

Network Working Group
Internet-Draft
Updates: 6740, 6741, 6742, 6748 (if approved)
Intended status: Experimental
Expires: 9 November 2026

S. N. Bhatti
University of St Andrews, UK
8 May 2026

ILNP addressing using Preference values
draft-bhatti-ilnp-preference-00

Abstract

The Identifier Locator Network Protocol (ILNP) for IPv6 is described in Experimental RFCs 6740-6744. This document clarifies how addressing in ILNP makes use of Preference values with Locator (L64) and Node Identifier (NID) values. This includes the way that Preference values are used for forming Identifier-Locator Vector (I-LV) values, and selecting I-LVs for use. This document updates RFC6740, RFC6741, RFC6742, RFC6748.

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-bhatti-ilnp-preference/>.

Discussion of this document takes place on the Network Network Working Group mailing list (<mailto:saleem@st-andrews.ac.uk>).

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 9 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.

Table of Contents

1. Introduction	3
1.1. Purpose	4
1.2. Rationale	4
2. Conventions and Definitions	5
2.1. Definitions from other documents	5
2.2. New definitions	5
3. Updates to previous RFC documents	6
3.1. RFC6740 and RFC6741	6
3.2. RFC6742	6
3.3. RFC6748	6
4. Purpose of the Preference value	6
4.1. Presence of Preference values	7
4.2. Discovery of Preference values	7
4.3. Use of Preference values in constructing I-LV values	7
5. General method for generating I-LV values from L64 and NID values	8
5.1. I-LV construction	8
5.2. Source I-LV construction	9
5.3. Destination I-LV construction	10
5.4. Selection of I-LV values	10
5.5. Changes to the lists of L64 and NID values	11
5.6. Locally-defined I-LV values	12
5.7. Preference values for locally-defined L64, NID, or I-LV values	12
6. Security Considerations	12
7. Privacy Considerations	12
8. IANA Considerations	12
9. References	13
9.1. Normative References	13
9.2. Informative References	14
Acknowledgements	15
Author's Address	15

1. Introduction

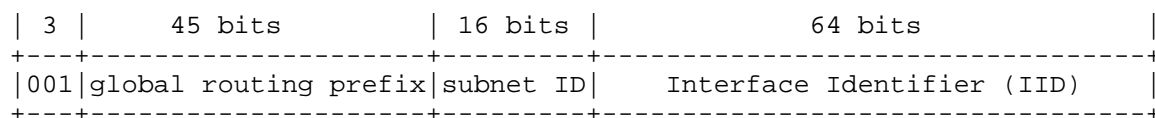
The Identifier Locator Network Protocol (ILNP) redefines the IP addressing architecture by use of new addressing data-types [RFC6740] [RFC6741]. The ILNP addressing data-types considered in this document are:

- * `_Locator (L64)_`: A 64-bit value (8 bytes, in network / canonical byte order) that is a label for a network.
- * `_Node Identifier (NID)_`: A 64-bit value (8 bytes, in network / canonical byte order) that is a label for a node.
- * `_Identifier-Locator Vector (I-LV)_`: The 128-bit concatenation of a single L64 value and single NID value for use in the IPv6 packet header in the source address and destination address fields.

L64 and NID values each have an associated `_Preference_`, a 16-bit value in the range 0-65535 (i.e. a 16-bit, unsigned integer), with lower values indicating higher preference.

From an engineering perspective, for backwards compatibility, the data-types are realised and used within the context of IPv6 [RFC6741]. An ILNP packet will use the address fields in an IPv6 packet header [RFC8200] to carry I-LV values constructed from L64 and NID values based on the Preference values, even though the Preference values themselves are not sent in the packet. The relationships between addressing data-types for IPv6 and ILNP are summarised in the diagram of Figure 1.

