds of thousands of parameters- are very hard to debug or troubleshoot in case of failure.

While this is a common issue for all applications of AI, many areas work well with uncertainty and the black-box behaviour of AI-based solutions. For instance, users accept an inherent error in recommender systems or computer vision solutions.

The networking field has already produced a set of well-established network management algorithms and methods, with clear performance guarantees and troubleshooting mechanisms [Rex06][Kr14]. As such, improving debugging, troubleshooting and guarantees on AI-based solutions for networking is a must. AI researchers and practitioners are devoting large research efforts to improve this aspect of ML models, which is commonly known as explainability [XAI].

This set of techniques provides insights and, in some cases, guarantees on the performance and behaviour of ML-based solutions. Understanding such techniques, researching and applying them to network AI is critical for the success of the field.

There exist several ML-based methods that are human-understandable, although not widely used today. For instance, [Mar20] shows a method for building anticipation models (prediction) that provide explanations while determining some actions for tuning some parameters of the network. There are other challenges that should be addressed, such as providing explanations for other ML methods that are quite extended. For instance, xNN/SVM models can be accompanied by Digital Twins of the network that are reversely explored to explain some output from the ML model (e.g., xNN/SVM). In this context, there already exist several methods [Zil20][Puj21] that produce human-readable interpretations of trained NN models, by analysing their neural activations on different inputs. It is worth noting that Digital Twins are not considered per se an AI approach; they merely serve to provide a digital representation of a network that can serve as its proxy and offer a layer of indirection between management applications and actual network resources. However, it is conceivable that AI-based management applications can be combined and operate in conjunction with Digital Twin technology, for example to use a Digital Twins as an experimentation sandbox or staging ground for AI-driven applications.)

7.2. AI-based products and algorithms in production systems

AI-based network management and optimization algorithms are first trained, then the resulting model is used to produce relevant inferences in operation, either in management or optimization scenarios. A relevant question for the success of AI-based solutions is: where does this training occur?

Traditionally, AI-based models have been trained in the same scenario where they operate [Val17][Xu18], this is the customer network. However, this presents critical drawbacks. First, training an AI model for management and operation typically requires generating network configurations and scenarios that can break the network. This is because training requires seeing a broad spectrum of

scenarios. Thus, training in production networks is very challenging. Second, customer networks may not be equipped with the monitoring infrastructure required to collect the data used in the training process (e.g., performance metrics). Performing learning directly into a production network is possible assuming imperfect models and the need of several step of refinement before it gets stable. For non-critical management task, such assumption can hold, and additional safe-guarding mechanisms should be considered in order to keep outputs of ML algorithms (such as decisions) within acceptable boundaries.

A more sensible approach is to train the AI-based product in a lab, for instance in the vendor's premises. In the lab, AI models can be trained in a controlled testbed, with any configuration, even ones that break the network. However, the main challenge here arises from the fundamental differences between the lab's network and the customer networks. For instance, the topology of the lab's network might be smaller, etc. As a result, there is a need for the learned models to generalize. In this context, generalization means that models should be able to operate in other scenarios not seen during training, with different topologies, routing configurations, scheduling policies, etc. This well-known problem is due to the dependency between training and testing data when benchmarking models.

In order to address this generalization problem, multiple complementary approaches are possible:

One approach is training on diverse data that represents large parts of the expected problem space. For example, training with various traffic patterns will help improving the generalization of an AI-based traffic analyser. Another approach is to leverage AI designs or architectures that facilitate generalization. One example is GNNs [gnn1][gnn2]. They are a type of neural networks able to operate and generalize over graphs. Indeed, networks are fundamentally represented as graphs: topology, routing, etc. With GNNs, vendors can train the AI model in a lab with a certain topology and then directly use the resulting model in different customer networks, even with different network topologies. Finally, another approach is Transfer Learning [tl1]. With this technique, the knowledge gained in the lab's training is used to operate in the customer network. Transfer Learning still requires that some data from the customer is used to re-train and fine-tune the model (e.g., accurate performance measurements). This means that, for each customer network, re-training is required. This may be problematic since it requires added cost and access to customer data.

In addition to the challenge of generalizing from training to production environment, there are also challenges in terms of interoperability between different AI approaches and different deployment environments. As mentioned above, AI approaches may be deployed in diverse environments, e.g., for training and production, but also for local development, for testing, and for validation or in different part of the production systems. These environments may differ in available compute resources, network topology, operating systems, cloud providers, etc. (single node machine, single cluster, many distributed clusters, etc.). Deploying the same AI solutions in these different environments can lead to various challenges in terms of interoperability. Common AI frameworks support scaling across networks of different size. Yet, many frameworks are often combined, e.g., for data collection, processing, predictions, validation, etc. Again, ensuring interoperability between these frameworks can be tedious.

This shares similarities with problems described in Section 5.4 and particularly emphasizes the need for network environments to provide interfaces and descriptions suitable for AI solutions to be properly instantiated and configured.

One approach to address these interoperability challenges is through meta-frameworks that interface with most available AI frameworks. These meta-frameworks provide a higher level of abstraction and often allow seamless deployment across different environments (e.g., on-premises or at different cloud providers) [Mor18].

7.3. AI with humans in the loop

Depending on the network management task, AI can automate and replace manual human control, or it can complement human experts and keep them in the loop. Keeping humans in the loop will be an important step of building trust in AI approaches and help ensure the desired outcomes. There are various ways of keeping humans in the loop in the different fields of AI [Wu22], which could be useful for different aspects of network management.

In classification tasks (e.g., detecting security breaches, malware or detecting anomalies), trained AI models provide a confidence score in addition to the predicted class. If the confidence is high, the prediction is used directly. If the confidence is too low, a human expert may jump in and make the decision – thereby also providing valuable training data to improve the AI model. Such approaches are already being used in industry, e.g., to automatically label datasets (AWS SageMaker). Similar approaches could also be used for other supervised learning tasks, e.g., regression. Still, it is an open challenge to keep humans in the loop in all phases of the learning process.

When using RL, e.g. to control service scaling and placement or route traffic flows, the agents typically interact with the environment (i.e., the simulated or real network) completely autonomously without human feedback. However, there is a growing number of approaches to put human experts back into the loop. One approach is offline reinforcement learning, where the training data does not come from the reinforcement learning agent's own exploration but from pre-recorded traces of human experts (e.g., placement decisions that were made by humans before). Another approach is to reward the RL agent based on human feedback rather than a pre-defined reward function [Lee21]. Again, while there are first promising approaches, more work is required in this area. Overall, it is an open challenge to both leverage the benefits of AI but keep human experts in the loop where it is useful.

8. Conclusions and Future Research Directions

This document has established a six-characteristic taxonomy to map the inherent architectural challenges found at the intersection of Network Management and Artificial Intelligence. To prevent this research from remaining siloed in standalone academic configurations, the NMRG and the broader NM community must focus technical resources on resolving the friction points where abstract AI/ML models meet deterministic network operations.

Key takeaways include:

- * NM problems often require network-specific solutions rather than direct imports from other AI domains.
- * Data quality, representation, and privacy-aware sharing are fundamental enablers for effective AI-based NM.
- * Lightweight and distributed AI architectures are needed to meet timing, resource, and deployment constraints.

- * Seamless integration with existing NM processes, explainability, and human-in-the-loop interaction are essential for acceptability.
- * Practical deployment also requires evaluation methods, safeguards, and mechanisms for models to generalize and adapt over time.

9. Security considerations

This document introduces the challenges of coupling AI and NM. Since the aim of this document is not to address a particular NM problem by defining a solution and because many possible ones can be developed further, it is not possible at this stage to define security concerns specific to a solution. However, examples of applications mentioned and cited in the different sections may face their own security concerns. In this section, our objective is to highlight high-level security considerations to be considered when coupling AI and NM. Those concerns serve as the common basis to be refined according when a particular NM application is developed.

9.1. AI-based security solutions

The first security consideration refers to the use of AI for NM problem related to security of the managed networked systems. There are multiple scenarios where AI can be leveraged: to perform traffic filtering, to detect anomalies or to decide on target moving defence strategies. In these cases, the performance of the AI algorithms impacts on the security performance (e.g. detection or mitigation effectiveness) like any other non-AI system. However, AI methods generally tend to obfuscate how predictions are made and decisions taken. Explainability of AI is thus highly important and is addressed globally in section Section 7.1.

Assuming a ML trained model, there is always an uncertainty regarding reachable performance on the wild once the solution is deployed as it can suffer from a poor generalization due to different reasons. There are two major problems which are well known in the ML field: overfitting or under-fitting. In the first case, the learned model is too specific to the training data while in the second case the model does not infer any valuable knowledge from data. To avoid these issues, hyper-parameter fine-tuning is necessary. For example, the number of iterations is an essential hyper-parameter to be adjusted to learn a neural network model. If it is too low, the learning does not converge to a representative model of the training data (underfitting). With a high value, there is a risk that the model is too close to the training data (overfitting). In general, finding the right hyper-parameters is helpful to find a good ML algorithm configuration. There are different techniques ranging from grid-search to Bayesian optimization falling into the AI area of

Hyper-Parameter Optimization (HPO) [Bis23]. This consideration goes beyond the sole problem of hyper-parameters settings but as a full analysis pipeline also assumes the ML algorithm to be selected or the data pre-processing to be configured. This reflects challenges from the AI research area covered by AutoML technique. In this document, this is also referred in section Section 5.1.3 when considered in the context of NM. As highlighted in the aforementioned section, some expertise or area-specific knowledge can help guiding automated configuration processes.

Besides, machine learning assumes to have representative training data. The quality of dataset for learning is a vast problem. Additionally, the representation of data needs to be addressed carefully to be properly analysed by AI models, for example with pre-processing techniques to normalize data, balance classes or encode categorical features depending on the type of algorithms. Actually, section Section 6 of this document fully addresses the concerns related to data in regard to NM problems.

9.2. Security of AI

Although ensuring a good performance of AI algorithms is already challenging, assuming an attacker aiming at compromising it emphasizes the problem. Adversarial AI and notably adversarial ML have attracted a lot of attention over the last year. Adversarial AI and ML relates to both attack and defences. While this is out of scope of the document, evaluating threats against an ML system before deploying it is an important aspect. This supposes to assess what types of information the attacker can access (training data, trained model, algorithm configuration...) and the performable malicious actions (inject false training data, test the system, poison a model, etc.) to evaluate the magnitude of the impact of possible attacks.

For illustration purposes, we refer hereafter to some examples. In the case of an intrusion detector, an attacker may try to poison training data by providing adversarial samples to ensure that the detector will miss-classify the future attacks [Jmi22]. In a white-box approach where the model is known from the attacker, the attack can be carefully crafted to avoid being properly labelled. For instance, packet sizes and timings can be easily modified to bypass ML-based traffic classification system [Nas21]. In a black-box model, the attacker ignores the functioning or training data of the ML systems but can try to infer some information. For example, the attacker can try to reconstruct sensitive information which have been used for training. This type of attack qualified as model inversion [Fre15] raises concern regarding privacy.

All these threats are exacerbated in the context of solutions relying on distributed AI involving multiple entities that are not necessarily controlled by the same authority. Once the threats are assessed, solutions need to be developed and deployed which can be either proactive by providing some guarantees regarding the involved entities using authentication or trust mechanisms but also reactive by validating data processing through voting mechanisms or knowledge proofs. Other solutions include defensive techniques to rate limit or filter queries to a particular deployed model. All these examples are for illustration purposes and are not exhaustive.

9.3. Relevance of AI-based outputs

Security breaches can be created by an AI-driven application. Generally, any system that will be used to guide or advice on actions to be perform on network raise the same issue. For example, if an AI algorithm decides to change the filtering tables in a network it may compromise access control policies. Irrelevant results could be also produced. In the area of QoS, an AI system could allocate a bandwidth to a flow higher to the real link capacity. As shown from these two examples, an AI can produce decisions or values which are out of bounds of normal operations. To avoid such issues, safeguards can be added to discard or correct irrelevant outputs. Detecting such type of outputs can be also challenging in complex and distributed systems such as a network. Formal verification methods or testing techniques are helpful in that context.

10. IANA Considerations

This document has no IANA actions.

