Network Working Group
Internet-Draft
Intended status: Informational
Expires: 5 July 2026

C. de Kater
Independent
N. Rustignoli
SCION Association
J. C. Hugly
Independent
S. Hitz
Anapaya Systems
1 January 2026

SCION Data Plane
draft-dekater-scion-dataplane-10

Abstract

This document describes the data plane of the path-aware, inter-domain network architecture SCION (Scalability, Control, and Isolation On Next-generation networks). A fundamental characteristic of SCION is that it gives path control to SCION-capable endpoints.

The SCION Control Plane is responsible for discovering these paths and making them available as path segments to the endpoints. The role of the SCION Data Plane is to combine these path segments into end-to-end paths, and forward data between endpoints according to the specified path. It fundamentally differs from an IP-based data plane in that it is _path-aware_ and interdomain forwarding directives are embedded in the packet header.

This document provides a detailed specification of the SCION data packet format as well as the structure of the SCION header, including extension headers. It also describes the life of a SCION packet while traversing a SCION network, followed by a specification of the SCION path authorization mechanisms and the packet processing at routers.

This document contains new approaches to secure path aware networking. It is not an Internet Standard, has not received any formal review of the IETF, nor was the work developed through the rough consensus process. The approaches offered in this work are offered to the community for its consideration in the further evolution of the Internet.

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://scionassociation.github.io/scion-dp_I-D/draft-dekater-scion-dataplane.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-dekater-scion-dataplane/.

Source for this draft and an issue tracker can be found at https://github.com/scionassociation/scion-dp_I-D.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

SCION is a path-aware internetworking routing architecture as described in [RFC9217]. It allows endpoints and applications to select paths across the network to use for traffic, based on trusted path properties. SCION is an inter-domain network architecture and is therefore not concerned with intra-domain forwarding.

SCION relies on three main components:

PKI - providing cryptographic material within an unique trust model. It is described in [I-D.dekater-scion-pki].

Control Plane - performing inter-domain routing by discovering and securely disseminating path information. It is described in [I-D.dekater-scion-controlplane].

Data Plane - carrying out secure packet forwarding between SCION-enabled ASes over paths selected by endpoints. It is described in this document.

A more detailed introduction, motivation, and problem statement are provided in [I-D.dekater-scion-controlplane]. Readers are encouraged to read the introduction in that document first.

The SCION architecture was initially developed outside of the IETF by ETH Zurich with significant contributions from Anapaya Systems. It is deployed in the Swiss finance sector to provide resilient connectivity between financial institutions. The aim of this document is to document the existing protocol specification as deployed, to encourage interoperability among implementations, and to introduce new concepts that can potentially be further improved to address particular problems with the current Internet architecture.

1.1. Terminology

SCION Autonomous System (AS): A SCION Autonomous System is a network under a common administrative control. For example, the network of a SCION service provider, company, or university can constitute an AS. While functionally similar to a BGP AS, a SCION AS operates within an Isolation Domain (ISD), utilizes a different address scheme, and serves as a locator in the addressing of end hosts. References to ASes throughout this document refer to SCION ASes.

Core AS: Each Isolation Domain (ISD) is administered by a set of distinguished SCION autonomous systems (ASes) called core ASes, which are responsible for initiating the path discovery and path construction process (called "beaconing" in SCION).

Data Plane: The data plane (sometimes also referred to as the forwarding plane) is responsible for forwarding data packets that endpoints have injected into the network. After routing information has been disseminated by the control plane, packets are forwarded by the data plane in accordance with such information.

Egress/Ingress: Refers to the direction of travel. In SCION, path construction with beaconing happens in one direction, while actual traffic might follow the opposite direction. This document clarifies on a case-by-case basis whether ’egress’ or ’ingress’ refers to the direction of travel of the SCION packet or to the direction of beaconing.

Endpoint: An endpoint is the start or the end of a SCION path, as defined in [RFC9473].

