Internet Engineering Task Force (IETF)
Request for Comments: 8928
Updates: 8505
Category: Standards Track
ISSN: 2070-1721

P. Thubert, Ed.
Cisco
B. Sarikaya
M. Sethi
Ericsson
R. Struik
Struik Security Consultancy

November 2020

Address-Protected Neighbor Discovery for Low-Power and Lossy Networks

Abstract

This document updates the IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery (ND) protocol defined in RFCs 6775 and 8505. The new extension is called Address-Protected Neighbor Discovery (AP-ND), and it protects the owner of an address against address theft and impersonation attacks in a Low-Power and Lossy Network (LLN). Nodes supporting this extension compute a cryptographic identifier (Crypto-ID), and use it with one or more of their Registered Addresses. The Crypto-ID identifies the owner of the Registered Address and can be used to provide proof of ownership of the Registered Addresses. Once an address is registered with the Crypto-ID and a proof of ownership is provided, only the owner of that address can modify the registration information, thereby enforcing Source Address Validation.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8928.

described in the Simplified BSD License.

Table of Contents

1. Introduction

Neighbor Discovery optimizations for 6LoWPAN networks (aka 6LoWPAN ND) [RFC6775] adapts the original IPv6 Neighbor Discovery protocols defined in [RFC4861] and [RFC4862] for constrained Low-Power and Lossy Networks (LLNs). In particular, 6LoWPAN ND introduces a unicast host Address Registration mechanism that reduces the use of multicast compared to the Duplicate Address Detection (DAD) mechanism

defined in IPv6 ND. 6LoWPAN ND defines a new Address Registration Option (ARO) that is carried in the unicast Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages exchanged between a 6LoWPAN Node (6LN) and a 6LoWPAN Router (6LR). It also defines the Duplicate Address Request (DAR) and Duplicate Address Confirmation (DAC) messages between the 6LR and the 6LoWPAN Border Router (6LBR). In LLNs, the 6LBR is the central repository of all the Registered Addresses in its domain.

The registration mechanism in "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)" [RFC6775] prevents the use of an address if that address is already registered in the subnet (first come, first served). In order to validate address ownership, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery" [RFC8505] defines a Registration Ownership Verifier (ROVR) field. [RFC8505] enables a 6LR and 6LBR to validate the association between the Registered Address of a node and its ROVR. The ROVR can be derived from the link-layer address of the device (using the 64-bit Extended Unique Identifier (EUI-64) address format specified by IEEE). However, the EUI-64 can be spoofed; therefore, any node connected to the subnet and aware of a registered-address-to-ROVR mapping could effectively fake the ROVR. This would allow an attacker to steal the address and redirect traffic for that address. [RFC8505] defines an Extended Address Registration Option (EARO) that transports alternate forms of ROVRs and is a prerequisite for this specification.

In this specification, a 6LN generates a cryptographic identifier (Crypto-ID) and places it in the ROVR field during the registration of one (or more) of its addresses with the 6LR(s). Proof of ownership of the Crypto-ID is passed with the first registration exchange to a new 6LR and enforced at the 6LR. The 6LR validates ownership of the Crypto-ID before it creates any new registration state or changes existing information.

The protected address registration protocol proposed in this document provides the same conceptual benefit as Source Address Validation Improvement (SAVI) [RFC7039] in that only the owner of an IPv6 address may source packets with that address. As opposed to [RFC7039], which relies on snooping protocols, the protection provided by this document is based on a state that is installed and maintained in the network by the owner of the address. With this specification, a 6LN may use a 6LR for forwarding an IPv6 packet if and only if it has registered the address used as the source of the packet with that 6LR.

With the 6LoWPAN adaptation layer in [RFC4944] and [RFC6282], a 6LN can obtain better compression for an IPv6 address with an Interface ID (IID) that is derived from a Layer 2 (L2) address. Such compression is incompatible with "SEcure Neighbor Discovery (SEND") [RFC3971] and "Cryptographically Generated Addresses (CGAs)" [RFC3972], since they derive the IID from cryptographic keys. This specification, on the other hand, separates the IID generation from cryptographic computations and can enable better compression.

2. Terminology

2.1. 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.

2.2. Background

The reader may get additional context for this specification from the following references:

* "SEcure Neighbor Discovery (SEND)" [RFC3971],

* "Cryptographically Generated Addresses (CGA)" [RFC3972],

* "Neighbor Discovery for IP version 6 (IPv6)" [RFC4861] ,

* "IPv6 Stateless Address Autoconfiguration" [RFC4862], and

* "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals" [RFC4919].

2.3. Abbreviations

This document uses the following abbreviations:

6BBR:	6LoWPAN Backbone Router
6LBR:	6LoWPAN Border Router
6LN:	6LoWPAN Node
6LR:	6LoWPAN Router
AP-ND:	Address-Protected Neighbor Discovery
CGA:	Cryptographically Generated Address
DAD:	Duplicate Address Detection
EARO:	Extended Address Registration Option
ECC:	Elliptic Curve Cryptography
ECDH:	Elliptic Curve Diffie-Hellman
ECDSA:	Elliptic Curve Digital Signature Algorithm
EDAC:	Extended Duplicate Address Confirmation
EDAR:	Extended Duplicate Address Request
CIPO:	Crypto-ID Parameters Option
LLN:	Low-Power and Lossy Network
NA:	Neighbor Advertisement
ND:	Neighbor Discovery
NDP:	Neighbor Discovery Protocol
NDPSO:	Neighbor Discovery Protocol Signature Option
NS:	Neighbor Solicitation
ROVR:	Registration Ownership Verifier
RA:	Router Advertisement
RS:	Router Solicitation
RSAO:	RSA Signature Option
SHA:	Secure Hash Algorithm
SLAAC:	Stateless Address Autoconfiguration

TID: Transaction ID

3. Updating RFC 8505

Section 5.3 of [RFC8505] introduces the ROVR that is used to detect and reject duplicate registrations in the DAD process. The ROVR is a generic object that is designed for both backward compatibility and the capability to introduce new computation methods in the future. Using a Crypto-ID per this specification is the RECOMMENDED method. Section 7.5 discusses collisions when heterogeneous methods to compute the ROVR field coexist inside a network.

This specification introduces a new identifier called a Crypto-ID that is transported in the ROVR field and used to indirectly prove the ownership of an address that is being registered by means of [RFC8505]. The Crypto-ID is derived from a cryptographic public key and additional parameters.

The overall mechanism requires the support of Elliptic Curve Cryptography (ECC) and a hash function as detailed in Section 6.2. To enable the verification of the proof, the Registering Node needs to supply certain parameters including a nonce and a signature that will demonstrate that the node possesses the private key corresponding to the public key used to build the Crypto-ID.

The elliptic curves and the hash functions listed in Table 1 in Section 8.2 can be used with this specification; more may be added in the future to the corresponding IANA registry. The cryptographic algorithms used (including the curve and the representation conventions) are signaled by the Crypto-Type field in a new IPv6 ND Crypto-ID Parameters Option (CIPO) (see Section 4.3) that contains the parameters that are necessary for address validation. A new NDP Signature Option (Section 4.4) is also specified in this document to carry the resulting signature. A Nonce Option [RFC3971] is added in the NA(EARO) that is used to request the validation, and all three options are needed in the NS(EARO) that provides the validation.

4. New Fields and Options

4.1. New Crypto-ID

The Crypto-ID is transported in the ROVR field of the EARO and the Extended Duplicate Address Request (EDAR) message and is associated with the Registered Address at the 6LR and the 6LBR. The ownership of a Crypto-ID can be demonstrated by cryptographic mechanisms, and by association, the ownership of the Registered Address can be ascertained.