IPv6 (RFC8200 / STD86) general IPv6 global address format:



ILNP (RFC6741) Identifier Locator Vector (I-LV):

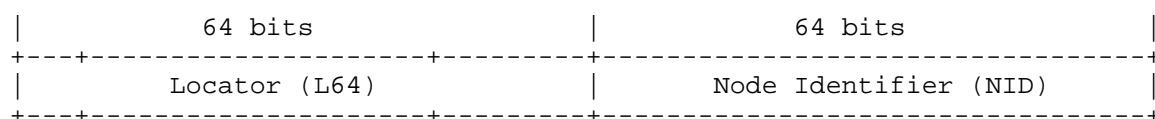


Figure 1: An IPv6 address (RFC8200 / STD86) and an ILNP Identifier-Locator Vector (RFC6741), as used in the address fields of the IPv6 packet header.

An ILNP packet is distinguished from an IPv6 packet by the presence of a `_Nonce Header_`, the name used for the IPv6 Destination Option Header specified in [RFC6744] for use only with ILNP. Examining only the address fields in an IPv6 packet is not sufficient to determine whether the packet is an ILNP packet, the Nonce Header must be present [draft-bhatti-ilnp-nonce].

A more detailed description of the ILNP architecture and its relationship to IPv6 can be found in [RFC6740] and [RFC6741].

L64 and NID values also have corresponding DNS Resource Record (RR) definitions in [RFC6742], which also include a Preference value as part of each L64 or NID RR.

L64 values with Preference values are also used in ILNP Locator Update (LU) messages as defined in [RFC6743].

1.1. Purpose

This document clarifies the way that Preference values are used in ILNP in the following ways:

1. The semantics of the Preference value as used with L64, NID, and I-LV values.
2. How I-LV values are formed using Preference values when a node has multiple L64 and NID values.
3. How Preference values impact the choice of I-LV at the transmitting node.

ILNP is defined for use with IPv6 and for IPv4, but all references in this document to ILNP are for IPv6 only. It is possible to apply the methods described in this document to ILNP for IPv4 (with the use of the L32 data-type from [RFC6740] [RFC6742] in place of the L64 data-type), but that exercise is outside the scope of this document.

1.2. Rationale

An ILNP node can use multiple L64 values and multiple NID values simultaneously, and separate Preference values are associated with each L64 and NID. A 128-bit I-LV is created from a single L64 concatenated with a single NID value. When L64 and NID values are used to create an I-LV for transmission, the Preference value affects the selection process for I-LV values at the transmitting node, subject to local policy [RFC6740] [RFC6741].

The use of Preference values in addressing in ILNP needs clarification, as it affects the use of L64 and NID values to form I-LVs. The use of local policy to determine how Preference values are to be used remains, and the definition of such local policy is beyond the scope of this document. However, in the absence of such local policy, for the ILNP addressing architecture in general, this document clarifies how Preference values are to be used.

So, a clear description of the use of the Preference value is needed to understand how it should be used at the ILNP node.

ILNP can also be used by unmodified IPv6 applications, and that usage is described in [draft-bhatti-ilnp-ip6-apps].

Use of DHCPv6 [RFC6939] specifically for ILNP is not yet defined, and is outside the scope of this document.

2. Conventions and Definitions

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.

2.1. Definitions from other documents

The following definitions are taken from [RFC6740]:

- * Locator, L64
- * Node Identifier, NID
- * Identifier-Locator Vector, I-LV
- * I-L Communications Cache, ILCC
- * Source I-LV
- * Destination I-LV

2.2. New definitions

This document defines two terms:

L64 Preference A Preference value for a L64 value. This MUST be used in place of the term _Locator Preference Indicator (LPI)_ in [RFC6740] [RFC6741] [RFC6748].

NID Preference A Preference value for a NID value. This was not explicitly defined or discussed in [RFC6740] or [RFC6741], but it is clear from [RFC6742] that NID values also have a Preference.