11. References

11.1. Normative References

- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

- [RFC9315] Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", RFC 9315, DOI 10.17487/RFC9315, October 2022, <<https://www.rfc-editor.org/info/rfc9315>>.

11.2. Informative References

- [Abd10] Jalil, K. A., Kamarudin, M. H., and M. N. Masrek, "A Diagnosis Expert System for Network Traffic Management", 2010. IEEE international conference on networking and information technology
- [Abd24] Bouroudi, A., Outtagarts, A., and Y. Hadjadj-Aoul, "A dynamic AI-based algorithm selection for Virtual Network Embedding", 2024. Springer Annals of Telecommunication
- [Ahm21] Ahmad, S. and A. H. Mir, "Scalability, Consistency, Reliability and Security in SDN Controllers: A Survey of Diverse SDN Controllers", 2021. Springer Journal of Network and Systems Management, 29(2)
- [Ake24] Akem, A. T. J., Fraysse, G., and M. Fiore, "Encrypted Traffic Classification at Line Rate in Programmable Switches with Machine Learning", 2024. IEEE/IFIP Network Operations and Management Symposium (NOMS)
- [Ami24] Amin, R. M., Hammer, P., and A. Butz, "Using Machine Learning to Improve Interactive Visualizations for Large Collected Traffic Detector Data", 2024. Proceedings of the 29th International Conference on Intelligent User Interfaces (ITUI'24), ACM
- [Ans22] Anser, O., FranÃ§ois, J., and I. Chrisment, "Auto-tuning of Hyper-parameters for Detecting Network Intrusions via Meta-learning", 2022. IEEE/IFIP Network Operations and Management Symposium (NOMS) - AnNet workshop
- [Ans25] Anser, O., FranÃ§ois, J., Chrisment, I., and D. Kondo, "TATA: Benchmark NIDS Test Sets Assessment and Targeted Augmentation", 2025. ESORICS (European Symposium on Research in Computer Security)
- [Ans26] Anser, O., FranÃ§ois, J., and I. Chrisment, "An Empirical Study on Configuration Robustness of Unsupervised IDS", 2026. IEEE/IFIP Network Operations and Management Symposium (NOMS)

- [Arz21] Arzani, B., Hsieh, K., and H. Chen, "Interpretable Feedback for AutoML and a Proposal for Domain-customized AutoML for Networking", 2021. ACM Workshop on Hot Topics in Networks (HotNets)
- [Att23] Attaoui, W., Sabir, E., Elbiaze, H., and M. Guizani, "VNF and CNF Placement in 5G: Recent Advances and Future Trends", 2023. IEEE Transactions on Network and Service Management, 20(4), pp. 4698-4733
- [Bar23] Barsellotti, L., Marinis, L. D., Cugini, F., and F. Paolucci, "FTG-Net: Hierarchical Flow-to-Traffic Graph Neural Network for DDoS Attack Detection", 2023. IEEE International Conference on High Performance Switching and Routing (HPSR)
- [Beg19] Bega, D., Gramaglia, M., Fiore, M., Banchs, A., and X. Costa-Perez, "DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning", 2019. IEEE INFOCOM
- [Bis23] Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D., and M. Lindauer, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges", 2023. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 13(2)
- [Bos13] Bosshart, P., Gibb, G., Kim, H.-S., Varghese, G., McKeown, N., Izzard, M., Mujica, F., and M. Horowitz, "Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN", 2013. ACM SIGCOMM
- [Bos14] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., and D. Walker, "P4: Programming Protocol-Independent Packet Processors", 2014. SIGCOMM Comput. Commun. Rev. 44
- [Bou18] Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities", 2018. Journal of Internet Services and Applications 9, 16

- [Bril19] Brissaud, P.-O., François, J., Chrisment, I., Cholez, T., and O. Bettan, "Transparent and Service-Agnostic Monitoring of Encrypted Web Traffic", 2019. IEEE Transactions on Network and Service Management, 16 (3)
- [Chal18] Chaignon, P., Lazri, K., François, J., Delmas, T., and O. Festor, "Oko: Extending Open vSwitch with Stateful Filters", 2018. ACM Symposium on SDN Research (SOSR)
- [Che19] Chen, Y., Yen, L., Wang, W., Chuang, C., Liu, Y., and C. Tseng, "P4-Enabled Bandwidth Management", 2019. Asia-Pacific Network Operations and Management Symposium (APNOMS)
- [Che22] Chen, H., Huang, S., Zhang, D., Xiao, M., Skoglund, M., and H. V. Poor, "Federated Learning Over Wireless IoT Networks With Optimized Communication and Resources", 2022. IEEE Internet of Things Journal, 9(17)
- [Col22] Collet, A., Banchs, A., and M. Fiore, "Using Machine Learning to Improve Interactive Visualizations for Large Collected Traffic Detector Data", 2022. IEEE Conference on Computer Communications (Infocom)
- [Cor22] Coronado, E., Behraves, R., Subramanya, T., Fernández-Fernández, A., Siddiqui, M. S., and X. Costa-Pérez, "Zero Touch Management: A Survey of Network Automation Solutions for 5G and 6G Networks", 2022. IEEE Communications Surveys & Tutorials, 24(4)
- [czb20] Clemm, A., Zhani, M. F., and R. Boutaba, "Network Management 2030: Operations and Control of Network 2030 Services", 2020. Springer Journal of Network and Systems Management (JNSM)
- [Das23] Dasari, A.K., Biswas, S. K., Thounaojam, D. M., Devi, D., and B. Purkayastha, "Ensemble Learning Techniques and Their Applications: An Overview", 2023. Advances in Cognitive Science and Communications
- [Dat18] Datta, R., Choi, S., Chowdhary, A., and Y. Park,, "P4Guard: Designing P4 Based Firewall", 2018. IEEE Military Communications Conference (MILCOM)
- [Dij19] Dijkhuizen, N. V., Ham, J. V. D., and X. Li, "A Survey of Network Traffic Anonymisation Techniques and Implementations", 2014. ACM Comput. Surv. 51, 3, Article 52

- [Dub24] Dube, R., "Faulty use of the CIC-IDS 2017 dataset in information security research", 2024. Journal of Computer Virology and Hacking Techniques
- [Dze23] Dzeperoska, K., Lin, J., Tizghadam, A., and A. Leon-Garcia,, "LLM-Based Policy Generation for Intent-Based Management of Applications", 2023. International Conference on Network and Service Management (CNSM)
- [Evr19] Evrard, L., FranÃ§ois, J., Colin, J.-N., and F. Beck, "port2dist: Semantic Port Distances for Network Analytics", 2019. IFIP/IEEE Symposium on Integrated Network and Service Management (IM)
- [For23] Foroughi, P., Brockners, F., and J.-L. Rougier, "ADT: AI-Driven network Telemetry processing on routers", 2023. Elsevier Computer Networks (220)
- [Fre15] Fredrikson, M., Jha, M. S., and T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures", 2015. ACM SIGSAC Conference on Computer and Communications Security
- [Fua24] Fuad, A., Ahmed, A. H., Riegler, M. A., and T. An, "Intent-based Networks Framework based on Large Language Models", 2024. IEEE International Conference on Network Softwarization (NetSoft)
- [Gin24] Ginige, Y., Dahanayaka, T., and S. Seneviratne, "TrafficGPT: An LLM Approach for Open-Set Encrypted Traffic Classification", 2024. Asian Internet Engineering Conference
- [gnn1] Battaglia, P. W. and E. al, "Relational inductive biases, deep learning, and graph networks", 2018. arXiv preprint arXiv:1806.01261
- [gnn2] Rusek, K., SuÃ±ez-Varela, J., Mestres, A., Barlet-Ros, P., and A. Cabellos-Aparicio, "Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN", 2019. ACM Symposium on SDN Research
- [Gre19] Greeshma, K. V. and K. Sreekumar, "Fashion-MNIST classification based on HOG feature descriptor using SVM", 2019. Int. J. Innov. Technol. Explor. Eng. 8, 5 (2019)

- [Gup18] Gupta, A., Harrison, R., Canini, M., Feamster, N., Rexford, J., and W. Willinger, "Sonata: query-driven streaming network telemetry", 2018. ACM SIGCOMM Conference
- [Ham21] Hamdan, M., Hassan, E., Abdelaziz, A., Mohammed, B., Khan, S., Vasilakos, A. V., and M.N. Marsono, "Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey", 2021. Journal of Network and Computer Applications, Elsevier
- [Ham23] Hammi, S., Hammami, S. M., and L.H. Belguith, "Advancing aspect-based sentiment analysis with a novel architecture combining deep learning models CNN and bi-RNN with the machine learning model SVM", 2023. Soc. Netw. Anal. Min., 13(117)
- [Hau23] Hauser, F., HÄmberle, M., Merling, D., Lindner, S., Gurevich, V., Zeiger, F., Frank, R., and M. Menth, "A survey on data plane programming with P4: Fundamentals, advances, and applied research", 2023. Elsevier Journal of Network and Computer Applications (212)
- [Hir15] Hirth, M., Hossfeld, T., Mellia, M., Schwartz, C., and F. Lehrieder, "Crowdsourced network measurements: Benefits and best practices", 2015. Computer Networks. 90
- [Hoa21] Hoang, N. P., Niaki, A. A., Gill, P., and M. Polychronakis, "Domain name encryption is not enough: Privacy leakage via IP-based website fingerprinting", 2021. Proceedings on Privacy Enhancing Technologies
- [Hoo18] Hooft, J. V. D., Claeys, M., Bouten, N., Wauters, T., SchÄnwald, J., Stiller, A. P. B., Charalambides, M., Badonnel, R., Serrat, J., Santos, C. R. P. D., and F. D. Turck, "Updated Taxonomy for the Network and Service Management Research Field", 2018. Journal of Network System Management (JNSM) 26, 790â\200\223808
- [Hua19] Huang, C., Zhai, S., Talbott, W., Bautista, M. A., Sun, S.-Y., Guestrin, C., and J. Susskind, "Addressing the Loss-Metric Mismatch with Adaptive Loss Alignment", 2020. ICRL
- [Hua25] Huang, Y., Du, H., Zhang, X., Niyato, D., Kang, J., and Z. Xiong, "Large Language Models for Networking: Applications, Enabling Techniques, and Challenges", 2025. IEEE Network, 39(1)

- [Hui22] Hui, S., Wang, H., Wang, Z., Yang, X., Liu, Z., Jin, D., and Y. Li, "Knowledge Enhanced GAN for IoT Traffic Generation", 2022. ACM Web Conference 2022 (WWW)
- [Jia24] Jiang, X., Liu, S., Gember-Jacobson, A., Bhagoji, A. N., Schmitt, P., Bronzino, F., and N. Feamster., "NetDiffusion: Network Data Augmentation Through Protocol-Constrained Traffic Generation", 2024. ACM on Measurement and Analysis of Computing Systems
- [Jie22] Jiang, W., "Graph-based deep learning for communication networks: A survey", 2022. Computer Communications, Elsevier, vol. 185
- [Jmi22] Jmila, H. and M. I. Khedher, "Adversarial machine learning for network intrusion detection: A comparative study", 2022. Computer Networks
- [Jos21] Jose, M., Lazri, K., FranÃ§ois, J., and O. Festor, "InREC: In-network REal Number Computation", 2021. IFIP/IEEE International Symposium on Integrated Network Management (IM)
- [Jos22] Jose, M., Lazri, K., FranÃ§ois, J., and O. Festor, "NetREC Network-wide in-network REal-value Computation.", 2022. IEEE International Conference on Network Softwarization (NetSoft)
- [Kaf19] Kafle, V. P., Martinez-Julia, P., and T. Miyazawa, "Automation of 5G Network Slice Control Functions with Machine Learning", 2019. IEEE Communications Standards Magazine, vol. 3, no. 3, pp. 54-62
- [Ke23] Ke, H., Dongseong, K. D., and M. R. Asghar, "Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey", 2023. IEEE Communications Surveys & Tutorials, 25(1), 538-566.
- [Krl14] Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., and S. Uhlig, "Software-defined networking: A comprehensive survey", 2015. Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76
- [Lee21] Lee, K., Smith, L., and P. Abbeel, "Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training", 2021. arXiv preprint arXiv:2106.05091