A node in possession of the necessary cryptographic primitives SHOULD use Crypto-ID by default as ROVR in its registrations. Whether a ROVR is a Crypto-ID is indicated by a new "C" flag in the EARO of the NS(EARO) message.

The Crypto-ID is derived from the public key and a modifier as follows:

1. The hash function used internally by the signature scheme and indicated by the Crypto-Type (see Table 1 in Section 8.2) is applied to the CIPO. Note that all the reserved and padding bits MUST be set to zero.

2. The leftmost bits of the resulting hash, up to the desired size, are used as the Crypto-ID.

At the time of this writing, a minimal size for the Crypto-ID of 128 bits is RECOMMENDED unless backward compatibility is needed [RFC8505] (in which case it is at least 64 bits). The size of the Crypto-ID is likely to increase in the future.

4.2. Updated EARO

This specification updates the EARO to enable the use of the ROVR field to transport the Crypto-ID. The resulting format is as follows:

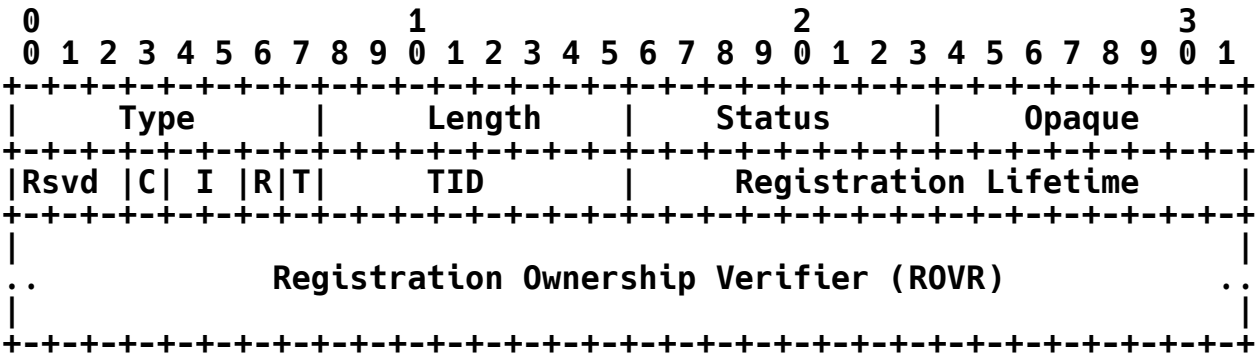


Figure 1: Enhanced Address Registration Option

Type: 33

Length: Defined in [RFC8505] and copied in the "EARO Length" field in the associated CIPO.

Status: Defined in [RFC8505].

Opaque: Defined in [RFC8505].

Rsvd (Reserved): 3-bit unsigned integer. It MUST be set to zero by the sender and MUST be ignored by the receiver.

C: This "C" flag is set to indicate that the ROVR field contains a Crypto-ID and that the 6LN MAY be challenged for ownership as specified in this document.

I, R, T: Defined in [RFC8505].

TID and Registration Lifetime: Defined in [RFC8505].

Registration Ownership Verifier (ROVR): When the "C" flag is set, this field contains a Crypto-ID.

This specification uses the status codes "Validation Requested" and "Validation Failed", which are defined in [RFC8505].

This specification does not define any new status codes.

4.3. Crypto-ID Parameters Option

This specification defines the CIPO. The CIPO carries the parameters used to form a Crypto-ID.

In order to provide cryptographic agility [BCP201], this specification supports different elliptic-curve-based signature schemes, indicated by a Crypto-Type field:

* The ECDSA256 signature scheme, which uses ECDSA with the NIST P-256 curve [FIPS186-4] and the hash function SHA-256 [RFC6234] internally, MUST be supported by all implementations.

* The Ed25519 signature scheme, which uses the Pure Edwards-Curve Digital Signature Algorithm (PureEdDSA) [RFC8032] with the twisted Edwards curve Edwards25519 [RFC7748] and the hash function SHA-512 [RFC6234] internally, MAY be supported as an alternative.

* The ECDSA25519 signature scheme, which uses ECDSA [FIPS186-4] with the Weierstrass curve Wei25519 (see Appendix B.4) and the hash function SHA-256 [RFC6234] internally, MAY also be supported.

This specification uses signature schemes that target similar cryptographic strength but rely on different curves, hash functions, signature algorithms, and/or representation conventions. Future specification may extend this to different cryptographic algorithms and key sizes, e.g., to provide better security properties or a simpler implementation.

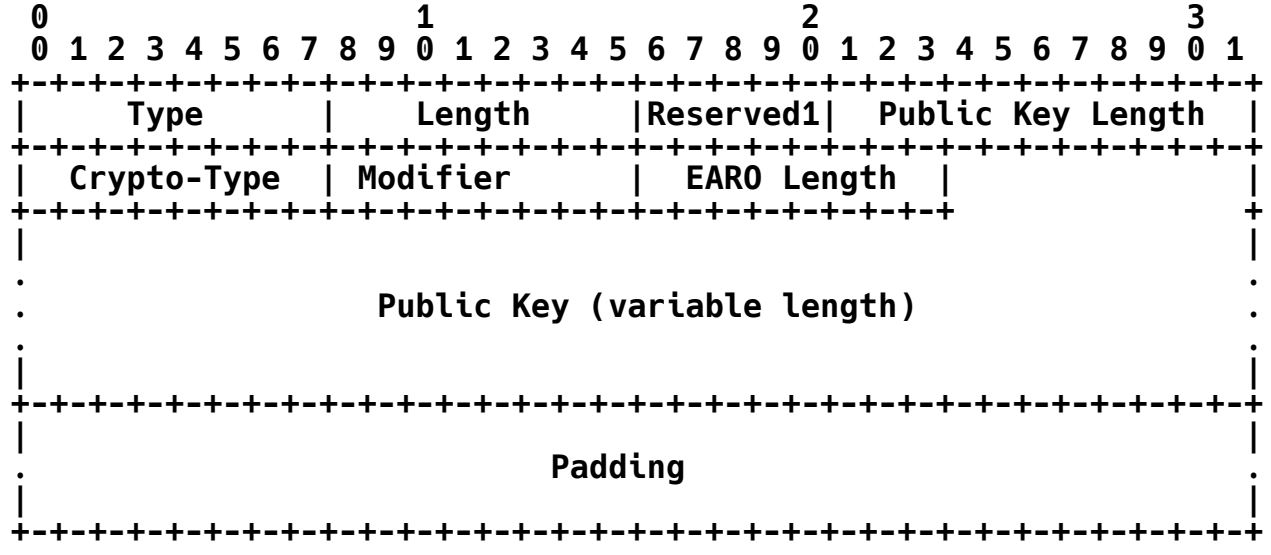


Figure 2: Crypto-ID Parameters Option

Type: 8-bit unsigned integer. IANA has assigned value 39; see Table 2.

Length: 8-bit unsigned integer. The length of the option in units of 8 octets.

Reserved1: 5-bit unsigned integer. It MUST be set to zero by the sender and MUST be ignored by the receiver.

Public Key Length: 11-bit unsigned integer. The length of the Public Key field in bytes. The actual length depends on the Crypto-Type value and how the public key is represented. The valid values with this document are provided in Table 1.

Crypto-Type: 8-bit unsigned integer. The type of cryptographic algorithm used in calculation of the Crypto-ID indexed by IANA in the "Crypto-Types" subregistry in the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry (see Section 8.2).

Modifier: 8-bit unsigned integer. Set to an arbitrary value by the creator of the Crypto-ID. The role of the modifier is to enable the formation of multiple Crypto-IDs from the same key pair. This reduces the traceability and, thus, improves the privacy of a constrained node without requiring many key pairs.