3. Updates to previous RFC documents

RFC documents that are updated by this document are:

- * RFC6740 "Identifier-Locator Network Protocol (ILNP) Architectural Description" [RFC6740].
- * RFC6741 "Identifier-Locator Network Protocol (ILNP) Engineering Considerations" [RFC6741].
- * RFC6742 "DNS Resource Records for the Identifier-Locator Network Protocol (ILNP)" [RFC6742].
- * RFC6748 "Optional Advanced Deployment Scenarios for the Identifier-Locator Network Protocol (ILNP)" [RFC6748].

3.1. RFC6740 and RFC6741

[RFC6740] discusses use of a Locator Preference Indicator (LPI) for L64 values, but does not discuss Preference values for NID values. [RFC6741] does not discuss Preference values. However, neither document states how Preference values are to be used by a node.

3.2. RFC6742

[RFC6742] defines Preference values for use in L64 and NID Resource Records (RRs) for DNS. However, it does not state how Preference values are to be used by a node.

3.3. RFC6748

[RFC6748] mentions the use of the LPI. That document also mentions the use of local policy to determine how Preference values would be used. However, it does not give any more detail.

4. Purpose of the Preference value

As ILNP nodes might make use of multiple L64 and NID values, Preference values SHOULD be given with each L64 or NID value so that a node can make a selection of which L64 and NID values to use.

4.1. Presence of Preference values

The Preference value is included for ILNP nodes as follows, and its presence can be summarised in the following three rules:

1. A Preference value SHOULD be given for every L64 (L64 Preference) value or NID (NID Preference) value in use by a node.
2. A Preference value SHOULD be given for every L64 (L64 Preference) value or NID (NID Preference) value when they are used for configuration purposes, e.g. for local end-system configurations, or for application-specific configurations.
3. When a L64 Preference value or NID Preference value is not known or given it MUST be considered to be zero.

From the rules above, the "MUST" in rule 3 allows flexibility for a Preference value not to be given explicitly (hence the "SHOULD" in rules 1 and 2).

Preference values are not carried in the IPv6 header, and are not present in an ILNP packet apart from the case of the LU message. The LU message carries a L64 Preference value for each L64 in the payload as defined in [RFC6743], but not as part of the IPv6 header.

There might be local policy or application-specific usage that impacts the use of the Preference values, and the definition of such usage is out of scope for this document.

Preference values can be used on an ILNP node for unmodified IPv6 applications (applications that are not ILNP-aware) to gain some of the benefits of ILNP, and such usage is described in [draft-bhatti-ilnp-ip6-apps].

4.2. Discovery of Preference values

Preference values can be learned from DNS [RFC6742] or from local configuration files. When learned from DNS, they are included in ILNP DNS Resource Records (RRs), i.e. L64 or NID RRs as defined in [RFC6742]. When learned from local configuration files, Preference values SHOULD be given with the respective definitions of values for L64s, NIDs, or I-LVs.

4.3. Use of Preference values in constructing I-LV values

I-LV values must be constructed so that they can be used in place of IPv6 address values in the IPv6 packet header.

1. When a node has multiple L64 values and NID values defined for it to use, it must construct I-LV values for use as a Source I-LV, to be carried as the Source Address in the IPv6 header.
2. When a node discovers multiple L64 and NID values defined for use for a remote node, it must construct I-LV values to use as a Destination I-LV, to be carried as the Destination Address in the IPv6 header.

In both cases, the Preference value is used by the transmitting node to create an I-LV value.

An ILNP node MAY construct I-LV values directly from the available L64 and NID values based on application-specific requirements or based on local policy -- please see Section 5.6. Otherwise, this document (specifically Section 5) describes how Source I-LV values and Destination I-LV values MUST be constructed from L64 and NID values using their respective Preference values.

5. General method for generating I-LV values from L64 and NID values

5.1. I-LV construction

The general method for generating I-LV values from L64 and NID values uses the definitions below and the simple algorithm defined in Figure 2. This method can be used for generating Source I-LV values and Destination I-LV values.