- [Li19] Li, R., Makhijani, K., Yousefi, H., Westphal, C., Dong, L., Wauters, T., and F. D. Turck., "A Framework for Qualitative Communications Using Big Packet Protocol", 2019. ACM SIGCOMM Workshop on Networking for Emerging Applications and Technologies (NEAT)
- [Lin21] Lin, Z., Bi, S., and Y. -J. A. Zhang, "Optimizing AI Service Placement and Resource Allocation in Mobile Edge Intelligence Systems", 2021. IEEE Transactions on Wireless Communications, 20(11)
- [Liu16] Liu, Z., Manousis, A., Vorsanger, G., Sekar, V., and V. Braverman, "One Sketch to Rule Them All: Rethinking Network Flow Monitoring with UnivMon", 2016. ACM SIGCOMM Conference
- [Liu21] Liu, Q., Choi, N., and T. Han, "Constraint-Aware Deep Reinforcement Learning for End-to-End Resource Orchestration in Mobile Networks", 2021. IEEE International Conference on Network Protocols (ICNP)
- [Liu22] Liu, B., Ding, M., Shaham, S., Rahayu, W., Farokhi, F., and Z. Lin, "When Machine Learning Meets Privacy: A Survey and Outlook", 2022. ACM Comput. Surv., 54(2)
- [Liu22b] Liu, L., Engelen, G., Lynar, T., Essam, D., and W. Joosen, "Error Prevalence in NIDS datasets: A Case Study on CIC-IDS-2017 and CSE-CIC-IDS-2018", 2022. IEEE Conference on Communications and Network Security (CNS)
- [Lop20] L  pez, J., Labonne, M., Poletti, C., and D. Belabed, "Priority Flow Admission and Routing in SDN: Exact and Heuristic Approaches", 2020. IEEE International Symposium on Network Computing and Applications (NCA)
- [Mac22] Machado, I. A., Costa, C., and M. Y. Santos, "Data Mesh: Concepts and Principles of a Paradigm Shift in Data Architectures", 2022. Procedia Computer Science, Volume 196
- [Mar18] Martinez-Julia, P., Kafle, V. P., and H. Harai, "Exploiting External Events for Resource Adaptation in Virtual Computer and Network Systems", 2018. IEEE Transactions on Network and Service Management, Vol. 15, N. 2,

- [Mar20] Martinez-Julia, P., Kafle, V. P., and H. Asaeda, "Explained Intelligent Management Decisions in Virtual Networks and Network Slices", 2020. Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)
- [Mcc93] McCanne, S. and V. Jacobson, "The BSD packet filter: a new architecture for user-level packet capture", 1993. USENIX Winter Conference
- [Mek24] Mekrache, A., Ksentini, A., and C. Verikoukis, "Intent-Based Management of Next-Generation Networks: an LLM-Centric Approach", 2024. IEEE Network, 38(5)
- [Mor18] Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M., and I. Stoica, "Ray: A Distributed Framework for Emerging AI Applications", 2018. USENIX Symposium on Operating Systems Design and Implementation (OSDI)
- [Mus18] Musumeci, F., Rottondi, C., Nag, A., Macaluso, I., Zibar, D., Ruffini, M., and M. Tornatore, "An overview on application of machine learning techniques in optical networks", 2018. IEEE Communications Surveys & Tutorials, 21(2), 1383–1408.
- [Mut24] Muteba, F., Dahanayaka, T., and S. Seneviratne, "Digital twin (DT)-based predictive maintenance of a 6G communication network", 2024. Procedia Computer Science, Elsevier, vol. 238
- [Naj24] Najm, I. A., Saeed, A. H., Ahmad, B.A., Ahmed, S. R., Sekhar, R., Shah, P., and B.S. Veena, "Enhanced Network Traffic Classification with Machine Learning Algorithms", 2024. Proceedings of the Cognitive Models and Artificial Intelligence Conference
- [Nas21] Nasr, M., Bahramali, A., and A. Houmansadr, "Defeating DNN-Based Traffic Analysis Systems in Real-Time With Blind Adversarial Perturbations", 2021. USENIX Security Symposium
- [Ngu20] Nguyen, T. G., Phan, T. V., Hoang, D. T., Nguyen, T. N., and C. So-In, "Efficient SDN-based traffic monitoring in IoT networks with double deep Q-network", 2020. International conference on computational data and social networks, Springer

- [Puj21] Pujol-Perich, D., Suárez-Varela, J., Xiao, S., Wu, B., Cabello, A., and P. Barlet-Ros, "NetXplain: Real-time explainability of Graph Neural Networks applied to Computer Networks", 2021. MLSys workshop on Graph Neural Networks and Systems (GNNSys)
- [Raj24] Rajasekaran, S., Ghobadi, M., and A. Akella, "CASSINI: Network-Aware Job Scheduling in Machine Learning Clusters", 2024. USENIX Symposium on Networked Systems Design and Implementation (NSDI)
- [Rex06] Rexford, J., "Route optimization in IP networks", 2006. Handbook of Optimization in Telecommunications (pp. 679–700), Springer
- [Rin17] Ring, M., Dallmann, A., Landes, D., and A. Hotho, "IP2Vec: Learning Similarities Between IP Addresses", 2017. IEEE International Conference on Data Mining Workshops (ICDMW)
- [Sah23] Saha, N., Zangooui, M., Golkarifard, M., and R. Boutaba, "Deep Reinforcement Learning Approaches to Network Slice Scaling and Placement: A Survey", 2023. IEEE Communications Magazine, 61(2)
- [Sal20] Salman, O., Elhajj, I. H., Kayssi, A., and A. Chehab, "A review on machine learning-based approaches for Internet traffic classification", 2020. Springer Annals of Telecommunication
- [Sch21] Schneider, S., Qarawlus, H., and H. Karl, "Distributed Online Service Coordination Using Deep Reinforcement Learning", 2021. IEEE International Conference on Distributed Computing Systems (ICDCS)
- [Scol11] Coull, S. E., Monrose, F., and M. Bailey, "On Measuring the Similarity of Network Hosts: Pitfalls, New Metrics, and Empirical Analyses", 2011. NDSS
- [Sen04] Sen, S., Spatscheck, O., and D. Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures", 2004. ACM International conference on World Wide Web (WWW)
- [Sen24] Senevirathna, T., La, V. H., Marchal, S., Siniarski, B., Liyanage, M., and S. Wang, "A Survey on XAI for 5G and Beyond Security: Technical Aspects, Challenges and Research Directions", 2024. IEEE Communications Surveys & Tutorials

- [Sha21] Shapira, T. and Y. Shavitt, "FlowPic: A Generic Representation for Encrypted Traffic Classification and Applications Identification", 2021. IEEE Transactions on Network and Service Management, 18(2), pp. 1218-1232
- [Sha21b] Shahraki, A., Abbasi, M., Taherkordi, A., and A. D. Jurcut,, "A comparative study on online machine learning techniques for network traffic streams analysis", 2022. Elsevier Computer Networks (207)
- [Sha24] Latif, S., Boulila, W., Koubaa, A., Zou, Z., and J. Ahmad, "DTL-IDS: An optimized Intrusion Detection Framework using Deep Transfer Learning and Genetic Algorithm", 2024. Elsevier Journal of Network and Computer Applications
- [Sha25] Sharma, G. and B. Priya, "Chapter 26 - Future now: unleashing the potentials of machine learning algorithms in 6G", 2025. Advances in Computational Methods and Modeling for Science and Engineering, Morgan Kaufmann, 331-339
- [Sil16] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Driessche, G. V. D., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Silver, D., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search", 2016. Nature, vol. 529 (2016), pp. 484-503
- [Sin22] Singh, A., Prakash, S., and S. Singh, "Optimization of reinforcement routing for wireless mesh network using machine learning and high-performance computing", 2022. Concurrency Computat Pract Exper., 35(15)
- [Sol20] Soliman, H. M., Salmon, G., Sovilij, D., and M. Rao, "A Graph Neural Network Approach for Scalable and Dynamic IP Similarity in Enterprise Networks", 2020. IEEE International Conference on Cloud Networking (CloudNet)
- [Ste92] Stern, D. and P. Chemouil, "A Diagnosis Expert System for Network Traffic Management", 1992. Networks, Kobe, Japan
- [Sul23] Sulaiman, M., Moayyedi, A., Ahmadi, M., Salahuddin, M. A., Boutaba, R., and A. Saleh, "Coordinated Slicing and Admission Control Using Multi-Agent Deep Reinforcement Learning", 2023. IEEE Transactions on Network and Service Management, 20(2), pp. 1110-1124

- [Swa23] Swamy, T., Zulfiqar, A., Nardi, L., Shahbaz, M., and K. Olukotun, "Homunculus: Auto-Generating Efficient Data-Plane ML Pipelines for Datacenter Networks", 2023. ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)
- [Tan20] Tangari, G., Charalambides, M., Pavlou, G., Grazian, C., and D. Tuncer, "Classification-assisted Query Processing for Network Telemetry", 2020. Network Traffic Measurement and Analysis Conference (TMA)
- [Tan20b] Lizhuang, T., Wei, S., Zhenyi, Z., Jingying, M., Xiaoxi, L., and L. Na, "In-band Network Telemetry: A Survey", 2020. Computer Networks. 186. 10.1016
- [Tan21] Tang, F., Mao, B., Kato, N., and G. Gu, "Comprehensive Survey on Machine Learning in Vehicular Network: Technology, Applications and Challenges", 2023. IEEE Communications Surveys & Tutorials, 23(3), 2027-2057.
- [tl11] Torrey, L. and J. Shavlik, "Transfer learning", 2010. Handbook of research on machine learning applications and trends: algorithms, methods, and techniques
- [Urb23] Urbanowicz, R., Zhang, R., Cui, Y., and P. Suri, "STREAMLINE: A Simple, Transparent, End-To-End Automated Machine Learning Pipeline Facilitating Data Analysis and Algorithm Comparison", 2023. Genetic Programming Theory and Practice XIX. Genetic and Evolutionary Computation. Springer
- [Val17] A., V., M., S., D., S., and T. A., "Learning to route", 2017. ACM HotNets
- [Vin21] Vincenzi, M., Lopez-Aguilera, E., and E. Garcia-Villegas, "Timely admission control for network slicing in 5G with machine learning", 2021. IEEE Access
- [Wan22] Wang, J., Han, H., Li, H., He, S., Sharma, P. K., and L. Chen, "Multiple Strategies Differential Privacy on Sparse Tensor Factorization for Network Traffic Analysis in 5G", 2022. IEEE Transactions on Industrial Informatics, 18(3)
- [Wan24] Wang, X., Yuan, Q., Wang, Y., Gou, G., Gu, C., Yu, G., and G. Xiong, "Combine intra- and inter-flow: A multimodal encrypted traffic classification model driven by diverse features", 2024. Computer Networks, Elsevier, vol. 245

- [Wu21] Wu, Y., Zhang, K., and Y. Zhang, "Digital Twin Networks: A Survey", 2021. IEEE Internet of Things Journal, 8(18)
- [Wu22] Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., and L. He, "A survey of human-in-the-loop for machine learning", 2022. Elsevier Future Generation Computer Systems
- [Wu24] Wu, D., Wang, X., Qiao, Y., Wang, Z., Jiang, J., Cui, S., and F. Wang, "NetLLM: Adapting Large Language Models for Networking", 2024. ACM SIGCOMM 2024 Conference
- [XAI] Samek, W., Wiegand, T., and K.-R. M  ller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models", 2017. arXiv preprint arXiv:1708.08296
- [Xie18] Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J., Wang, C., and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges", 2018. IEEE Communications Surveys & Tutorials
- [Xin24] Xing, J., Hsu, K.-F., Xia, Y., Cai, Y., Li, Y., Zhang, Y., and A. Chen, "Occam: A Programming System for Reliable Network Management", 2024. ACM European Conference on Computer Systems (Eurosys)
- [Xu18] Z., X., J., T., J., M., W., Z., Y., W., H., L. C., and Y. D., "Experience-driven networking: A deep reinforcement learning based approach", 2018. IEEE INFOCOM
- [Yan18] Yang, T., Jiang, J., Liu, P., Huang, Q., Gong, J., Zhou, Y., Miao, R., Li, X., and S. Uhlig, "Elastic sketch: adaptive and fast network-wide measurements", 2018. ACM SIGCOMM Conference
- [Yan20] Yang, H., Alphones, A., Xiong, Z., Niyato, D., Zhao, J., and K. Wu,, "Artificial-Intelligence-Enabled Intelligent 6G Networks", 2020. IEEE Network, vol. 34, no. 6, pp. 272-280
- [Yu14] Yu, Y., Qian, C., and X. Li, "Distributed and collaborative traffic monitoring in software defined networks", 2014. ACM Hot topics in software defined networking

- [Zha21] Zhao, J., Jing, X., Yan, Z., and W. Pedrycz, "Network traffic classification for data fusion: A survey", 2021. Information Fusion, Elsevier, vol. 72
- [Zha24] Zhang, K., Samaan, N., and A. Karmouch, "A Machine Learning-Based Toolbox for P4 Programmable Data-Planes", 2024. IEEE Transactions on Network and Service Management, 21(4), pp. 4450-4465
- [Zil20] Meng, Z., Wang, M., Bai, J., Xu, M., Mao, H., and H. Hu, "Interpreting Deep Learning-Based Networking Systems", 2020. ACM SIGCOMM

Acknowledgments

This document is the result of a collective work. Authors of this document are the main contributors and the editors, but contributions have been also received from the following people we acknowledge Laurent Ciavaglia, Felipe Alencar Lopes, Abdelkader Lahamdi, Albert Cabellos, JosuÃ© SuÃ¡rez-Varela, Marinos Charalambides, Ramin Sadre, Pedro Martinez-Julia and Flavio Esposito

This document was also partially supported by project AI@EDGE, funded from the European Union's Horizon 2020 H2020-ICT-52 call for projects, under grant agreement no. 101015922. This research is funded in part, by the Luxembourg National Research Fund (FNR), grant reference C23/IS/18088425/COCTEL. The views expressed in this document do not necessarily reflect those of the Bank of Canada's Governing Council.