EARO Length: 8-bit unsigned integer. The option length of the EARO that contains the Crypto-ID associated with the CIPO.

Public Key: A variable-length field; the size is indicated in the Public Key Length field.

Padding: A variable-length field that completes the Public Key field to align to the next 8-byte boundary. It MUST be set to zero by the sender and MUST be ignored by the receiver.

The implementation of multiple hash functions in a constrained device may consume excessive amounts of program memory. This specification enables the use of the same hash function SHA-256 [RFC6234] for two of the three supported ECC-based signature schemes. Some code factorization is also possible for the ECC computation itself.

[CURVE-REPR] provides information on how to represent Montgomery curves and (twisted) Edwards curves as curves in short-Weierstrass form, and it illustrates how this can be used to implement elliptic curve computations using existing implementations that already provide, e.g., ECDSA and ECDH using NIST [FIPS186-4] prime curves. For more details on representation conventions, refer to Appendix B.

4.4. NDP Signature Option

This specification defines the NDP Signature Option (NDPSO). The NDPSO carries the signature that proves the ownership of the Crypto-ID and validates the address being registered. The format of the NDPSO is illustrated in Figure 3.

As opposed to the RSA Signature Option (RSAO) defined in Section 5.2 of SEND [RFC3971], the NDPSO does not have a key hash field.

Instead, the leftmost 128 bits of the ROVR field in the EARO are used as hash to retrieve the CIPO that contains the key material used for signature verification, left-padded if needed.

Another difference is that the NDPSO signs a fixed set of fields as opposed to all options that appear prior to it in the ND message that bears the signature. This allows a CIPO that the 6LR already received to be omitted, at the expense of the capability to add arbitrary options that would be signed with an RSAO.

An ND message that carries an NDPSO MUST have one and only one EARO. The EARO MUST contain a Crypto-ID in the ROVR field, and the Crypto-ID MUST be associated with the key pair used for the digital signature in the NDPSO.

The CIPO may be present in the same message as the NDPSO. If it is not present, it can be found in an abstract table that was created by a previous message and indexed by the hash.

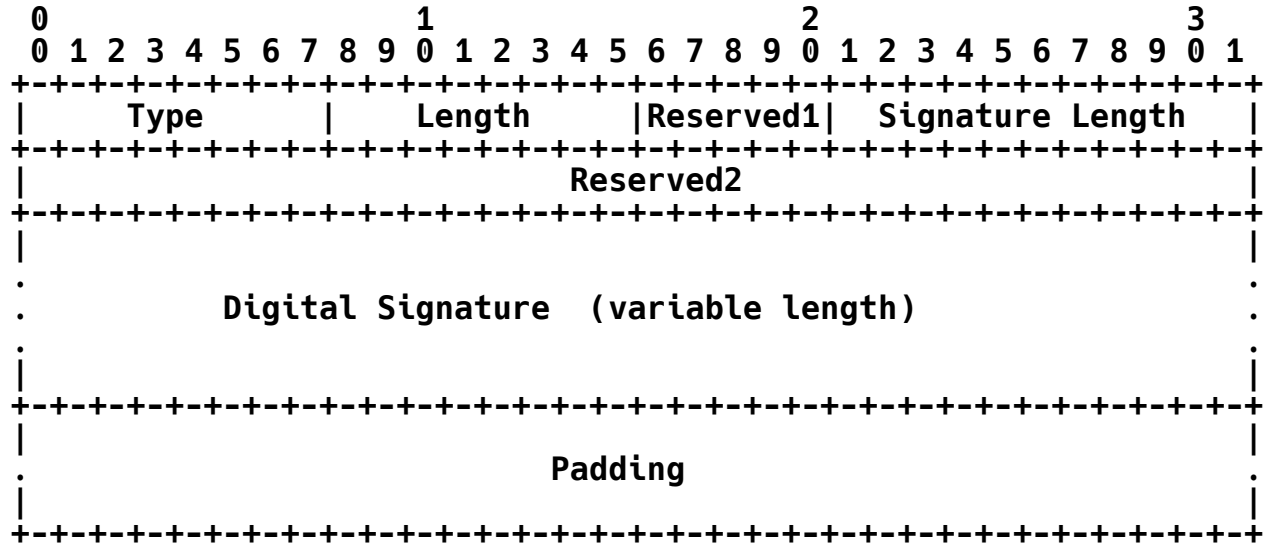


Figure 3: NDP Signature Option

Type: IANA has assigned value 40; see Table 2.

Length: 8-bit unsigned integer. The length of the option in units of 8 octets.

Reserved1: 5-bit unsigned integer. It MUST be set to zero by the sender and MUST be ignored by the receiver.

Digital Signature Length: 11-bit unsigned integer. The length of the Digital Signature field in bytes.

Reserved2: 32-bit unsigned integer. It MUST be set to zero by the sender and MUST be ignored by the receiver.

Digital Signature: A variable-length field containing the digital signature. The length and computation of the digital signature both depend on the Crypto-Type, which is found in the associated

CIPO; see Appendix B. For the values of the Crypto-Type defined in this specification, and for future values of the Crypto-Type unless specified otherwise, the signature is computed as detailed in Section 6.2.

Padding: A variable-length field completing the Digital Signature field to align to the next 8-byte boundary. It MUST be set to zero by the sender and MUST be ignored by the receiver.

4.5. Extensions to the Capability Indication Option

This specification defines one new capability bit in the 6LoWPAN Capability Indication Option (6CIO), as defined by [RFC7400], for use by the 6LR and 6LBR in IPv6 ND RA messages.

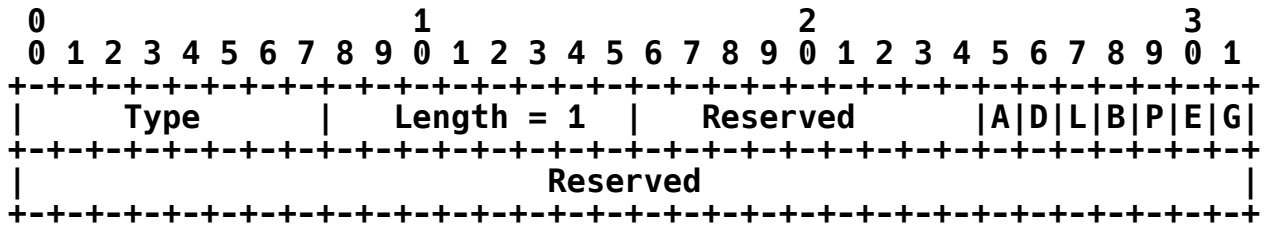


Figure 4: New Capability Bit in the 6CIO

New Option Field:

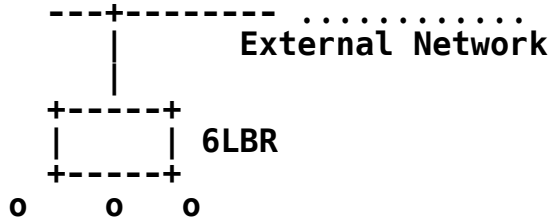
A: 1-bit flag. Set to indicate that AP-ND is globally activated in the network.

The "A" flag is set by the 6LBR that serves the network and is propagated by the 6LRs. It is typically turned on when all 6LRs are migrated to this specification.

5. Protocol Scope

The scope of the protocol specified here is a 6LoWPAN LLN, typically a stub network connected to a larger IP network via a border router called a 6LBR per [RFC6775]. A 6LBR has sufficient capability to satisfy the needs of DAD.