- * Let $\{L_l\}$ be a sequence of l L64 values each with a L64 Preference value, $l \geq 1$.
 - $\{L_l\}$ MUST be ordered with `_increasing_` L64 Preference; and
 - $\{L_l\}$ MUST NOT have duplicates for the 64-bit L64 values.
- * Let $\{N_n\}$ be a sequence of n NID values each with a NID Preference value, $n \geq 1$.
 - $\{N_n\}$ MUST be ordered with `_increasing_` NID Preference; and
 - $\{N_n\}$ MUST NOT have duplicates for the 64-bit NID values.

The relative ordering of elements in each list $\{L_l\}$ or $\{N_n\}$ which have the same Preference value is considered to be either subject to local policy or to be application-specific.

- * Let $\{A_a\}$ be a sequence of a I-LV values, initially $a = 0$, i.e. the sequence is empty.

- * Let l64 be a single L64 64-bit value (8 bytes, network / canonical byte order).
- * Let nid be a single NID 64-bit value (8 bytes, network / canonical byte order).
- * Let x::y be the concatenation of x and y.

Then the algorithm for generating {A_a} is given in Figure 2:

```
foreach nid in {N_n}:  
  foreach l64 in {L_l}:  
    A_a.append(l64::nid)
```

Figure 2: General algorithm for generating a list of I-LV, the ordered I-LV sequence, {A_a}, from a list of L64 values, {L_l}, and a list of NID values, {N_n}. The lists {L_l} and {N_n} each MUST be ordered with increasing Preference value for their elements and MUST NOT contain, respectively, duplicate L64 or NID values.

The sequence {A_a} is now the ordered I-LV sequence, and contains a list of 128-bit I-LV values, each of which can be used as an IPv6 address, with the most preferred I-LV values at the start of the list.

The lists {L_l} and {N_n} are both maintained by the operating system in an ILNP node through the function of the ILCC. When the lists {L_l} and {N_n} are complete, or if either list changes, the mechanism described in this section can be invoked to (re-)generate the ordered I-LV sequence, {A_a}. Please see Section 5.5 for more details.

The precise details of how an ILNP-aware application can make use of {L_l} and {N_n} directly is likely to be application-specific or subject to local policy, so is outside the scope of this document, but please see Section 5.4.

For IPv6 applications running on an ILNP node, the use of {A_a} is described in [draft-bhatti-ilnp-ip6-apps].

5.2. Source I-LV construction

For Source I-LV values, L64 and NID values could be learned from a number of sources.

As L64 values are IPv6 address prefixes, ILNP allows L64 values to be discovered via an IPv6 Router Advertisement (RA) [RFC4861]. ILNP NID values can be generated dynamically using any of the mechanisms defined for IPv6, e.g. for privacy by using [RFC8981] or [HB2025], or for verifiable addresses as in [RFC3972]. In such cases, Preference values will be zero for L64 and NID values unless local policy defines otherwise. Such automatic configuration could be used for default / bootstrap configurations, and allows backwards compatibility with IPv6.

If only a L64 value is specified locally, the NID can still take a generated NID value using any mechanism already defined for generating IPv6 IID values. If only a NID value is specified locally, the L64 can still take a value of an IPv6 address prefix learned from an IPv6 RA. For example, if a node has a single L64, L_1 , learned from an IPv6 RA, and a single NID, N_1 , defined locally, then it will have a single source I-LV, $L_1:N_1$. If Preference values are not specified, they should be considered to have the value of zero.

However, it is possible for a node to have a fixed I-LV defined for it through local configuration, just as it is possible for a node to have a fixed IPv6 address. Please see also Section 5.6 and Section 5.7.

5.3. Destination I-LV construction

For remote nodes, L64 and NID values for constructing Destination I-LV values will typically be learned from the DNS, e.g. from DNS L64 and NID RRs (please see [draft-bhatti-ilnp-textual-representations]). L64 and NID RRs have Preference values [RFC6742]. However, it is also possible for L64 and NID values to be configured through some locally-defined mechanism, e.g. operating system files, or application-specific configuration. Please see also Section 5.6 and Section 5.7.