Authors' Addresses

JÃ©rÃ©me FranÃ§ois
University of Luxembourg and Inria
6 Rue Richard Coudenhove-Kalergi
L- Luxembourg
Luxembourg
Email: jerome.francois@uni.lu

Alexander Clemm
Independent
United States of America
Email: ludwig@clemm.org

Dimitri Papadimitriou
3NLab Belgium Research Center
Leuven
Belgium
Email: papadimitriou.dimitri.be@gmail.com

Stenio Fernandes
Canada Post
Canada
Email: stenio.fernandes@ieee.org

Stefan Schneider
Digital Railway (DSD) at Deutsche Bahn
Germany
Email: stefanbschneider@outlook.com

T2TRG Research Group
Internet-Draft
Intended status: Informational
Expires: 25 February 2027

M. Richardson
Sandelman Software Works
24 August 2026

A Taxonomy of operational security considerations for manufacturer
installed keys and Trust Anchors
draft-irtf-t2trg-taxonomy-manufacturer-anchors-21

Abstract

This document provides a taxonomy of methods used by manufacturers of silicon and devices to secure private keys and public trust anchors. This deals with two related activities: how trust anchors and private keys are installed into devices during manufacturing, and how the related manufacturer held private keys are secured against disclosure.

This document does not evaluate the different mechanisms, but rather just serves to name them in a consistent manner in order to aid in communication.

```
// This document is a product of the Internet Research Task Force
// (IRTF). The IRTF publishes the results of Internet-related
// research and development activities. These results might not be
// suitable for deployment.
```

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at
<https://datatracker.ietf.org/doc/draft-irtf-t2trg-taxonomy-manufacturer-anchors/>.

Discussion of this document takes place on the t2trg Research Group mailing list (<mailto:t2trg@irtf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/t2trg/>. Subscribe at <https://www.ietf.org/mailman/listinfo/t2trg/>.

Source for this draft and an issue tracker can be found at <https://github.com/t2trg/draft-irtf-t2trg-taxonomy-manufacturer-anchors>.

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 25 February 2027.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	3
1.1. Terminology	5
2. Applicability Model	6
2.1. A reference manufacturing/boot process	7
3. Types of Trust Anchors	8
3.1. First-Stage Bootloader Trust Anchor	10
3.2. Software Update Trust Anchor	10
3.3. Trusted Application Manager Anchor	11
3.4. Public WebPKI Anchors	11
3.5. DNSSEC root	11
3.6. Private/Cloud PKI anchors	12
3.7. Onboarding and other Enrollment anchors	12
3.8. Onboarded network-local anchors	13
4. Types of Device Identities	13
4.1. Manufacturer installed IDevID certificates	13
4.1.1. Avocado method: On-device private key generation . .	15
4.1.2. Broccoli method: Off-device private key generation .	16
4.1.3. Carrot method: Key setup based on secret seed	18

4.1.4. Squash method: on-device generation with Secure Element	19
4.1.5. Spinach method: Secure Element factory generation . .	19
5. Public Key Infrastructures (PKI)	20
5.1. Number of levels of certification authorities (pkilevel)	22
5.2. Protection of CA private keys	23
5.3. Preservation of operationally critical private keys . . .	24
5.3.1. Secret sharing, k-of-n	25
5.4. Supporting provisioned anchors in devices	27
6. Evaluation Questions	27
6.1. Integrity and Confidentiality of on-device data	27
6.2. Integrity and Privacy of device identity infrastructure	28
6.3. Integrity and Privacy of included trust anchors	29
7. Security Considerations	29
8. IANA Considerations	29
9. Acknowledgements	30
10. Changelog	30
11. References	30
11.1. Normative References	30
11.2. Informative References	30
Author's Address	36

1. Introduction

An increasing number of protocols derive a significant part of their security by using trust anchors [RFC4949] that are installed by manufacturers into a device during manufacturing. Disclosure of the list of trust anchors does not usually cause a problem, but changing them in any way does. This includes adding, replacing or deleting anchors.

The document [RFC6024] deals with how trust anchors are managed in the device which uses them. This document deals with how the PKI associated with such a trust anchor is managed.

Many protocols also leverage manufacturer installed identities. These identities are usually in the form of [ieee802-1AR] Initial Device Identity certificates (IDevID). The 3GPP Vendor Base Station Certificate [3GPP.33.310] is another form that uses a similar design pattern. The identity has two components: a private key that must remain under the strict control of a trusted part of the device, and a public part (the certificate), which (ignoring, for the moment, personal privacy concerns) may be freely disclosed.

There also situations where identities are tied up in the provision of symmetric shared secrets. A common example is the SIM card [3GPP.51.011]. Provisioning of a physical SIM card is generally considered a one-touch operation. A virtual SIM (an eSIM) could be factory provisioned.

It is further not unusual for many devices (particularly smartphones) to also have one or more group identity keys. This is used, for instance, in [fidotechnote] to make claims about being a particular model of phone. The key pair that does this is loaded into large batches of phones for privacy reasons: If a single key was used, then this would allow tracking. The shared key pair can not be used to identify a specific device.

The trust anchors are used for a variety of purposes. For instance, trust anchors are used to verify:

- * the signature on a software update (as per [RFC9019]),
- * a TLS Server Certificate, such as when setting up an HTTPS connection,
- * the [RFC8366] format voucher that provides proof of an ownership change.

Device identity keys are used when performing enrollment requests (in [RFC8995], and in some uses of [RFC9140].) The device identity certificate is also used to sign Evidence by an Attesting Environment (see [RFC9334]).

These security artifacts are used to anchor other chains of information such as: an Entity Attestation Token (EAT) [RFC9711] Claim as to the version of software/firmware running on a device, an EAT claim about legitimate network activity (via [I-D.ietf-iotops-mud-rats], or embedded in the IDevID in [RFC8520]).

Known software versions lead directly to vendor/distributor signed Software Bill of Materials (SBOM), such as those described by [RFC9393] and the NTIA/SBOM work [ntiasbom] and CISQ/OMG SBOM work underway [cisqsbom].

In order to manage risks and assess vulnerabilities in a Supply Chain, it is necessary to determine a degree of trustworthiness in each device. A device may mislead audit systems as to its provenance, about its software load or even about what kind of device it is (see [RFC7168] for a humorous example).

In order to properly assess the security of a Supply Chain it is necessary to understand the kinds and severity of the threats which a device has been designed to resist. To do this, it is necessary to understand the ways in which the different trust anchors and identities are initially provisioned, are protected, and are updated.

To do this, this document details the different trust anchors and identities (IDs) found in typical devices. The privacy and integrity of the trust anchors and IDs is often provided by a different, superior artifact. This relationship is examined.

While many might desire to assign numerical values to different mitigation techniques in order to be able to rank them, this document does not attempt to do that, as there are too many other (mostly human) factors that would come into play. Such an effort is more properly in the purview of a formal processes such as [ISO27001].

This document is a product of the IRTF Thing-to-Thing Research Group (T2TRG) and represents the consensus of the research group.

1.1. Terminology

This document is not a standards track document, and it does not make use of formal requirements language.

This document defines a number of hyphenated terms, and they are summarized here:

birth-certificates: another term for IDevID

device-generated: a private or symmetric key that is generated on the device

infrastructure-generated: a private or symmetric key that is generated by some system, likely located at the factory that built the device

IDevID-certificates: an initial device identity certificate as per [ieee802-1AR]

key-executives: the people who are entrusted with pieces (shares) of a split secret Section 5.3.1

pkilevel: the number of Certification Authorities in a public key infrastructure, counting the root, or trust-anchor as the first level, and not counting the End-Entity certificates Section 5.1

manufacturer-installed-certificates: or MIC, see idevid

mechanically-installed: when a key or certificate is programmed directly into non-volatile storage by an out-of-band mechanism such as JTAG [JTAG]

network-transferred: when a key or certificate is transferred into a system via private interface, such as serial console, JTAG managed mailbox, or other physically private interface

network-transferred: when a key or certificate is transferred into a system using a network interface which would be available after the device has shipped. This applies even if the network is physically attached using a bed-of-nails [BedOfNails].

device/infrastructure-co-generated: when a private or symmetric key is derived from a secret previously synchronized between the silicon vendor and the factory using a common algorithm.

TPM: a Trusted Platform Module, such as the set of devices standardized by the Trusted Computing Group (TCG). See [TPM20spec].

In addition, Section 4.1 introduces three primary private key generation techniques named *_arbitrarily_* after three vegetables (avocado, broccoli, and carrot) and two secondary ones named after two more (squash and spinach). The two secondary ones refer to methods where a secure element is involved, and mnemonically start with the same letter: S.

2. Applicability Model

There is a wide variety of devices to which this analysis can apply (see [I-D.ietf-iotops-7228bis].) This document will use a J-group processor as a sample. This class is sufficiently large to experience complex issues among multiple CPUs, packages and operating systems, but at the same time, small enough that this class is often deployed in single-purpose IoT-like uses. Devices in this class often have Secure Enclaves, and can include silicon manufacturer controlled processors in the boot process. Note that access to the secure enclave is often not available to system integrators. A very common IoT platform, the Raspberry PI, has a secure enclave with the GPU, but access to it is not available.

Almost all larger systems (servers, laptops, desktops) include a Baseboard Management Controller (BMC), which ranges from an M-Group Class 3 MCU, to a J-Group Class 10 CPU (see, for instance [openbmc] which uses a Linux kernel and system inside the BMC). As the BMC usually has complete access to the main CPU's memory, I/O hardware and disk, the boot path security of such a system needs to be understood first as being about the security of the BMC.

2.1. A reference manufacturing/boot process

In order to provide for immutability and privacy of the critical trust anchors and IDs, many CPU manufacturers will provide for some kind of private memory area which is only accessible when the CPU is in certain privileged states. See the Terminology and Architecture sections of [RFC9397], notably Trusted Execution Environment (TEE), Rich Execution Environment (REE), and Trusted Application Manager (TAM).

The private memory that is important is usually non-volatile and rather small. It may be located inside the CPU silicon die, or it may be located externally. If the memory is external, then it is usually encrypted by a hardware mechanism on the CPU, with only the key kept inside the CPU.

The entire mechanism may be external to the CPU in the form of a hardware-TPM module, or it may be entirely internal to the CPU in the form of a firmware-TPM. It may use a custom interface to the rest of the system, or it may implement the TPM 1.2 or TPM 2.0 specifications (see [TPM20spec].) Those details are important to performing a full evaluation, but do not matter much to this model (see initial-enclave-location below).