The 6LBR maintains registration state for all devices in its attached LLN. Together with the first-hop router (the 6LR), the 6LBR assures uniqueness and grants ownership of an IPv6 address before it can be used in the LLN. This is in contrast to a traditional network that relies on IPv6 address autoconfiguration [RFC4862], where there is no guarantee of ownership from the network, and each IPv6 Neighbor Discovery packet must be individually secured [RFC3971].



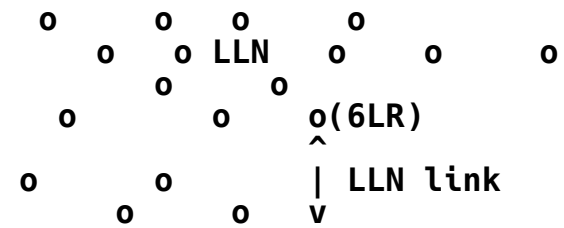


Figure 5: Basic Configuration

In a mesh network, the 6LR is directly connected to the host device. This specification mandates that the peer-wise L2 security is deployed so that all the packets from a particular host are protected. The 6LR may be multiple hops away from the 6LBR. Packets are routed between the 6LR and the 6LBR via other 6LRs.

This specification mandates that all the LLN links between the 6LR and the 6LBR are protected so that a packet that was validated by the first 6LR can be safely routed by other on-path 6LRs to the 6LBR.

6. Protocol Flows

The 6LR/6LBR ensures first come, first served by storing the ROVR associated to the address being registered upon the first registration and rejecting a registration with a different ROVR value. A 6LN can claim any address as long as it is the first to make that claim. After a successful registration, the 6LN becomes the owner of the Registered Address, and the address is bound to the ROVR value in the 6LR/6LBR registry.

This specification protects the ownership of the address at the first hop (the edge). Its use in a network is signaled by the "A" flag in the 6CIO. The flag is set by the 6LBR and propagated unchanged by the 6LRs. Once every node in the network is upgraded to support this specification, the "A" flag can be set to turn the protection on globally.

The 6LN places a cryptographic identifier, the Crypto-ID, in the ROVR that is associated with the address at the first registration, enabling the 6LR to later challenge it to verify that it is the original Registering Node. The challenge may happen at any time at the discretion of the 6LR and the 6LBR. A valid registration in the 6LR or the 6LBR MUST NOT be altered until the challenge is complete.

When the "A" flag in a subnet is set, the 6LR MUST challenge the 6LN before it creates a Binding with the "C" flag set in the EARO. The 6LR MUST also challenge the 6LN when a new registration attempts to change a parameter of an already validated Binding for that 6LN, for instance, its Source link-layer address. Such verification protects against an attacker that attempts to steal the address of an honest node.

The 6LR MUST indicate to the 6LBR that it performed a successful validation by setting a status code of 5 ("Validation Requested") in

the EDAR. Upon a subsequent EDAR from a new 6LR with a status code that is not 5 for a validated Binding, the 6LBR MUST indicate to the new 6LR that it needs to challenge the 6LN using a status code of 5 in the Extended Duplicate Address Confirmation (EDAC).

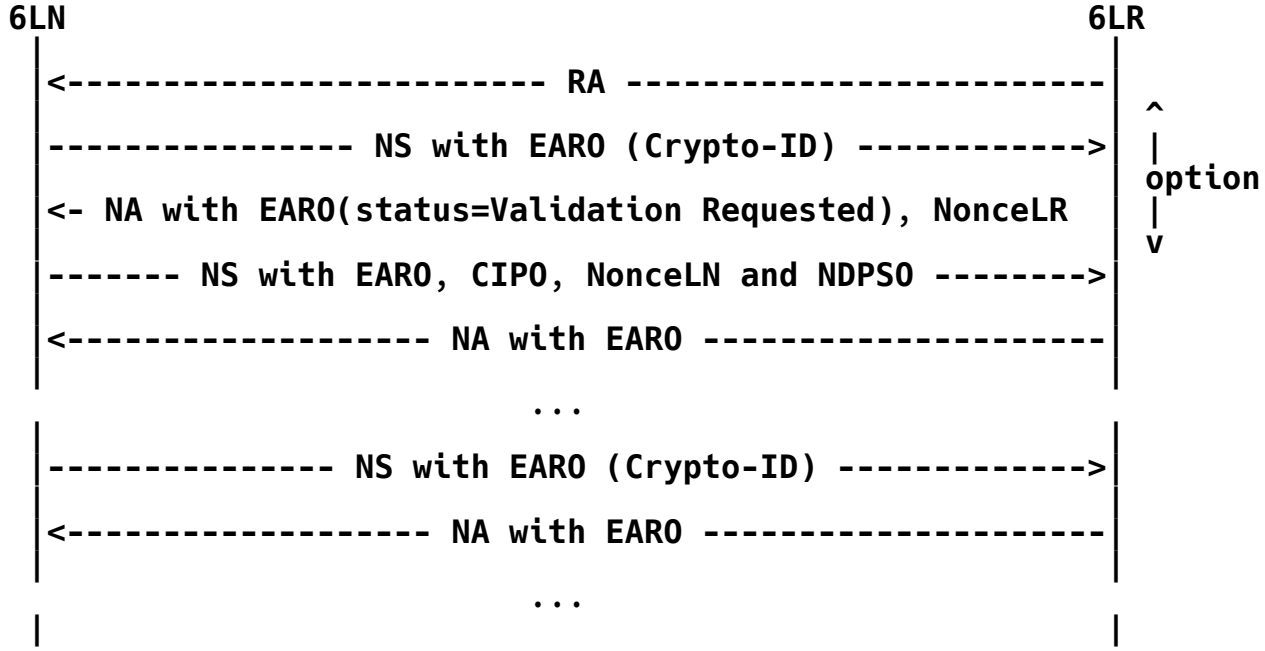
The 6LR MUST challenge the 6LN when the 6LBR signals to do so, which is done with an EDAC message with a status code of 5. The EDAC is echoed by the 6LR in the NA(EARO) back to the Registering Node. The 6LR SHOULD also challenge all its attached 6LNs at the time the 6LBR turns the "A" flag on in the 6CIO in orders to detect an issue immediately.

If the 6LR does not support the Crypto-Type, it MUST reply with an EARO status code of 10 "Validation Failed" without a challenge. In that case, the 6LN may try another Crypto-Type until it falls back to Crypto-Type 0, which MUST be supported by all 6LRs.

A node may use more than one IPv6 address at the same time. The separation of the address and the cryptographic material avoids the need for the constrained device to compute multiple keys for multiple addresses. The 6LN MAY use the same Crypto-ID to prove the ownership of multiple IPv6 addresses. The 6LN MAY also derive multiple Crypto-IDs from the same key pair by changing the modifier.

6.1. First Exchange with a 6LR

A 6LN registers to a 6LR that is one hop away from it with the "C" flag set in the EARO, indicating that the ROVR field contains a Crypto-ID. The Target Address in the NS message indicates the IPv6 address that the 6LN is trying to register [RFC8505]. The on-link (local) protocol interactions are shown in Figure 6. If the 6LR does not have a state with the 6LN that is consistent with the NS(EARO), then it replies with a challenge NA(EARO, status=Validation Requested) that contains a Nonce Option (shown as NonceLR in Figure 6).