5.4. Selection of I-LV values

The ordering mechanism means that the first I-LV, A_1 , in $\{A_a\}$ is always the one that has the combination of first the lowest NID Preference and then the lowest L64 Preference, i.e. the most preferred NID value and most preferred L64 value.

If {A_a} contains more than one value, the process of selecting which value to use is out of scope for this document. The choice could be subject to local policy, or could be an application-specific choice. In the case of it being application-specific, it will depend on the API available, and if the application is ILNP-aware or not (please see [draft-bhatti-ilnp-ip6-apps]).

If {L_l} and {N_n} are directly accessible on an ILNP node, then an ILNP-aware application can:

- * construct I-LV values directly if it has access to {L_l} and {N_n}; or
- * choose the I-LV to use directly from {A_a}; and
- * use Preference values in making such selections as required.

5.5. Changes to the lists of L64 and NID values

The L64 list, {L_l}, and NID list, {N_n}, could change as new L64 and NID values become available or existing L64 or NID values can no longer be used. A (non-exhaustive) list of reasons the lists could change is:

1. L64 and NID values were discovered from DNS, and the Time-to-Live (TTL) for the respective DNS RR expires.
2. NID values were generated using privacy mechanisms and so have a limited lifetime based on the privacy mechanism used.
3. Routing changes mean that a L64 value discovered from an IPv6 RA expires and so cannot be used any longer.
4. An IPv6 RA appears that shows a new address prefix is available to use as a L64 value.
5. A LU message is received from a remote node with one or more new L64 values for that remote node, and previous L64 values for that node are not present in the LU message so are no longer available for use for that remote node.
6. A local policy makes changes to the available L64 or NID values for the node or for a remote node.

Overall, if {L_l} or {N_n} change, the ordered I-LV sequence, {A_a}, could change. However, as explained in Section 5.1, the ILCC maintains {L_l} and {N_n}, so {A_a} can be (re-)generated as needed.

5.6. Locally-defined I-LV values

It is possible to define complete, fixed I-LV values locally, e.g. in /etc/hosts (please see [draft-bhatti-ilnp-textual-representations]). If such local I-LV definitions exist, whether Source I-LV (for this node) or Destination I-LV (for a remote node), they MUST NOT be decomposed to separate L64 and NID values for the formation of other I-LV values for inclusion in the ordered I-LV sequence, {A_a}. I-LV values MUST be ordered according to the NID Preference value and L64 Preference value for that I-LV only for inclusion in {A_a}. This allows for a default local configuration if required, and also allows backwards compatibility with IPv6.

5.7. Preference values for locally-defined L64, NID, or I-LV values

Overall, locally-defined values of L64s, NIDs, or I-LVs are subject to local policy. Where no such local policy exists, locally-defined L64, NID, or I-LV values SHOULD have explicit Preference values defined also, and those Preference values SHOULD be non-zero so that other dynamically generated or learned values with lower Preference values can be used as required. However, the "SHOULD" in the above text is specifically included so as not to constrain local policy or application-specific configuration.

6. Security Considerations

There are no new security considerations.

The existing security mechanisms already defined for ILNP remain unchanged (please see Section 9 of [RFC6740], Section 11 of [RFC6741]).

7. Privacy Considerations

There are no new privacy considerations.

The existing identity privacy and location privacy properties already defined for ILNP remain unchanged (please see Section 10 of [RFC6740], Section 12 of [RFC6741], Section 7 of [RFC6748]). NID values can also be generated using privacy mechanisms such as [RFC8981]. ILNP can also use enhanced identity privacy and location privacy mechanisms which remain backwards compatible with IPv6, such as described in [BHY2021], [HB2024], and [HB2025].