During the manufacturing process, once the components have been soldered to the board, the system is usually put through a system-level test. This is often done as a "bed-of-nails" test [BedOfNails], where the board has key points attached mechanically to a test system. A [JTAG] process tests the System Under Test, and then initializes some firmware into the still empty flash storage.

It is now common for a factory test image to be loaded first: this image will include code to initialize the private memory key described above, and will include a first-stage bootloader and some kind of (primitive) Trusted Application Manager (TAM). The TAM is a piece of software that lives within the trusted execution environment.

Embedded in the first-stage bootloader will be a Trust Anchor that is able to verify the second-stage bootloader image.

After the system has undergone testing, the factory test image is erased, leaving the first-stage bootloader. One or more second-stage bootloader images are installed. The production image may be installed at that time, or if the second-stage bootloader is able to install it over the network, it may be done that way instead.

There are many variations of the above process, and this section is not attempting to be prescriptive, but to provide enough illustration to motivate subsequent terminology.

The process may be entirely automated, or it may be entirely driven by humans working in the factory, or a combination of the above.

These steps may all occur on an access-controlled assembly line, or the system boards may be shipped from one place to another (maybe another country) before undergoing testing.

Some systems are intended to be shipped in a tamper-proof state, but it is usually not desirable that bed-of-nails testing be possible without tampering, so the initialization process is usually done prior to rendering the system tamper-proof. An example of a one-way tamper-proof, weather resistant treatment might to mount the system board in a case and fill the case with resin.

Quality control testing may be done prior to, as well as after, the application of tamper-proofing, as systems which do not pass inspection may be reworked to fix flaws, and this should ideally be impossible once the system has been made tamper-proof.

3. Types of Trust Anchors

Trust Anchors are fundamentally public keys with authorizations implicitly attached through the code that references them.

They are used to validate other digitally signed artifacts. Typically, these are chains of PKIX certificates leading to an End-Entity certificate (EE).

The chains are usually presented as part of an externally provided object, with the term "externally" to be understood as being as close as untrusted flash, to as far as objects retrieved over a network.

There is no requirement that there be any chain at all: the trust anchor can be used to validate a signature over a target object directly.

The trust anchors are often stored in the form of self-signed certificates. The self-signature does not offer any cryptographic assurance, but it does provide a form of error detection, providing verification against non-malicious forms of data corruption. If storage is at a premium (such as inside-CPU non-volatile storage) then only the public key itself need to be stored. For a 256-bit ECDSA key, this is 32 bytes of space.

The following subsections explain a number of different kinds of trust anchors that a manufacturer can include into a device. Each kind of trust anchor has different kinds of trust associated with it based on what authorization is associated with the key. That is, these different anchors do different things, and so the risk associated with the anchor depends upon what those things that can be done are.

There are some characteristics associated with each trust anchor that depend upon the associated risk.

1. There is a risk around whether or not a trust anchor can be replaced or modified so that it's a different anchor, (with the associated private key) controlled by a different entity. This is not a risk associated with loss of the original manufacturer's private key, but with integrity of the public key.
2. There is a risk that multiple trust anchors could be added to the device. This would not replace the manufacturer's anchor, but augment it with additional trust anchors not authorized by the manufacturer.
3. There is a risk that a trust anchor could be removed from the device, or rendered unusable. For instance, it might be impractical to replace a trust anchor, but it might be possible via fault injection or high-energy radiation for an attacker to corrupt one or two bytes of a trust anchor, rendering the anchor useless.
4. There is a risk associated with compromise of the associated private key that goes with the (public key) of the trust anchor. While this is the most obvious concern expressed: that an attacker gets control of the key, the above three risks may be more practical for some attackers.
5. There is a risk that the manufacturer will lose access to the private key, without the private key being compromised. No attacker possesses the private key, but neither can the manufacturer use the key anymore.

Should any of the above things occur, the manufacturer will be in a position where they might have to deploy new trust anchors to every device. For some trust anchors, they may be safely replaced with an over-the-air update. For the trust anchor that authorizes over-the-air updates, the manufacturer might be in a position where every single device has to be recalled and serviced with specialized hardware capable of updating the firmware. For some devices, the hardware will have to be entirely replaced, typically at great cost.

This has already happened, see for example: [leakedmsikey] and [hyundaiaexample].

3.1. First-Stage Bootloader Trust Anchor

This anchor is part of the first-stage bootloader, and it is used to validate a second-stage bootloader which may be stored in external flash.

This is called the first-stage bootloader trust anchor. In most cases this trust anchor is burnt into fuses in the CPU, often within the die along with the first-stage bootloader itself. It can not be changed without replacement of the entire device. Replacement, removal or modification of this trust anchor is improbable.

The anchor could be rendered unusable if additional fuses can be blown. Some fuse implementations allow for bits to be changed from 1 (unblown), to 0 (blown). Access to blow the fuses is usually restricted. On some devices, it requires voltages not normally present, making this impossible to do by software. However, it might be possible for an attacker to blow fuses using an external high voltage probe or via injection of gamma ray. The likely result however is that the device would no longer boot.

3.2. Software Update Trust Anchor

This anchor is used to validate the main application (or operating system) load for the device.

It can be stored in a number of places. First, it may be identical to the First Bootloader Trust Anchor.

Second, it may be stored in the second-stage bootloader, and therefore its integrity is protected by the First Bootloader Trust Anchor.

Third, it may be stored in the application code itself, where the application validates updates to the application directly (update in place), or via a double-buffer arrangement. The initial (factory) load of the application code initializes the trust arrangement.

In this situation the application code is not started in a secured boot path. The second-stage bootloader does not validate the application/operating system before starting it, but it may still provide measured boot mechanism (see [measuredboot], section 2.3.)

3.3. Trusted Application Manager Anchor

This anchor is the key used by the [RFC9397] Trusted Application Manager (TAM). Code which is signed by this anchor will be given execution privileges as described by the manifest which accompanies the code.

The TAM software `_itself_` will typically be verified by the first-stage bootloader, so it needs to be signed by the First Bootloader Trust Anchor. The TAM could also be a component of the first or second stage bootloader.

This privilege may include updating anchors that are present within the TAM, or elsewhere in the Trusted Execution Environment.

3.4. Public WebPKI Anchors

These anchors are used to authenticate HTTPS certificates from websites. These anchors are typically distributed as part of desktop browsers, and via desktop operating systems. On IoT devices with specific purposes [RFC8520], a limited number of connections is expected, so a limited number of trust anchors is usually distributed.

The exact set of these anchors is not precisely defined: it is usually determined by the browser vendor (e.g., Mozilla, Google, Apple, Microsoft), or the operating system vendor (e.g., Apple, Google, Microsoft, Ubuntu). In most cases these vendors look to the CA/Browser Forum [CABFORUM] for inclusion criteria.

3.5. DNSSEC root

This anchor is part of the DNS Security extensions. It provides an anchor for integrity protection of DNS lookups. Secure DNS lookups may be important in order to get access to software updates. This anchor is now scheduled to change approximately every 3 years, with the new key announced several years before it is used, making it possible to embed keys that will be valid for up to five years.

This trust anchor is typically part of the application/operating system code and is usually updated by the manufacturer when they do updates. However, a system that is connected to the Internet may update the DNSSEC anchor itself through the mechanism described in [RFC5011].

There are concerns that there may be a chicken-and-egg situation for devices that have remained in a powered off state (or disconnected from the Internet) for a period of many years. Upon being reconnected, the device would be unable to do DNSSEC validation because the trust anchor within the device would be too old.

This could be fixed with a software/firmware update. However, it could also be that in such a situation, that the device would be unable to validate the DNSSEC names required in order to obtain the operating system update.

3.6. Private/Cloud PKI anchors

It is common for many IoT and network appliances to have links to vendor provided services. For instance, the IoT device that calls home for control purposes, or the network appliance that needs to validate a license key before it can operate. (This may be identical to, or distinct from a Software Update anchor. In particular, the device might call home over HTTPS to learn if there is a software update that needs to be done, but the update is signed by another key.)

Such vendor services can be provided with public certificates, but the very short lifetime for public certificates [CABForum90day] precludes their use in many operational environments. Instead, a private PKI anchor is included. This can be in the form a multi-level PKI (as described in Section 5.1), or degenerate to a level-1 PKI: a self-signed certificate.

A level-1 PKI is very simple to create and operate, and there are innumerable situations where there is just a call to the "curl" utility [curl], with a "--pinnedpubkey" option to specify the anchor.

3.7. Onboarding and other Enrollment anchors

[RFC8995], [RFC8572], and [RFC8366] specify mechanisms for onboarding of new devices. The voucher artifact is transferred to the device by various means, and the device must verify the signature on it. This requires a trust anchor to be built-in to the device, and some kind of private PKI be maintained by the vendor (or its authorized signing designate). [I-D.ietf-anima-masa-considerations] provides some advice on choices in PKI design for a MASA. The taxonomy presented

in this document applies to describing how this PKI has been designed.

3.8. Onboarded network-local anchors

[RFC7030], [RFC8995] and [RFC9641] provide mechanisms by which new trust anchors may be loaded by a device during an onboarding process. The trust anchors involved are typically local to an enterprise and are used to validate connections to other devices in the network.

This typically includes connections to network management systems that may also load or modify other trust anchors in the system. [I-D.ietf-anima-masa-considerations] provides some advice in the BRSKI [RFC8995] case for appropriate PKI complexity for such local PKIs.

Note that this trust anchor is that of the network operator, and it is not related to the trust anchor described in the previous section that is used to validate an ownership transfer.

4. Types of Device Identities

Device identities are installed during manufacturing time for a variety of purposes.

Identities require some private component. Asymmetric identities (e.g., RSA, ECDSA, EdDSA systems) require a corresponding public component, usually in the form of a certificate signed by a trusted third party.

This certificate associates the identity with attributes.

The process of making this coordinated key pair and then installing it into the device is called identity provisioning.

4.1. Manufacturer installed IDevID certificates

[ieee802-1AR] defines a category of certificates that are installed by the manufacturer which contain a device unique serial number.

A number of protocols depend upon this certificate.

- * [RFC8572] and [RFC8995] introduce mechanisms for new devices (called pledges) to be onboarded into a network without intervention from an operator. A number of derived protocols such as [RFC9733], [I-D.ietf-anima-constrained-voucher], [I-D.ietf-anima-brski-cloud] extend this in a number of ways.

- * [RFC9334] depends upon a key provisioned into the Attesting Environment to sign Evidence. The IDevID may be used for this.
- * [RFC9019] depends upon a key provisioned into the device in order to decrypt software updates. Both symmetric and asymmetric keys are possible. In both cases, the decrypt operation depends upon the device having access to a private key provisioned in advance. The IDevID can be used for this if algorithm choices permit. ECDSA keys do not directly support encryption in the same way that RSA does, for instance, but the addition of HPKE [RFC9180] allows for use of other key types. There may be other legal considerations why the IDevID might not be used, and a second key provisioned.

The manufacturer has the responsibility to provision a key pair into each device as part of the manufacturing process. There are a variety of mechanisms to accomplish this, which this section details.

There are three fundamental ways to generate IDevID certificates for devices:

1. generating a private key on the device, creating a Certificate Signing Request (or equivalent), and then returning a certificate to the device.
2. generating a private key outside the device, signing the certificate, and then installing both into the device.
3. deriving the private key from a previously installed secret seed, that is shared with only the manufacturer.

There are additional variations where the IDevID is provided as part of a Trusted Platform Module (TPM) or Secure Element (SE). The vendor may purchase such devices from another vendor, and that vendor often offers provisioning of a key pair into the device as a service.

The document [I-D.moskowitz-ecdsa-pki] provides some practical instructions on setting up a reference implementation for ECDSA keys using a three-tier mechanism.

The names of the five methods below are intended to be mnemonic.

- * Avocados have big seeds inside of them.
- * Broccoli has a seed on the top (which is eaten).
- * Carrots have a complicated seed process involving multiple years and many leaves.

- * The two methods involving Secure Elements are named with the letter S.

4.1.1. Avocado method: On-device private key generation

In this method, the device generates a private key on the device. This is done within some very secure aspect of the device, such as in a Trusted Execution Environment, and the resulting private key is then stored in a secure and permanent way. The permanency may extend beyond use of on-CPU flash, and could even involve blowing of one-time fuses.