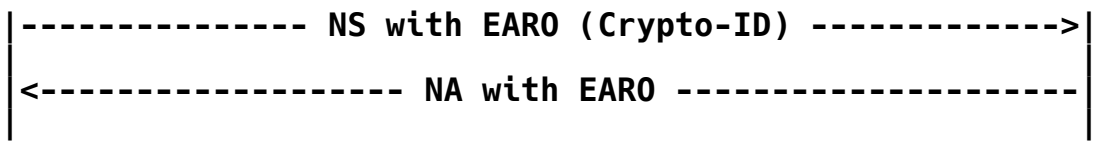


Figure 6: On-Link Protocol Operation

The Nonce Option contains a nonce value that, to the extent possible for the implementation, was never used before. This specification inherits the idea from [RFC3971] that the nonce is a random value. Ideally, an implementation uses an unpredictable cryptographically random value [BCP106]. But that may be impractical in some LLN scenarios with resource-constrained devices.

Alternatively, the device may use an always-incrementing value saved in the same stable storage as the key, so they are lost together, and start at a best-effort random value as either the nonce value or a component to its computation.

The 6LN replies to the challenge with an NS(EARO) that includes the Nonce Option (shown as NonceLN in Figure 6), the CIPO (Section 4.3), and the NDPSO containing the signature. Both nonces are included in the signed material. This provides a "contributory behavior" that results in better security even when the nonces of one party are not generated as specified.

The 6LR MUST store the information associated with a Crypto-ID on the first NS exchange where it appears in a fashion that the CIPO parameters can be retrieved from the Crypto-ID alone.

The steps for the registration to the 6LR are as follows:

Upon the first exchange with a 6LR, a 6LN will be challenged to prove ownership of the Crypto-ID and the Target Address being registered in the Neighbor Solicitation message. When a 6LR receives an NS(EARO) registration with a new Crypto-ID as a ROVR, and unless the registration is rejected for another reason, it MUST challenge by responding with an NA(EARO) with a status code of "Validation Requested".

Upon receiving a first NA(EARO) with a status code of "Validation Requested" from a 6LR, the Registering Node SHOULD retry its registration with a CIPO (Section 4.3) that contains all the necessary material for building the Crypto-ID, the NonceLN that it generated, and the NDP Signature Option (Section 4.4) that proves its ownership of the Crypto-ID and intent of registering the Target Address. In subsequent revalidation with the same 6LR, the 6LN MAY try to omit the CIPO to save bandwidth, with the expectation that the 6LR saved it. If the validation fails and it gets challenged again, then it SHOULD add the CIPO again.

In order to validate the ownership, the 6LR performs the same steps as the 6LN and rebuilds the Crypto-ID based on the parameters in the CIPO. If the rebuilt Crypto-ID matches the ROVR, the 6LN also verifies the signature contained in the NDPSO. At that point, if the signature in the NDPSO can be verified, then the validation succeeds.

Otherwise, the validation fails.

If the 6LR fails to validate the signed NS(EARO), it responds with a status code of "Validation Failed". After receiving an NA(EARO) with a status code of "Validation Failed", the Registering Node SHOULD try an alternate Crypto-Type; even if Crypto-Type 0 fails, it may try to register a different address in the NS message.

6.2. NDPSO Generation and Verification

The signature generated by the 6LN to provide proof of ownership of the private key is carried in the NDPSO. It is generated by the 6LN in a fashion that depends on the Crypto-Type (see Table 1 in Section 8.2) chosen by the 6LN as follows:

* Form the message to be signed, by concatenating the following byte-strings in the order listed:

    1. The 128-bit Message Type tag [RFC3972] (in network byte order). For this specification, the tag is given in Section 8.1. (The tag value has been generated by the editor of this specification on <https://www.random.org>.)

    2. The CIPO.

    3. The 16-byte Target Address (in network byte order) sent in the NS message. It is the address that the 6LN is registering with the 6LR and 6LBR.

    4. The NonceLR received from the 6LR (in network byte order) in the NA message. The nonce is at least 6 bytes long as defined in [RFC3971].

    5. The NonceLN sent from the 6LN (in network byte order). The nonce is at least 6 bytes long as defined in [RFC3971].

    6. The 1-byte option length of the EARO containing the Crypto-ID.

* Apply the signature algorithm specified by the Crypto-Type using the private key.

Upon receiving the NDPSO and CIPO options, the 6LR first checks that the EARO Length in the CIPO matches the length of the EARO. If so, it regenerates the Crypto-ID based on the CIPO to make sure that the leftmost bits up to the size of the ROVR match.

If, and only if, the check is successful, it tries to verify the signature in the NDPSO using the following steps:

* Form the message to be verified, by concatenating the following byte-strings in the order listed:

    1. The 128-bit Message Type tag given in Section 8.1 (in network byte order).

    2. The CIPO.

3. The 16-byte Target Address (in network byte order) received in the NS message. It is the address that the 6LN is registering with the 6LR and 6LBR.

4. The NonceLR sent in the NA message. The nonce is at least 6 bytes long as defined in [RFC3971].

5. The NonceLN received from the 6LN (in network byte order) in the NS message. The nonce is at least 6 bytes long as defined in [RFC3971].

6. The 1-byte EARO Length received in the CIPO.

* Verify the signature on this message with the public key in the CIPO and the locally computed values using the signature algorithm specified by the Crypto-Type. If the verification succeeds, the 6LR propagates the information to the 6LBR using an EDAR/EDAC flow.

* Due to the first-come, first-served nature of the registration, if the address is not registered to the 6LBR, then flow succeeds and both the 6LR and 6LBR add the state information about the Crypto-ID and Target Address being registered to their respective abstract databases.

6.3. Multi-Hop Operation

A new 6LN that joins the network autoconfigures an address and performs an initial registration to a neighboring 6LR with an NS message that carries an EARO [RFC8505].

In a multi-hop 6LoWPAN, the registration with Crypto-ID is propagated to 6LBR as shown in Figure 7, which illustrates the registration flow all the way to a 6LoWPAN Backbone Router (6BBR) [RFC8929].

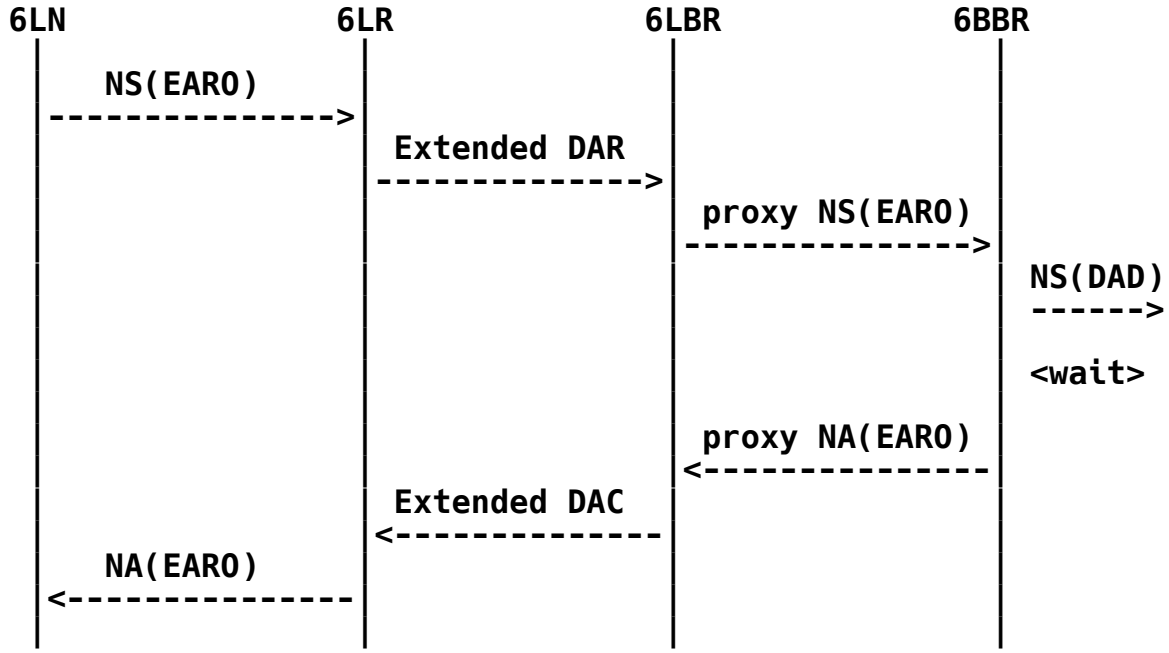


Figure 7: (Re-)Registration Flow

The 6LR and the 6LBR communicate using ICMPv6 EDAR and EDAC messages [RFC8505] as shown in Figure 7. This specification extends EDAR/EDAC messages to carry cryptographically generated ROVR.