Forwarding Key: A symmetric key that is shared between the control service (control plane) and the routers (data plane) of an AS. It is used to authenticate Hop Fields in the end-to-end SCION path. The forwarding key is an AS-local secret and is not shared with other ASes.

Forwarding Path: A complete end-to-end path between two SCION endpoints which is used to transmit packets in the data plane. It can be created with a combination of up to three path segments (an up segment, a core segment, and a down segment).

Hop Field (HF): As they traverse the network, Path-Segment Construction Beacons (PCBs) accumulate cryptographically protected AS-level path information in the form of Hop Fields. In the data plane, Hop Fields are used for packet forwarding: they contain the incoming and outgoing Interface IDs of the ASes on the forwarding path.

Info Field (INF): Each Path-Segment Construction Beacon (PCB) contains a single Info field, which provides basic information about the PCB. Together with Hop Fields (HFs), these are used to create forwarding paths.

Interface Identifier (Interface ID): A 16-bit identifier that designates a SCION interface at the end of a link connecting two SCION ASes, with each interface belonging to one border router. Hop fields describe the traversal of an AS by a pair of Interface IDs called ConsIngress and ConsEgress as they refer to the ingress and egress interfaces in the direction of path construction (beaconing). Each Interface ID MUST be unique within each AS. 0 is a reserved value that indicates the lack of an Interface ID and is used as the unspecified Interface ID (see Section 2.3.3).

Isolation Domain (ISD): SCION ASes are organized into logical groups called Isolation Domains or ISDs. Each ISD consists of ASes that span an area with a uniform trust environment (e.g. a common jurisdiction).

Message Authentication Code (MAC): In the rest of this document, "MAC" always refers to "Message Authentication Code" and never to "Medium Access Control". When "Medium Access Control address" is implied, the phrase "Link Layer Address" is used.

Path Authorization: A requirement for the data plane is that endpoints can only use paths that were constructed and authorized by ASes in the control plane. The goal of path authorization is to prevent endpoints from crafting Hop Fields (HFs) themselves, modifying HFs in authorized path segments, or combining HFs of different path segments.

Path Control: The property of a network architecture that gives endpoints the ability to select how their packets travel through the network. Path control is stronger than path transparency.

Path Segment: These are derived from Path-Segment Construction Beacons (PCBs). A path segment can be (1) an up segment (i.e. a path between a non-core AS and a core AS in the same ISD), (2) a down segment (i.e. the same as an up segment, but in the opposite direction), or (3) a core segment (i.e. a path between core ASes). Up to three path segments can be used to create a forwarding path.

Path-Segment Construction Beacon (PCB): Core AS control planes generate PCBs to explore paths within their isolation domain (ISD) and between different ISDs. ASes further propagate selected PCBs to their neighboring ASes. These PCBs traverse each AS accumulating information, including Hop Fields (HFs) which can subsequently be used for traffic forwarding.

Path Transparency: This is a property of a network architecture that gives endpoints full visibility over the network paths their packets are taking. Path transparency is weaker than path control.

Peering Link: A link between two SCION border routers of different ASes that can be used as a shortcut. Peering link information is added to segment information during the beaconing process and used to shorten paths while assembling them from segments. It is possible to construct a path out of only two partial segments whose top-most hops are joined by a peering link. Two peering ASes may be in different ISDs and may exist between any ASes, including core ASes.

SCION Control Message Protocol (SCMP): A signaling protocol analogous to the Internet Control Message Protocol (ICMP), as described in [I-D.dekater-scion-controlplane].

1.2. 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.

1.3. Overview

The SCION Data Plane forwards inter-domain packets between SCION ASes. SCION routers are normally deployed at the edge of an AS and peer with neighboring SCION routers. Inter-domain forwarding is based on end-to-end path information contained in the packet header, which consists of a sequence of Hop Fields (HFs). Each Hop Field corresponds to an AS on the path, and it includes an ingress Interface ID as well as an egress Interface ID which unequivocally identifies the ingress and egress interfaces within the AS. The

information is authenticated with a Message Authentication Code (MAC) to prevent forgery.