Generating the key on-device has the advantage that the private key never leaves the device. The disadvantage is that the device may not have a verifiable random number generator. The use of a pseudo-random number generator needs to be well seeded as explained in [RFC4086]. [factoringrsa] is an example of a random number generation failure that was visible through survey of public keys, and led to a successful attack on the private keys.

There are a number of options of how to get the public key from the device to the certification authority. As it is a public key, privacy is less of a concern, and the focus is on integrity. (However, disclosing the public key may have impacts on trackability of the device.)

So, transmission must be done in an integral manner, and must be securely associated with an assigned serial number. The serial number goes into the certificate, and the resulting certificate needs to be loaded into the manufacturer's asset database, and returned to the device to be stored as the IDevID certificate.

One way to do the transmission is during a factory Bed of Nails test (see [BedOfNails]) or JTAG Boundary Scan. There are two ways this can be done. In the _device-generated_ / _mechanically-installed_ method, the public key or certificate signing request (CSR) is retrieved by doing JTAG operations directly with an EEPROM or RAM. The CPU is not involved, and the memory accessed is likely not encrypted. (The private key might be in the same place, or it could be safely elsewhere).

A different JTAG mechanism would involve reading/writing to some other area of the CPU, such as an on-CPU static RAM or debug area. In this case, the CPU is involved. When done via a physical connection like this, then this is referred to as a _device-generated_ / _mechanically-transferred_ method. The key difference is that CPUs usually have much better defenses against JTAG use after manufacturing, often via one time fuses that can be blown.

There are other ways that could be used where a certificate signing request is sent over a special network channel when the device is powered up in the factory. This is referred to as the `_avocado device-generated_` / `_network-transferred_` method.

Regardless of how the certificate signing request is sent from the device to the factory, and how the certificate is returned to the device, a concern from production line managers is that the assembly line may have to wait for the certification authority to respond with the certificate. This is inherently a synchronous process, as the process can not start until the private key is generated and stored.

After the key generation, the device needs to set a flag such that it no longer will generate a new key, and will not accept a new IDevID via the factory network connection. This may be a software setting, or could involve blowing a one-time fuse.

Devices are typically constructed in a fashion such that the device is unable to ever disclose the private key via an external interface. This is usually done using a secure-enclave provided by the CPU architecture in combination with on-chip non-volatile memory.

The risk is that if an attacker with physical access is able to put the device back into an unconfigured mode, then the attacker may be able to substitute a new certificate into the device. It is difficult to construct a rationale for doing this as the attacker would not be able to forge a certificate from the manufacturer's CA. Other parties that rely on the IDevID would see the device as an imposter if another CA was used. However, if the goal is theft of the device itself (without regard to having access to firmware updates), then use of another manufacturer identity may be profitable. Stealing a very low value item, such as a light bulb, makes very little sense. Stealing medium value items, such as appliances, or high-value items such as cars, yachts or even airplanes, would make sense. Replacing the manufacturer IDevID permits the attacker to also replace the authority to transfer ownership in protocols like [RFC8995].

4.1.2. Broccoli method: Off-device private key generation

In this method, a key pair is generated in the factory, outside of the device. The factory keeps the private key in a restricted area, but uses it to form a Certificate Signing Request (CSR). The CSR is passed to the manufacturer's Certification Authority (CA), and a certificate is returned. Other meta-data is often also returned, such as a serial number.

Generating the key off-device has the advantage that the randomness of the private key can be better ensured. As the private key is available to the manufacturing infrastructure, the authenticity of the public key is well known ahead of time.

The private key and certificate can be programmed into the device along with the initial bootloader firmware in a single step.

As the private key can be known to the factory in advance of the device being ready for it, the certificate can also be generated in advance. This hides the latency to talk to the CA, and allows for the connectivity to the CA to be less reliable without shutting down the assembly line. A single write to the flash of the device can contain the entire firmware of the device, including configuration of trust anchors and private keys.

The major downside to generating the private key off-device is that it could be seen by the manufacturing infrastructure. It could be compromised by humans in the factory, or the equipment could be compromised. The use of this method increases the value of attacking the manufacturing infrastructure.

If private keys are generated by the manufacturing plant, and are immediately installed, but never stored, then the window in which an attacker can gain access to the private key is immensely reduced. But, the process then becomes more synchronous, negating much of the advantage of such a system.

As in the previous case, the transfer may be done via physical interfaces such as bed-of-nails, giving the `_broccoli infrastructure-generated_ / _mechanically-transferred_` method.

There is also the possibility of having a `_broccoli infrastructure-generated_ / _network-transferred_` key. There is a support for "server-generated" keys in [RFC7030], [RFC8894], and [RFC4210]. All methods strongly recommend encrypting the private key for transfer. This is difficult to comply with here as there is not yet any private key material in the device, so in many cases it will not be possible to encrypt the private key. Still, it may be acceptable if the device is connected directly by a wired network and unroutable addresses are used. A private wired network can often be made as secure as a bed-of-nails interface.

4.1.3. Carrot method: Key setup based on secret seed

In this method, a random symmetric seed is generated by a component manufacturer. This is typically the manufacturer of the CPU, often a system on a chip (SOC). In this section there are two suppliers involved: the first is the familiar one that is responsible for the entire device (the device supplier), and the second one is the vendor of silicon chip. This silicon chip is where a symmetric seed key has been provisioned.

In this process, the silicon chip supplier provisions a unique secret into each device shipped. This is typically at least 256 bits in size, but often is larger. Note that the silicon chip supplier has to create an inventory specific for each client.

This can be via fuses blown in a CPU's Trusted Execution Environment [RFC9397], a TPM, or a Secure Element that is provisioned at its fabrication time.

This seed value is revealed to the board/system manufacturer only via a secure channel. Upon first boot, the system (within a TEE, a TPM, or Secure Element) will generate a key pair using this seed to initialize a Pseudo-Random-Number-Generator (PRNG). The manufacturer, in a separate system, will initialize the same PRNG and, thus generate the same key pair. The manufacturer then derives the public key part, and signs it with their certification authority (CA), which inserts this public key into a certificate. The private part is then destroyed by the manufacturer, ideally never stored or seen by anyone. This same mechanism is used by the TCG for manufacturer provisioning of EK certificates in TPMs, as explained by Section 2.1.1 of [TPMEK].

The certificate (being public information) is placed into a database, in some cases it is loaded by the device as its IDevID certificate, in other cases, it is retrieved during the onboarding process based upon a unique serial number asserted by the device.

In some ways, this method appears to have all the tradeoffs of the previous two methods: the device must correctly derive its own private key, and the manufacturer has access to the private key, making it also vulnerable. But, the device does not depend upon any internal source of random numbers to derive its key.

The manufacturer does all the private key manipulation in a secure place, probably offline, and need never involve the actual physical factory. The manufacturer can even do this in a different country.

The security of the process rests upon the difficulty in extracting the seed provided by the silicon chip supplier. While the silicon chip supplier must operate their factory in a much more secure fashion, which has a much higher cost, the exposure for this facility can be much better controlled. The device manufacturing factory, which has many more components as input, including device testing, can operate at a much lower risk level as a result.

Additionally, there are some other advantages to the device manufacturer: The private keys and certificates may be calculated by the manufacturer asynchronously to the manufacturing process, either done in batches in advance of actual manufacturing, or on demand when an IDevID is requested.

There are, however, additional downsides of this method for manufacturer: the cost of the silicon is often higher, and may involve additional discrete parts. The higher cost is the result of outsourcing the security risk to the silicon manufacturer supplier.

The weakest link in this process is that the resulting seeds must be communicated to the device manufacturer securely, usually in batches. This will often be done by physical courier, but other arrangements are possible. The device manufacturer must store and care for these seeds very carefully, but as soon as a certificate has been generated, the seeds can be destroyed.

4.1.4. Squash method: on-device generation with Secure Element

In this method, a key-pair is generated by the device using a secure element. (It may be a discrete TPM, but when TPM is used to generate keys, that method is considered to fall into the avocado category)

The secure element provides additional assurance that the private key was properly generated. Secure elements are designed specifically so that private keys can not be extracted from the device, so even if the device is attacked in a sophisticated way, using hardware, the private key will not be disclosed.

4.1.5. Spinach method: Secure Element factory generation

In this method, a key-pair is generated by a Secure Element silicon chip supplier in their factory. This method is essentially identical to the carrot method, but it occurs in a different factory!

As a result the choice of which certification authority (CA) gets used may be very different. It is typical for the silicon chip supplier to operate a CA themselves. There are a few options:

1. they may put IDevIDs into the device which are generic to the silicon chip supplier
2. they may operate a CA on behalf of the device manufacturer,
3. they may even connect over a network to the device manufacturer's CA.

The device manufacturer receives the secure element devices in batches in a similar way that they receive other parts. The secure elements are placed by the device manufacturer's plant into the devices.

Upon first boot the device manufacturer firmware can read the IDevID certificate that has been placed into the secure elements, and can ask the secure element to perform signing operations using the private key contained in the secure element. However, the private key can not be extracted!

Despite the increased convenience of this method, there may be a risk if the secure elements are stolen in transport. A thief could use them to generate signatures that would appear to be from device manufacturer devices. To deal with this, there is often a simple activation password that the device manufacturer firmware must provide to the secure element in order to activate it. Note that this password is probably stored in the clear in the device manufacturer's firmware: it can't be encrypted, because the secure source of decryption keys would be in the secure element, creating a cyclic dependency.

Note that the Secure Element silicon vendor has to create an inventory specific for each client.

5. Public Key Infrastructures (PKI)

This section covers the attributes of the Public Key Infrastructure that a manufacturer must operate in order to provision IDevID certificates to devices.

[RFC5280] describes the format for PKIX certificates. Numerous mechanisms of enrollment have been defined (including: EST [RFC7030], CMP [RFC4210], SCEP [RFC8894]).

[RFC5280] provides mechanisms to deal with multi-level certification authorities, but it is not always clear what operating rules apply.

The certification authority (CA) that is central to [RFC5280]-style public key infrastructures can suffer three kinds of failures:

1. disclosure of the private key,
2. loss of access to the private key,
3. inappropriate/unauthorized signing of a certificate or artifact by an unauthorized actor.

A PKI which discloses one or more private certification authority keys is no longer secure.

An attacker can create new identities, and forge certificates connecting existing identities to attacker controlled public/private key pairs. This can permit the attacker to impersonate any specific device.

In case 3, a failure occurs if the CA is convinced to sign (or issue) a certificate which it is not authorized to sign. See for instance [ComodoGate]. This is an authorization failure, and while a significant event, it does not result in the CA having to be re-initialized from scratch as no private keys were disclosed. In ecosystems where CRLs or OCSP is used, then the unauthorized certificate can be revoked.

This is distinguished from when access to the private key is lost, but the key has not been disclosed. Any signatures that have already been made continue to be trustworthy, however, no new signature can be made. This renders the CA useless. If this concerns the firmware signing key, it likely results in a recall of all products that need to use this trust anchor. (Note: that there are some situations where a trust anchor will be created, it will then be used to sign a number of subordinate CAs, enough for the anticipated lifespan of the anchor, and then the private key intentionally destroyed.)

If the PKI uses Certificate Revocation Lists (CRL), then an attacker that has access to the private key can also revoke existing identities.

In the other direction, a PKI which loses access to a private key can no longer function. This does not immediately result in a failure, as existing identities remain valid until their expiry time (notAfter). However, if CRLs or OCSP are in use, then the inability to sign a fresh CRL or OCSP response will result in all identities becoming invalid once the existing CRLs or OCSP statements expire.

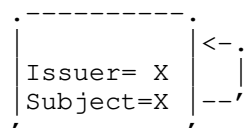
This section details some nomenclature about the structure of certification authorities.

5.1. Number of levels of certification authorities (pkilevel)

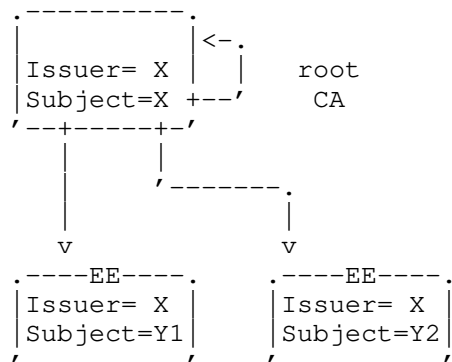
Section 6.1 of [RFC5280] provides a Basic Path Validation. In the formula, the certificates are arranged into a list.