The assumption is that the 6LR and the 6LBR maintain a security association to authenticate and protect the integrity of the EDAR and EDAC messages, so there is no need to propagate the proof of ownership to the 6LBR. The 6LBR implicitly trusts that the 6LR performs the verification when the 6LBR requires it, and if there is no further exchange from the 6LR to remove the state, the verification succeeded.

7. Security Considerations

7.1. Brown Field

Only 6LRs that are upgraded to this specification are capable of challenging a registration and avoiding an attack. In a brown (mixed) network, an attacker may attach to a legacy 6LR and fool the 6LBR. So even if the "A" flag could be set at any time to test the protocol operation, the security will only be effective when all the 6LRs are upgraded.

7.2. Threats Identified in RFC 3971

Observations regarding the following threats to the local network in [RFC3971] also apply to this specification.

Neighbor Solicitation/Advertisement Spoofing: Threats in Section 9.2.1 of [RFC3971] apply. AP-ND counters the threats on NS(EARO) messages by requiring that the NDPSO and CIPO be present in these solicitations.

Duplicate Address Detection DoS Attack: Inside the LLN, duplicate addresses are sorted out using the ROVR. A different ROVR for the same Registered Address entails a rejection of the second registration [RFC8505]. DADs coming from the backbone network are not forwarded over the LLN to provide some protection against DoS attacks inside the resource-constrained part of the network. However, the EARO is present in the NS/NA messages exchanged over the backbone network. This protects against misinterpreting node movement as a duplication and enables the Backbone Routers to determine which subnet has the most recent registration [RFC8505] and is thus the best candidate to validate the registration [RFC8929].

Router Solicitation and Advertisement Attacks: This specification does not change the protection of RS and RA, which can still be protected by SEND.

Replay Attacks: Nonces should never repeat but they do not need to be unpredictable for secure operation. Using nonces (NonceLR and NonceLN) generated by both the 6LR and 6LN ensures a contributory

behavior that provides an efficient protection against replay attacks of the challenge/response flow. The quality of the protection by a random nonce depends on the random number generator.

Neighbor Discovery DoS Attack: A rogue node that can access the L2 network may form many addresses and register them using AP-ND. The perimeter of the attack is all the 6LRs in range of the attacker. The 6LR MUST protect itself against overflows and reject excessive registration with a status code of 2 "Neighbor Cache Full". This effectively blocks another (honest) 6LN from registering to the same 6LR, but the 6LN may register to other 6LRs that are in its range but not in that of the attacker.

7.3. Related to 6LoWPAN ND

The threats and mitigations discussed in 6LoWPAN ND [RFC6775] [RFC8505] also apply here, in particular, denial-of-service (DoS) attacks against the registry at the 6LR or 6LBR.

Secure ND [RFC3971] forces the IPv6 address to be cryptographic since it integrates the CGA as the IID in the IPv6 address. In contrast, this specification saves about 1 KB in every NS/NA message. Also, this specification separates the cryptographic identifier from the registered IPv6 address so that a node can have more than one IPv6 address protected by the same cryptographic identifier.

With this specification, the 6LN can freely form its IPv6 address(es) in any fashion, thereby enabling either 6LoWPAN compression for IPv6 addresses that are derived from L2 addresses or temporary addresses that cannot be compressed, e.g., formed pseudorandomly and released in relatively short cycles for privacy reasons [RFC8064][RFC8065].

This specification provides added protection for addresses that are obtained following due procedure [RFC8505] but does not constrain the way the addresses are formed or the number of addresses that are used in parallel by a same entity. An attacker may still perform a DoS attack against the registry at the 6LR or 6LBR or attempt to deplete the pool of available addresses at L2 or L3.

7.4. Compromised 6LR

This specification distributes the challenge and its validation at the edge of the network, between the 6LN and its 6LR. This protects against DoS attacks targeted at that central 6LBR. This also saves back-and-forth exchanges across a potentially large and constrained network.

The downside is that the 6LBR needs to trust the 6LR to perform the checking adequately, and the communication between the 6LR and the 6LBR must be protected to avoid tampering with the result of the validation.

If a 6LR is compromised, and provided that it knows the ROVR field used by the real owner of the address, the 6LR may pretend that the owner has moved, is now attached to it, and has successfully passed

the Crypto-ID validation. The 6LR may then attract and inject traffic at will on behalf of that address, or let an attacker take ownership of the address.

7.5. ROVR Collisions

A collision of ROVRs (i.e., the Crypto-ID in this specification) is possible, but it is a rare event. Assuming that the hash used for calculating the Crypto-ID is a well-behaved cryptographic hash, and, thus, random collisions are the only ones possible, if n = 2^(k) is the maximum number of hash values (i.e., a k-bit hash) and p is the number of nodes, then (assuming one Crypto-ID per node) the formula 1 - e^(-p^(2)/(2n)) provides an approximation of the probability that there is at least one collision (birthday paradox).

If the Crypto-ID is 64 bits (the least possible size allowed), the chance of a collision is 0.01% for a network of 66 million nodes. Moreover, the collision is only relevant when this happens within one stub network (6LBR). In the case of such a collision, an honest node might accidentally claim the Registered Address of another legitimate node (with the same Crypto-ID). To prevent such rare events, it is RECOMMENDED that nodes do not derive the address being registered from the ROVR.

7.6. Implementation Attacks

The signature schemes referenced in this specification comply with NIST [FIPS186-4] or Crypto Forum Research Group (CFRG) standards [RFC8032] and offer strong algorithmic security at roughly a 128-bit security level. These signature schemes use elliptic curves that either were specifically designed with exception-free and constant-time arithmetic in mind [RFC7748] or have extensive implementation experience of resistance to timing attacks [FIPS186-4].

However, careless implementations of the signing operations could nevertheless leak information on private keys. For example, there are micro-architectural side channel attacks that implementors should be aware of [breaking-ed25519]. Implementors should be particularly aware that a secure implementation of Ed25519 requires a protected implementation of the hash function SHA-512, whereas this is not required with implementations of the hash function SHA-256 used with ECDSA256 and ECDSA25519.

7.7. Cross-Algorithm and Cross-Protocol Attacks

The key pair used in this specification can be self-generated, and the public key does not need to be exchanged, e.g., through certificates, with a third party before it is used.

New key pairs can be formed for new registrations if the node desires. However, the same private key MUST NOT be reused with more than one instantiation of the signature scheme in this specification. Also, the same private key MUST NOT be used for anything other than computing NDPSO signatures per this specification.

ECDSA shall be used strictly as specified in [FIPS186-4]. In

particular, each signing operation of ECDSA MUST use randomly generated ephemeral private keys and MUST NOT reuse the ephemeral private key k across signing operations. This precludes the use of deterministic ECDSA without a random input for the determination of k, which is deemed dangerous for the intended applications this document aims to serve.