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

- [draft-bhatti-ilnp-ip6-apps]
Bhatti, S. N., Haywood, G. T., Yanagida, R., and R. W. Grimes, "ILNP usage by IPv6 applications", 8 May 2026. A related draft that is being produced in parallel.
- [draft-bhatti-ilnp-nonce]
Bhatti, S. N., "Use of the ILNP Nonce Destination Option Header", 8 May 2026. A related draft that is being produced in parallel.
- [draft-bhatti-ilnp-textual-representations]
Bhatti, S. N., Haywood, G. T., Yanagida, R., and R. W. Grimes, "ILNP textual representations", 8 May 2026. A related draft that is being produced in parallel.
- [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>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<https://www.rfc-editor.org/rfc/rfc3972>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/rfc/rfc4861>>.
- [RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <<https://www.rfc-editor.org/rfc/rfc6740>>.
- [RFC6741] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Engineering Considerations", RFC 6741, DOI 10.17487/RFC6741, November 2012, <<https://www.rfc-editor.org/rfc/rfc6741>>.
- [RFC6742] Atkinson, RJ., Bhatti, SN., and S. Rose, "DNS Resource Records for the Identifier-Locator Network Protocol (ILNP)", RFC 6742, DOI 10.17487/RFC6742, November 2012, <<https://www.rfc-editor.org/rfc/rfc6742>>.

- [RFC6743] Atkinson, RJ. and SN. Bhatti, "ICMP Locator Update Message for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)", RFC 6743, DOI 10.17487/RFC6743, November 2012, <<https://www.rfc-editor.org/rfc/rfc6743>>.
- [RFC6744] Atkinson, RJ. and SN. Bhatti, "IPv6 Nonce Destination Option for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)", RFC 6744, DOI 10.17487/RFC6744, November 2012, <<https://www.rfc-editor.org/rfc/rfc6744>>.
- [RFC6748] Atkinson, RJ. and SN. Bhatti, "Optional Advanced Deployment Scenarios for the Identifier-Locator Network Protocol (ILNP)", RFC 6748, DOI 10.17487/RFC6748, November 2012, <<https://www.rfc-editor.org/rfc/rfc6748>>.
- [RFC6939] Halwasia, G., Bhandari, S., and W. Dec, "Client Link-Layer Address Option in DHCPv6", RFC 6939, DOI 10.17487/RFC6939, May 2013, <<https://www.rfc-editor.org/rfc/rfc6939>>.
- [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>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/rfc/rfc8200>>.
- [RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <<https://www.rfc-editor.org/rfc/rfc8981>>.

9.2. Informative References

- [BHY2021] Bhatti, S., Haywood, G., and R. Yanagida, "End-to-End Privacy for Identity & Location with IP", IEEE, 2021 IEEE 29th International Conference on Network Protocols (ICNP) pp. 1-6, DOI 10.1109/icnp52444.2021.9651909, November 2021, <<https://doi.org/10.1109/icnp52444.2021.9651909>>.
- [HB2024] Haywood, G. and S. Bhatti, "Defence against Side-Channel Attacks for Encrypted Network Communication Using Multiple Paths", MDPI AG, Cryptography vol. 8, no. 2, pp. 22, DOI 10.3390/cryptography8020022, May 2024, <<https://doi.org/10.3390/cryptography8020022>>.

[HB2025] Haywood, G. and S. Bhatti, "Ephemeral Node Identifiers for Enhanced Flow Privacy", MDPI AG, Future Internet vol. 17, no. 5, pp. 196, DOI 10.3390/fi17050196, April 2025, <<https://doi.org/10.3390/fi17050196>>.

Acknowledgements

The author is grateful to the many members of the IETF community for their feedback on ILNP during IETF meetings, and to the IETF NOC Team who made possible testing and experiments for ILNP during those meetings and the IETF Hackathon events.

Gregor Haywood, Ryo Yanagida, and Rodney Grimes provided feedback on the original text for this document which helped to improve clarity.

This work was partly supported by the _ICANN Grant Program_.

Author's Address

Saleem N. Bhatti
University of St Andrews, UK
Email: saleem@st-andrews.ac.uk