The certification authority (CA) starts with a Trust Anchor. This is counted as the first level of the authority.

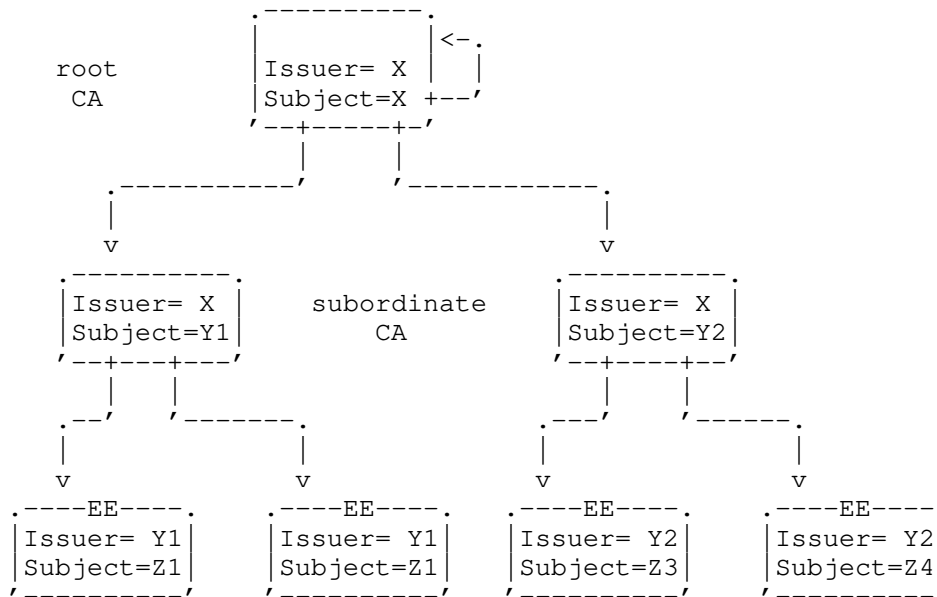
In the degenerate case of a self-signed certificate, this constitutes a one level PKI.



The private key associated with the Trust Anchor signs one or more certificates. When this first level authority signs only End-Entity (EE) certificates, then this is a two level PKI.



When this first level authority signs subordinate certification authorities, and those certification authorities sign End-Entity certificates, then this is a three level PKI.



In general, when arranged as a tree, with the End-Entity certificates at the bottom, and the Trust Anchor at the top, then the pkilevel is defined to be where the deepest EE certificates are, counting from one.

It is quite common to have a three-level PKI, where the root (level one) of the CA is stored in a Hardware Security Module (HSM) in a way that it cannot be continuously accessed ("offline"), while the level two subordinate CA can sign certificates at any time ("online").

5.2. Protection of CA private keys

The private key for the certification authorities must be protected from disclosure. The strongest protection is afforded by keeping them in an offline device, passing Certificate Signing Requests (CSRs) to the offline device by human process.

For examples of extreme measures, see [kskceremony]. There is however a wide spectrum of needs, as exemplified in [rootkeyceremony].

This is inconvenient, and may involve latencies of days, possibly even weeks to months if the offline device is kept in a locked environment that requires multiple keys to be present.

There is therefore a tension between protection and availability. Convenient and timely access to sign new artifacts is not something that is just nice to have. If access is inconvenient then it may cause delays for signing of new code releases, or it may incentivize technical staff to build in workarounds in order that they can get their job done faster. To address the conflicting needs, some levels of the PKI be offline, and some levels of the PKI be online.

5.3. Preservation of operationally critical private keys

A public key (or certificate) is installed into target device(s) as a trust anchor. It is there in order to verify further artifacts, and it represents a significant investment. Trust anchors must not be easily replaced by attackers, and securing the trust anchor against such tampering may involve burning the trust anchor into unchangeable fuses inside a CPU.

Replacement of the anchor can involve a physical recall of every single device. It is therefore important that the trust anchor is usable for the entire lifetime of every single one of the devices.

The protections discussed in the previous section aim to thwart attacks. The attacker wants to get access to the private key material, or to convince the infrastructure to use the private key material to their bidding. Such an event, if it was undetected, would be catastrophic. It would render almost every device useless (or potentially dangerous) until the anchor could be replaced.

There is a different situation, however, which would lead to a similar result. If the legitimate owner of the trust anchor infrastructure loses access to the private keys, then an equally catastrophic situation occurs.

Code signing is usually done with a keypair that is attached to an End-Entity (EE) Certificate, from a trust anchor that has been assigned code signing authorization. The private key of this EE also needs to be treated with significant care. However, as long as the code signing CA's trust anchor and private key are preserved, it is usually preferable to lose access to the code signing EE certificate private key rather than risk disclosure. A new code signing EE certificate can be created if needed. This loss is not hassle-free; access to the code signing key can be critical path to getting fixes deployed.

There are many situations that could lead to this. The most typical situation would seem to be some kind of physical damage: a flood, a fire. Less obvious situations could occur if CA key staff lose access to authentication secrets, such as a password, or if the human with the password(s) dies, or becomes incapacitated.

Backups of critical material are routinely done. Storage of backups offsite addresses the risk of physical damage, and in many cases the organization maintains an entire set of equipment at another location.

The question then becomes how the backups are unlocked or activated. Why attack the primary site physically if an attacker can target the backup site, or the people whose job it is to activate the backup site?

Consider the situation where a hurricane or earthquake takes out all power and communications at an organizations' primary location, and it becomes necessary to activate the backup site. What does it take to do that?

Typically, the secrets will be split using [shamir79] into multiple pieces, each piece being carried with a different trusted employee.

In [kskceremony], the pieces are stored on smartcards that are kept in a vault, and the trusted people carry keys to the vault.

One advantage of this mechanism is that if necessary, the doors to the vault can be drilled out. This takes some significant time and leaves significant evidence, so it can not be done quietly by an attacker. In the case of the DNSSEC Root, a failure of the vault to open actually required this to be done.

In other systems the digital pieces are carried on the person themselves, ideally encrypted with a password known only to that person.

[shamir79] allows for keys to be split up into many components (n of them), where only some smaller number of them, k , need to be present to reconstruct the secret. This is known as a (k, n) threshold scheme.

5.3.1. Secret sharing, k -of- n

In this document, each of the people who hold a piece of the secret are referred to as Key-Executives.

The choice of n , and the choice of k is therefore of critical concern. It seems unwise for an organization to publish them, as it provides some evidence as to how many Key-Executives would need to be coerced.

The identities of the n Key-Executives should also be confidential. The list of who they are should probably be limited to the members of the board and executive. There does not seem to be any particular reason for the Key-Executives to be members of the board, but having a long term relationship with the enterprise seems reasonable, and a clear understanding of when to use the piece.

The use of secret sharing, as opposed to providing each Key-Executive a complete backup copy ($k=1$), is that it means that no individual Key-Executive can perform operations on their own. Instead, a minimum of k Key-Executives is needed to do any operation. This protects the organization against corruption and coercion of any one Key-Executive. As the Key-Executives can not operate on their own, there is also a lower risk to them.

The minimum number of people (k) needed should also remain confidential.

A number that can be published is the difference between k and n , which represents the number of redundant Key-Executives that exist. The publication of this number is an assurance to the organizations' customers that there is a business continuity plan.

An enterprise that has operations in multiple places may be better positioned to survive incidents that disrupt travel. For instance, an earthquake, tsunami, or pandemic not only has the possibility to incapacitate Key-Executives but it could also damage smartcards or USB key that the shares are stored on. [shamir79] suggests that $n=2k-1$, which implies that a simple majority of Key-Executives are needed to reconstruct the secret.

However, there are other values of k with some different tradeoffs: requiring a majority of Key-Executives might be difficult when travel is impossible. For instance, a value of k set to be less than a simple majority. This enables the situation where Key-Executives are split between two or more continents (with each continent having at least k Key-Executives) would allow either continent to continue operations without the other group.

Secret sharing may also be a good way to manage a code signing or firmware update signing key, even when that is an End-Entity Certificate. Split it among development groups in three time zones (eight hours apart), such that any of those development groups have

enough shares, and can therefore issue an emergency security patch. Another way would be to have three End-Entity certificates that can sign code, and have each time zone sign their own code. That implies that there is at least a level two PKI around the code signing process, and that any bootloaders that need to verify the code being starting it are able to do PKI path operations.

5.4. Supporting provisioned anchors in devices

IDevID-type Identity (or Birth) Certificates which are provisioned into devices need to be signed by a certification authority maintained by the manufacturer. During the period of manufacture of new product, the manufacturer needs to be able to sign new Identity Certificates.

During the anticipated lifespan of the devices the manufacturer needs to maintain the ability for third parties to validate the Identity Certificates. If there are Certificate Revocation Lists (CRLs) involved, then they will need to re-signed during this period. Even for devices with a short active lifetime, the lifespan of the device could be very long if devices are kept in a warehouse for many decades before being activated.

Trust anchors which are provisioned in the devices will have corresponding private keys maintained by the manufacturer. The trust anchors will often anchor a PKI which is going to be used for a particular purpose. There will be End-Entity (EE) certificates of this PKI which will be used to sign particular artifacts (such as software updates), or messages in communications protocols (such as TLS connections). The private keys associated with these EE certificates are not stored in the device, but are maintained by the manufacturer. These need even more care than the private keys stored in the devices, as compromise of the software update key compromises all the devices, not just a single device.

6. Evaluation Questions

This section recaps the set of questions that may need to be answered. This document does not assign valuation to the answers.

6.1. Integrity and Confidentiality of on-device data

initial-enclave-location: Is the location of the initial software trust anchor internal to the CPU package? Some systems have a software verification public key which is built into the CPU package, while other systems store that initial key in a non-volatile device external to the CPU.

initial-enclave-integrity-key: If the first-stage bootloader is external to the CPU, it will often need integrity protection. To verify that, another key is needed. It could be a keyed hash. Where is the key for this hash located?

initial-enclave-confidentiality-key: If the first-stage data is external to the CPU, is it kept confidential by use of encryption?

first-stage-exposure: The number of people involved in the first stage initialization. An entirely automated system would have a number zero. A factory with three 8-hour shifts might have a number that is a multiple of three. A system with humans involved may be subject to bribery attacks, while a system with no humans may be subject to attacks on the system which are hard to notice.

first-second-stage-gap: how far and long does a board travel between being initialized with a first-stage bootloader to where it is locked down so that changes to the bootloader can no longer be made. For many situations, there is no distance at all as they occur in the same factory, but for other situations boards are manufactured and tested in one location, but are initialized elsewhere.

6.2. Integrity and Privacy of device identity infrastructure

For IDevID provisioning, which includes a private key and matching certificate installed into the device, the associated public key infrastructure that anchors this identity must be maintained by the manufacturer.

identity-pki-level: referring to Section 5.1, the level number at which End-Entity certificates are present.

identity-time-limits-per-subordinate: how long is each subordinate CA maintained before a new subordinate CA key is generated? There may be no time limit, only a device count limit.

identity-number-per-subordinate: how many identities are signed by a particular subordinate CA before it is retired? There may be no numeric limit, only a time limit.

identity-anchor-storage: how is the root CA key stored? An open description that might include whether an HSM is used, or not, or even the model of it.

identity-shared-split-extra: referring to Section 5.3.1, where a

private key is split up into n components, of which k are required to recover the key, this number is equal to $n-k$. This is the number of spare shares. Publishing this provides a measure of how much redundancy is present while not actually revealing either k or n .

identity-shared-split-continentals: the number of continents on which the private key can be recovered without trans-continental travel by any of the secret shareholders.