7.8. Public Key Validation

Public keys contained in the CIPO field (which are used for signature verification) shall be verified to be correctly formed, by checking that this public key is indeed a point of the elliptic curve indicated by the Crypto-Type and that this point does have the proper order.

For points used with the signature scheme Ed25519, one MUST check that this point is not in the small subgroup (see Appendix B.1 of [CURVE-REPR]); for points used with the signature scheme ECDSA (i.e., both ECDSA256 and ECDSA25519), one MUST check that the point has the same order as the base point of the curve in question. This is commonly called "full public key validation" (again, see Appendix B.1 of [CURVE-REPR]).

7.9. Correlating Registrations

The ROVR field in the EARO introduced in [RFC8505] extends the EUI-64 field of the ARO defined in [RFC6775]. One of the drawbacks of using an EUI-64 as ROVR is that an attacker that is aware of the registrations can correlate traffic for the same 6LN across multiple addresses. Section 3 of [RFC8505] indicates that the ROVR and the address being registered are decoupled. A 6LN may use the same ROVR for multiple registrations or a different ROVR per registration, and the IID must not be derived from the ROVR. In theory, different 6LNs could use the same ROVR as long as they do not attempt to register the same address.

The modifier used in the computation of the Crypto-ID enables a 6LN to build different Crypto-IDs for different addresses with the same key pair. Using that facility improves the privacy of the 6LN at the expense of storage in the 6LR, which will need to store multiple CIPOs that contain the same public key. Note that if an attacker gains access to the 6LR, then the modifier alone does not provide protection, and the 6LN would need to generate different key pairs and link-layer addresses in an attempt to obfuscate its multiple ownership.

8. IANA Considerations

8.1. CGA Message Type

This document defines a new 128-bit CGA Extension Type Tag under the "CGA Extension Type Tags" subregistry of the Cryptographically Generated Addresses (CGA) Message Type Name Space created by [RFC3972].

Tag: 0x8701 55c8 0cca dd32 6ab7 e415 f148 84d0.

8.2. Crypto-Type Subregistry

IANA has created the "Crypto-Types" subregistry in the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry. The registry is indexed by an integer in the interval 0..255 and contains an elliptic curve, a hash function, a signature algorithm, representation conventions, public key size, and signature size, as shown in Table 1, which together specify a signature scheme. Detailed explanations are provided in Appendix B.

The following Crypto-Type values are defined in this document:

Crypto-Type Value	0 (ECDSA256)	1 (Ed25519)	2 (ECDSA25519)
Elliptic Curve	NIST P-256 [FIPS186-4]	Curve25519 [RFC7748]	Curve25519 [RFC7748]
Hash Function	SHA-256 [RFC6234]	SHA-512 [RFC6234]	SHA-256 [RFC6234]
Signature Algorithm	ECDSA [FIPS186-4]	Ed25519 [RFC8032]	ECDSA [FIPS186-4]
Representation Conventions	Weierstrass, (un)compressed, MSB/msb-order, [SEC1]	Edwards, compressed, LSB/lsb-order, [RFC8032]	Weierstrass, (un)compressed, MSB/msb-order, [CURVE-REPR]
Public Key Size	33/65 bytes (compressed/ uncompressed)	32 bytes (compressed)	33/65 bytes (compressed/ uncompressed)
Signature Size	64 bytes	64 bytes	64 bytes
Reference	RFC 8928	RFC 8928	RFC 8928

Table 1: Crypto-Types

New Crypto-Type values providing similar or better security may be defined in the future.

Assignment of values for new Crypto-Type MUST be done through IANA with either "Specification Required" or "IESG Approval" as defined in BCP 26 [RFC8126].

8.3. IPv6 ND Option Types

This document registers two new ND option types under the subregistry "IPv6 Neighbor Discovery Option Formats":

Description	Type	Reference

Crypto-ID Parameters Option (CIPO)	39	RFC 8928
NDP Signature Option (NDPSO)	40	RFC 8928

Table 2: New ND Options

8.4. New 6LoWPAN Capability Bit

IANA has made an addition to the subregistry for "6LoWPAN Capability Bits" created for [RFC7400] as follows:

Bit	Description	Reference
9	AP-ND Enabled (1 bit)	RFC 8928

Table 3: New 6LoWPAN Capability Bit

9. References

9.1. Normative References

[FIPS186-4] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", FIPS 186-4, DOI 10.6028/NIST.FIPS.186-4, July 2013, <https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-4.pdf>.

[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>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <https://www.rfc-editor.org/info/rfc3971>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC7400] Bormann, C., "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 7400, DOI 10.17487/RFC7400, November

2014, <https://www.rfc-editor.org/info/rfc7400>.

[RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/info/rfc7748>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[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>.

[RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <https://www.rfc-editor.org/info/rfc8505>.

[SEC1] Standards for Efficient Cryptography, "SEC 1: Elliptic Curve Cryptography", Version 2, May 2009, <https://www.secg.org/sec1-v2.pdf>.

9.2. Informative References

[BCP106] 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/info/rfc4086>.

[BCP201] Housley, R., "Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms", BCP 201, RFC 7696, DOI 10.17487/RFC7696, November 2015, <https://www.rfc-editor.org/info/rfc7696>.

[breaking-ed25519] Samwel, N., Batina, L., Bertoni, G., Daemen, J., and R. Susella, "Breaking Ed25519 in WolfSSL", Topics in Cryptology - CT-RSA, pp. 1-20, March 2018, <https://link.springer.com/chapter/10.1007/978-3-319-76953-0_1>.

[CURVE-REPR] Struik, R., "Alternative Elliptic Curve Representations", Work in Progress, Internet-Draft, draft-ietf-lwig-curve-representations-14, 15 November 2020, <https://tools.ietf.org/html/draft-ietf-lwig-curve-representations-14>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <https://www.rfc-editor.org/info/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/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <https://www.rfc-editor.org/info/rfc4919>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC7039] Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, Ed., "Source Address Validation Improvement (SAVI) Framework", RFC 7039, DOI 10.17487/RFC7039, October 2013, <https://www.rfc-editor.org/info/rfc7039>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <https://www.rfc-editor.org/info/rfc7217>.

[RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <https://www.rfc-editor.org/info/rfc8064>.

[RFC8065] Thaler, D., "Privacy Considerations for IPv6 Adaptation-Layer Mechanisms", RFC 8065, DOI 10.17487/RFC8065, February 2017, <https://www.rfc-editor.org/info/rfc8065>.

[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>.

[RFC8929] Thubert, P., Ed., Perkins, C.E., and E. Levy-Abegnoli, "IPv6 Backbone Router", RFC 8929, DOI 10.17487/RFC8929, November 2020, <https://www.rfc-editor.org/info/rfc8929>.

Appendix A. Requirements Addressed in This Document

In this section, the requirements of a secure Neighbor Discovery protocol for LLNs are stated.

* The protocol MUST be based on the Neighbor Discovery Optimization for the LLN protocol defined in [RFC6775]. RFC 6775 utilizes optimizations such as host-initiated interactions for sleeping resource-constrained hosts and the elimination of multicast address resolution.

* New options to be added to Neighbor Solicitation messages MUST lead to small packet sizes, especially compared with existing protocols such as SEND. Smaller packet sizes facilitate low-power transmission by resource-constrained nodes on lossy links.

* The registration mechanism SHOULD be extensible to other LLN links and not be limited to IEEE 802.15.4 only. LLN links for which a 6lo "IPv6 over foo" specification exist, as well as low-power Wi-Fi, SHOULD be supported.

* As part of this protocol, a mechanism to compute a unique identifier should be provided with the capability to form a Link Local Address that SHOULD be unique at least within the LLN connected to a 6LBR.