This concept allows SCION routers to forward packets to a neighbor AS without inspecting the destination address and also without consulting an inter-domain forwarding table. A SCION border router reuses existing intra-domain infrastructure (e.g. IP) to communicate to other SCION routers or SCION endpoints within its AS. The last SCION router at the destination AS uses the destination address to forward the packet to the appropriate local endpoint.

This SCION design choice has the following advantages:

* It provides control and transparency over forwarding paths to endpoints.

* It simplifies the packet processing at routers. Instead of having to perform longest prefix matching on IP addresses which requires dedicated hardware and substantial amounts of energy, a router can simply access the next hop from the packet header after having verified the authenticity of the Hop Field’s MAC.

1.3.1. Inter- and Intra-Domain Forwarding

As SCION is an inter-domain network architecture, it is not concerned with intra-domain forwarding. This corresponds to the general practice today where BGP and IP are used for inter-domain routing and forwarding respectively, but ASes use an intra-domain protocol of their choice - e.g. OSPF or IS-IS for routing, and IP, MPLS, and various Layer 2 protocols for forwarding. In fact, even if ASes use IP forwarding internally, they typically encapsulate the original IP packet they receive at the edge of their network into another IP packet with the destination address set to the egress border router, in order to avoid full inter-domain forwarding tables on internal routers.

SCION emphasizes this separation as it is used exclusively for inter-domain forwarding; re-using the intra-domain network fabric to provide connectivity amongst all SCION infrastructure services, border routers, and endpoints. As a consequence, minimal change to the infrastructure is required for ISPs when deploying SCION.

In practice, in most existing SCION deployments the SCION routers communicate amongst themselves and with endpoints by enclosing the SCION header inside an UDP/IPv6 or UDP/IPv4 packet. The choice of using an UDP/IP as an intra-domain protocol between routers was driven by the need to maximize compatibility with existing networks. Id does not exclude that a SCION packet may be enclosed directly on top of a Layer 2 protocol, since the choice of intra-domain protocol is AS specific.

Figure 1 shows the SCION header within the protocol stack, in an AS where the SCION deployment uses UDP/IP as an intra-domain protocol. A similar model may be used for inter-domain links, depending on the individual choice of the two interconnected SCION router operators. A full example of the life of a SCION packet is later presented in Section 3. A list of currently used upper layer protocols on top of SCION is presented in Appendix "Assigned SCION Protocol Numbers".

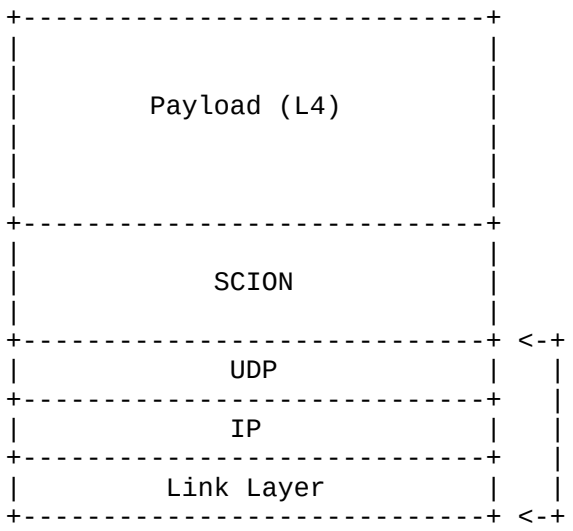

Figure 1: The SCION header within the protocol stack in a typical deployment

A complete SCION address is composed of the <ISD, AS, endpoint address> 3-tuple. The ISD-AS part is used for inter-domain routing, whilst the endpoint address part is only used for intra-domain forwarding at the source and destination ASes. This implies that endpoint addresses are only required to be globally unique within each SCION AS. An endpoint running a SCION stack using a [RFC1918] could therefore directly communicate with another SCION endpoint using a [RFC1918] endpoint address in a different SCION AS.

The data transmission order for SCION is the same as for IPv6 as defined in Introduction of [RFC8200].