6.3. Integrity and Privacy of included trust anchors

For each trust anchor (public key) stored in the device, there will be an associated PKI. For each of those PKI the following questions need to be answered.

pki-level: how deep is the EE that will be evaluated, based upon the criteria in Section 5.1

pki-algorithms: what kind of algorithms and key sizes can actively be used with the device.

pki-lifespan: what is the timespan for this anchor. Does it get replaced at some interval, and if so, by what means is this done?

pki-level-locked: (a Boolean) the level where the EE cert will be found locked by the device, or can levels be added or deleted by the PKI operator without code changes to the device.

pki-breadth: how many different non-expired EE certificates is the PKI designed to manage?

pki-lock-policy: can any EE certificate be used with this trust anchor to sign? Or, is there some kind of policy OID or Subject restriction? Are specific subordinate CAs needed that lead to the EE?

pki-anchor-storage: how is the private key associated with this trust anchor stored? How many people are needed to recover it?

7. Security Considerations

This entire document is about security considerations.

8. IANA Considerations

This document makes no IANA requests.

9. Acknowledgements

Robert Martin of MITRE provided some guidance about citing the SBOM efforts. Carsten Bormann and Sini Ruohomaa provided many editorial suggestions.

10. Changelog

11. References

11.1. Normative References

- [ieee802-1AR]
IEEE Standard, "IEEE 802.1AR Secure Device Identifier",
2018, <<https://standards.ieee.org/ieee/802.1AR/6995/>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S.,
Housley, R., and W. Polk, "Internet X.509 Public Key
Infrastructure Certificate and Certificate Revocation List
(CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008,
<<https://www.rfc-editor.org/rfc/rfc5280>>.

11.2. Informative References

- [3GPP.33.310]
3GPP and J. Anja, "Network Domain Security (NDS);
Authentication Framework (AF)", 30 September 2011,
<[http://www.3gpp.org/ftp/Specs/
archive/33_series/33.310/33310-970.zip](http://www.3gpp.org/ftp/Specs/archive/33_series/33.310/33310-970.zip)>.
- [BedOfNails]
Wikipedia, "Bed of nails tester", 30 December 2024,
<https://en.wikipedia.org/wiki/Bed_of_nails_tester>.
- [CABFORUM] CA/Browser Forum, "CA/Browser Forum Baseline Requirements
for the Issuance and Management of Publicly-Trusted
Certificates, v.1.7.3", October 2020,
<[https://cabforum.org/wp-content/uploads/CA-Browser-Forum-
BR-1.7.3.pdf](https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.7.3.pdf)>.
- [CABForum90day]
CA/Browser Forum, "CA/Browser Forum Latest Baseline
Requirements, version 2.1.5", n.d., <[https://cabforum.org/
working-groups/server/baseline-requirements/
requirements/](https://cabforum.org/working-groups/server/baseline-requirements/requirements/)>.

[cisqsbo] CISQ/Object Management Group, "TOOL-TO-TOOL SOFTWARE BILL OF MATERIALS EXCHANGE", 1 July 2020, <<https://www.it-cisq.org/software-bill-of-materials/index.htm>>.

[ComodoGate]
"Comodo-gate hacker brags about forged certificate exploit", 28 March 2011,
<https://www.theregister.com/2011/03/28/comodo_gate_hacker_breaks_cover/>.

[curl] Stenberg, D., "CURL", 28 May 2025,
<<https://curl.se/download.html>>.

[factoringrsa]
Bernstein, D. J., Chang, Y.-A., Cheng, C.-M., Chou, L.-P., Heninger, N., Lange, T., and N. van. Someren, "Factoring RSA keys from certified smart cards: Coppersmith in the wild", 16 September 2013,
<<https://eprint.iacr.org/2013/599>>.

[fidotechnote]
FIDO Alliance, "FIDO TechNotes: The Truth about Attestation", 19 July 2018, <<https://fidoalliance.org/fido-technotes-the-truth-about-attestation/>>.

[hyundaiaexample]
The Register, "Software developer cracks Hyundai car security with Google search", 17 August 2022,
<https://www.theregister.com/2022/08/17/software_developer_cracks_hyundai_encryption/>.

[I-D.ietf-anima-brski-cloud]
Friel, O., Shekh-Yusef, R., and M. Richardson, "Bootstrapping Remote Secure Key Infrastructure (BRSKI) Cloud Registrar", Work in Progress, Internet-Draft, draft-ietf-anima-brski-cloud-19, 9 September 2025,
<<https://datatracker.ietf.org/doc/html/draft-ietf-anima-brski-cloud-19>>.

[I-D.ietf-anima-constrained-voucher]
Richardson, M., Van der Stok, P., Kampanakis, P., and E. Dijk, "Constrained Bootstrapping Remote Secure Key Infrastructure (cBRSKI)", Work in Progress, Internet-Draft, draft-ietf-anima-constrained-voucher-31, 8 June 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-anima-constrained-voucher-31>>.

- [I-D.ietf-anima-masa-considerations]
Richardson, M. and T. Werner, "Operational Considerations for Voucher infrastructure for BRSKI MASA", Work in Progress, Internet-Draft, draft-ietf-anima-masa-considerations-03, 9 August 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-anima-masa-considerations-03>>.
- [I-D.ietf-iotops-7228bis]
Bormann, C., Ersue, M., Keränen, A., and C. Gomez, "Terminology for Constrained-Node Networks", Work in Progress, Internet-Draft, draft-ietf-iotops-7228bis-10, 6 July 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-iotops-7228bis-10>>.
- [I-D.ietf-iotops-mud-rats]
Birkholz, H., Richardson, M., and P. C. Liu, "MUD-Based RATS Resources Discovery", Work in Progress, Internet-Draft, draft-ietf-iotops-mud-rats-03, 23 July 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-iotops-mud-rats-03>>.
- [I-D.moskowitz-ecdsa-pki]
Moskowitz, R., Birkholz, H., Xia, L., and M. Richardson, "Guide for building an ECC pki", Work in Progress, Internet-Draft, draft-moskowitz-ecdsa-pki-10, 31 January 2021, <<https://datatracker.ietf.org/doc/html/draft-moskowitz-ecdsa-pki-10>>.
- [ISO27001] International Organization for Standardization, "ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection -- Information security management systems -- Requirements", 2022.
- [JTAG] "Joint Test Action Group", 26 August 2020, <<https://en.wikipedia.org/wiki/JTAG>>.
- [keyceremony2]
Wikipedia, "Key Ceremony", 25 June 2026, <https://en.wikipedia.org/wiki/Key_ceremony>.
- [kskceremony]
Verisign, "DNSSEC Practice Statement for the Root Zone ZSK Operator", 18 July 2024, <<https://www.iana.org/dnssec/procedures>>.

[leakedmsikey]

Binarily efiXplorer Team, "Leaked MSI Source Code with Intel OEM Keys: How Does This Affect Industry-wide Software Supply Chain?", 27 May 2025, <<https://www.binarily.io/blog/leaked-msi-source-code-with-intel-oem-keys-how-does-this-affect-industry-wide-software-supply-chain>>.

[measuredboot]

Trusted Computing Group, "TCG PC Client Specific Platform Firmware Profile Specification", 4 December 2023, <<https://trustedcomputinggroup.org/resource/pc-client-specific-platform-firmware-profile-specification/>>.

[ntiasbom] NTIA, "NTIA Software Component Transparency", 1 July 2020, <<https://www.ntia.doc.gov/SoftwareTransparency>>.

[openbmc] Linux Foundation/OpenBMC Group, "Defining a Standard Baseboard Management Controller Firmware Stack", 1 July 2020, <<https://www.openbmc.org/>>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <<https://www.rfc-editor.org/rfc/rfc4086>>.

[RFC4210] Adams, C., Farrell, S., Kause, T., and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", RFC 4210, DOI 10.17487/RFC4210, September 2005, <<https://www.rfc-editor.org/rfc/rfc4210>>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <<https://www.rfc-editor.org/rfc/rfc4949>>.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, DOI 10.17487/RFC5011, September 2007, <<https://www.rfc-editor.org/rfc/rfc5011>>.

[RFC6024] Reddy, R. and C. Wallace, "Trust Anchor Management Requirements", RFC 6024, DOI 10.17487/RFC6024, October 2010, <<https://www.rfc-editor.org/rfc/rfc6024>>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <<https://www.rfc-editor.org/rfc/rfc7030>>.

- [RFC7168] Nazar, I., "The Hyper Text Coffee Pot Control Protocol for Tea Efflux Appliances (HTCPCP-TEA)", RFC 7168, DOI 10.17487/RFC7168, April 2014, <<https://www.rfc-editor.org/rfc/rfc7168>>.
- [RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <<https://www.rfc-editor.org/rfc/rfc8366>>.
- [RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <<https://www.rfc-editor.org/rfc/rfc8520>>.
- [RFC8572] Watsen, K., Farrer, I., and M. Abrahamsson, "Secure Zero Touch Provisioning (SZTP)", RFC 8572, DOI 10.17487/RFC8572, April 2019, <<https://www.rfc-editor.org/rfc/rfc8572>>.
- [RFC8894] Gutmann, P., "Simple Certificate Enrolment Protocol", RFC 8894, DOI 10.17487/RFC8894, September 2020, <<https://www.rfc-editor.org/rfc/rfc8894>>.
- [RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <<https://www.rfc-editor.org/rfc/rfc8995>>.
- [RFC9019] Moran, B., Tschofenig, H., Brown, D., and M. Meriac, "A Firmware Update Architecture for Internet of Things", RFC 9019, DOI 10.17487/RFC9019, April 2021, <<https://www.rfc-editor.org/rfc/rfc9019>>.
- [RFC9140] Aura, T., Sethi, M., and A. Peltonen, "Nimble Out-of-Band Authentication for EAP (EAP-NOOB)", RFC 9140, DOI 10.17487/RFC9140, December 2021, <<https://www.rfc-editor.org/rfc/rfc9140>>.
- [RFC9180] Barnes, R., Bhargavan, K., Lipp, B., and C. Wood, "Hybrid Public Key Encryption", RFC 9180, DOI 10.17487/RFC9180, February 2022, <<https://www.rfc-editor.org/rfc/rfc9180>>.
- [RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <<https://www.rfc-editor.org/rfc/rfc9334>>.

- [RFC9393] Birkholz, H., Fitzgerald-McKay, J., Schmidt, C., and D. Waltermire, "Concise Software Identification Tags", RFC 9393, DOI 10.17487/RFC9393, June 2023, <<https://www.rfc-editor.org/rfc/rfc9393>>.
- [RFC9397] Pei, M., Tschofenig, H., Thaler, D., and D. Wheeler, "Trusted Execution Environment Provisioning (TEEP) Architecture", RFC 9397, DOI 10.17487/RFC9397, July 2023, <<https://www.rfc-editor.org/rfc/rfc9397>>.
- [RFC9641] Watsen, K., "A YANG Data Model for a Truststore", RFC 9641, DOI 10.17487/RFC9641, October 2024, <<https://www.rfc-editor.org/rfc/rfc9641>>.
- [RFC9711] Lundblade, L., Mandyam, G., O'Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", RFC 9711, DOI 10.17487/RFC9711, April 2025, <<https://www.rfc-editor.org/rfc/rfc9711>>.
- [RFC9733] von Oheimb, D., Ed., Fries, S., and H. Brockhaus, "BRSKI with Alternative Enrollment (BRSKI-AE)", RFC 9733, DOI 10.17487/RFC9733, March 2025, <<https://www.rfc-editor.org/rfc/rfc9733>>.
- [rootkeyceremony] Community, "Root Key Ceremony, Cryptography Wiki", 4 April 2020, <https://cryptography.fandom.com/wiki/Root_Key_Ceremony>.
- [shamir79] Shamir, A., "How to share a secret.", 1979, <<http://web.mit.edu/6.857/OldStuff/Fall03/ref/Shamir-HowToShareASecret.pdf>>.
- [TPM20spec] Trusted Computing Group, "TPM 2.0 Library", March 2025, <<https://trustedcomputinggroup.org/resource/tpm-library-specification/>>.
- [TPMEK] Trusted Computing Group, "TCG Credential Profile EK 2.0", 4 December 2024, <https://trustedcomputinggroup.org/wp-content/uploads/TCG-EK-Credential-Profile-for-TPM-Family-2.0-Level-0-Version-2.6_pub.pdf>.
- [_3GPP.51.011] 3GPP and P. L. Thanh, "Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface", 15 June 2005, <http://www.3gpp.org/ftp/Specs/archive/51_series/51.011/51011-4f0.zip>.

Author's Address

Michael Richardson
Sandelman Software Works
Email: mcr+ietf@sandelman.ca