* The Address Registration Option used in the ND registration SHOULD be extended to carry the relevant forms of the unique identifier.

* The Neighbor Discovery should specify the formation of a site-local address that follows the security recommendations from [RFC7217].

Appendix B. Representation Conventions

B.1. Signature Schemes

The signature scheme ECDSA256 corresponding to Crypto-Type 0 is ECDSA, as specified in [FIPS186-4], instantiated with the NIST prime curve P-256, as specified in Appendix D.1.2 of [FIPS186-4], and the hash function SHA-256, as specified in [RFC6234], where points of this NIST curve are represented as points of a short-Weierstrass curve (see [FIPS186-4]) and are encoded as octet strings in most-significant-bit first (msb) and most-significant-byte first (MSB) order. The signature itself consists of two integers (r and s), which are each encoded as fixed-size octet strings in MSB and msb order. For further details, see [FIPS186-4] for ECDSA, see Appendix B.3 for the encoding of public keys, and see Appendix B.2 for signature encoding.

The signature scheme Ed25519 corresponding to Crypto-Type 1 is EdDSA, as specified in [RFC8032], instantiated with the Montgomery curve Curve25519, as specified in [RFC7748], and the hash function SHA-512, as specified in [RFC6234], where points of this Montgomery curve are represented as points of the corresponding twisted Edwards curve Edwards25519 (see Appendix B.4) and are encoded as octet strings in least-significant-bit first (lsb) and least-significant-byte first (LSB) order. The signature itself consists of a bit string that

encodes a point of this twisted Edwards curve, in compressed format, and an integer encoded in LSB and lsb order. For details on EdDSA and the encoding of public keys and signatures, see the specification of pure Ed25519 in [RFC8032].

The signature scheme ECDSA25519 corresponding to Crypto-Type 2 is ECDSA, as specified in [FIPS186-4], instantiated with the Montgomery curve Curve25519, as specified in [RFC7748], and the hash function SHA-256, as specified in [RFC6234], where points of this Montgomery curve are represented as points of the corresponding short-Weierstrass curve Wei25519 (see Appendix B.4) and are encoded as octet strings in MSB and msb order. The signature itself consists of a bit string that encodes two integers (r and s), which are each encoded as fixed-size octet strings in MSB and msb order. For further details, see [FIPS186-4] for ECDSA, see Appendix B.3 for the encoding of public keys, and see Appendix B.2 for signature encoding.

B.2. Representation of ECDSA Signatures

With ECDSA, each signature is an ordered pair (r, s) of integers [FIPS186-4], where each integer is represented as a 32-octet string according to the FieldElement-to-OctetString conversion rules in [SEC1] and where the ordered pair of integers is represented as the right concatenation of these representation values (thereby resulting in a 64-octet string). The inverse operation checks that the signature is a 64-octet string and represents the left-side and right-side halves of this string (each a 32-octet string) as the integers r and s, respectively, using the OctetString-to-FieldElement conversion rules in [SEC1]. In both cases, the field with these conversion rules is the set of integers modulo n, where n is the (prime) order of the base point of the curve in question. (For elliptic curve nomenclature, see Appendix B.1 of [CURVE-REPR].)

B.3. Representation of Public Keys Used with ECDSA

ECDSA is specified to be used with elliptic curves in short-Weierstrass form. Each point of such a curve is represented as an octet string using the Elliptic-Curve-Point-to-Octet-String conversion rules in [SEC1], where point compression may be enabled (which is indicated by the leftmost octet of this representation). The inverse operation converts an octet string to a point of this curve using the Octet-String-to-Elliptic-Curve-Point conversion rules in [SEC1], whereby the point is rejected if this is the so-called point at infinity. (This is the case if the input to this inverse operation is an octet string of length 1.)

B.4. Alternative Representations of Curve25519

The elliptic curve Curve25519, as specified in [RFC7748], is a so-called Montgomery curve. Each point of this curve can also be represented as a point of a twisted Edwards curve or as a point of an elliptic curve in short-Weierstrass form, via a coordinate transformation (a so-called isomorphic mapping). The parameters of the Montgomery curve and the corresponding isomorphic curves in twisted Edwards curve and short-Weierstrass form are as indicated below. Here, the domain parameters of the Montgomery curve

Curve25519 and of the twisted Edwards curve Edwards25519 are as specified in [RFC7748]; the domain parameters of the elliptic curve Wei25519 in short-Weierstrass form comply with Section 6.1.1 of [FIPS186-4]. For further details on these curves and on the coordinate transformations referenced above, see [CURVE-REPR].

General parameters (for all curve models):

p 2^{255}-19
(=0x7fffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff fffffffed)
h 8
n
72370055773322622139731865630429942408571163593799076060019509382854454250989
(=2^{252} + 0x14def9de a2f79cd6 5812631a 5cf5d3ed)

Montgomery curve-specific parameters (for Curve25519):

A 486662
B 1
Gu 9 (=0x9)
Gv
14781619447589544791020593568409986887264606134616475288964881837755586237401
(=0x20ae19a1 b8a086b4 e01edd2c 7748d14c 923d4d7e 6d7c61b2 29e9c5a2 7eced3d9)

Twisted Edwards curve-specific parameters (for Edwards25519):

a -1 (-0x01)
d -121665/121666
(=37095705934669439343138083508754565189542113879843219016388785533085940283555)
(=0x52036cee 2b6ffe73 8cc74079 7779e898 00700a4d 4141d8ab 75eb4dca 135978a3)
Gx
15112221349535400772501151409588531511454012693041857206046113283949847762202
(=0x216936d3 cd6e53fe c0a4e231 fdd6dc5c 692cc760 9525a7b2 c9562d60 8f25d51a)
Gy 4/5
(=46316835694926478169428394003475163141307993866256225615783033603165251855960)
(=0x66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666658)

Weierstrass curve-specific parameters (for Wei25519):

a
19298681539552699237261830834781317975544997444273427339909597334573241639236
(=0x2aaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaa98 4914a144)
b
55751746669818908907645289078257140818241103727901012315294400837 9

56729358436
(=0x7b425ed0 97b425ed 097b425e d097b425 ed097b42 5ed097b4 260b5e9c 7710c864)

GX
192986815395526992372618308347813179755449974442734273399095973346 52188435546
(=0x2aaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaad245a)

GY
147816194475895447910205935684099868872646061346164752889648818377 55586237401
(=0x20ae19a1 b8a086b4 e01edd2c 7748d14c 923d4d7e 6d7c61b2 29e9c5a2 7eced3d9)

Acknowledgments

Many thanks to Charlie Perkins for his in-depth review and constructive suggestions. The authors are also especially grateful to Robert Moskowitz and Benjamin Kaduk for their comments and discussions that led to many improvements. The authors wish to also thank Shwetha Bhandari for actively shepherding this document and Roman Danyliw, Alissa Cooper, Mirja Kühlewind, Éric Vyncke, Vijay Gurbani, Al Morton, and Adam Montville for their constructive reviews during the IESG process. Finally, many thanks to our INT area ADs, Suresh Krishnan and Erik Kline, who supported us along the whole process.

Authors' Addresses

Pascal Thubert (editor)
Cisco Systems, Inc.
Building D
45 Allee des Ormes - BP1200
06254 MOUGINS - Sophia Antipolis
France

Phone: +33 497 23 26 34
Email: pthubert@cisco.com

Behcet Sarikaya

Email: sarikaya@ieee.org

Mohit Sethi
Ericsson
FI-02420 Jorvas
Finland

Email: mohit@piuha.net

Rene Struik
Struik Security Consultancy

Email: rstruik.ext@gmail.com