1.3.2. Intra-Domain Forwarding Process

When transiting an intermediate SCION AS, a packet gets forwarded by at most two SCION routers. The forwarding process consists of the following steps.

1. The AS's SCION ingress router receives a SCION packet from the neighboring AS.

2. The SCION router parses, validates, and authenticates the SCION header.

3. The SCION router maps the egress Interface ID in the current Hop Field of the SCION header to the destination address of the intra-domain protocol (e.g. MPLS or IP) of the egress border router.

4. The packet is forwarded within the AS by SCION-unaware routers and switches based on the header of the intra-domain protocol.

5. Upon receiving the packet, the SCION egress router strips off the header of the intra-domain protocol, again validates and updates the SCION header, and forwards the packet to the neighboring SCION router.

6. The last SCION router on the path forwards the packet to the packet's destination endpoint indicated by the field DstHostAddr of the Address Header (Section 2.2).

1.3.3. Configuration

Border routers require mappings from SCION Interface IDs to underlay addresses and such information MUST be supplied to each router in an out of band fashion (e.g in a configuration file). For each link to a neighbor, these values MUST be configured. A typical implementation will require:

* Interface ID.

* Link type (core, parent, child, peer). Link type depends on mutual agreements between the organizations operating the ASes at each end of each link.

* Neighbor ISD-AS number.

* Neighbor interface’s underlay address.

* For intra-domain forwarding: mapping of the AS interface IDs to intra-domain protocol address of the corresponding routers.

* The algorithm used to compute the Hop Field MAC (Section 4.1.1) which must be the same as that used by the Control Services within the AS.

In order to forward traffic to a service endpoint address (DT/DS as per Table 3), a border router translates the service number into a specific destination address. The method used to accomplish the translation is not defined by this document and is only dependent on the implementation and the choices of each AS’s administrator. In current practice this is accomplished by way of a configuration file.

In addition, routers require coarse time synchronization with control plane instances (see Section 4.2.2.3).

Note: The current SCION implementation runs over the UDP/IP protocol. However, the use of other lower layers protocols is possible.

1.4. Path Construction (Segment Combinations)

Paths are discovered by the Control Plane which makes them available to SCION endpoints in the form of path segments. As described in [I-D.dekater-scion-controlplane], there are three kinds of path segments: up, down, and core.

In the data plane, a SCION endpoint creates end-to-end paths from the path segments by combining multiple path segments. Depending on the network topology, a SCION forwarding path consists of at least one and up to three segments. Each path segment contains several Hop Fields representing the ASes on the segment as well as one Info Field with basic information about the segment - e.g. a timestamp.

Segments cannot be combined arbitrarily. To construct a valid forwarding path, the source endpoint MUST obey the following rules:

* There MUST be at most one of each type of segment (up, core, and down). Allowing multiple up or down segments would decrease efficiency and the ability of ASes to enforce path policies.

* If an up segment is present, it MUST be the first segment in the path.

* If a down segment is present, it MUST be the last segment in the path.

* If there are two path segments (one up and one down segment) that both announce the same peering link, then a shortcut via this peering link is possible.

* If there are two path segments (one up and one down segment) that share a common ancestor AS (in the direction of beaconing), then a shortcut via this common ancestor AS is possible. The up-then-down constraint still applies.

* Additionally, all segments without any peering possibility MUST consist of at least two Hop Fields.

Note that the type of segment is known to the endpoint but it is not explicitly visible in the path header of data packets. Therefore, a SCION router needs to explicitly verify that these rules were followed correctly by performing checks described in Section 4.2.2.1.

Besides enabling the enforcement of path policies, the above rules also protect the economic interest of ASes as they prevent building "valley paths". A valley path contains ASes that do not profit economically from traffic on this route, with the name coming from the fact that such paths go "down" (following parent-child links) before going "up" (following child-parent links).

Figure 2 and Figure 3 show valid segment combinations with each node representing a SCION AS. It is assumed that the source and destination endpoints are in different ASes (as endpoints from the same AS use an empty forwarding path to communicate with each other).

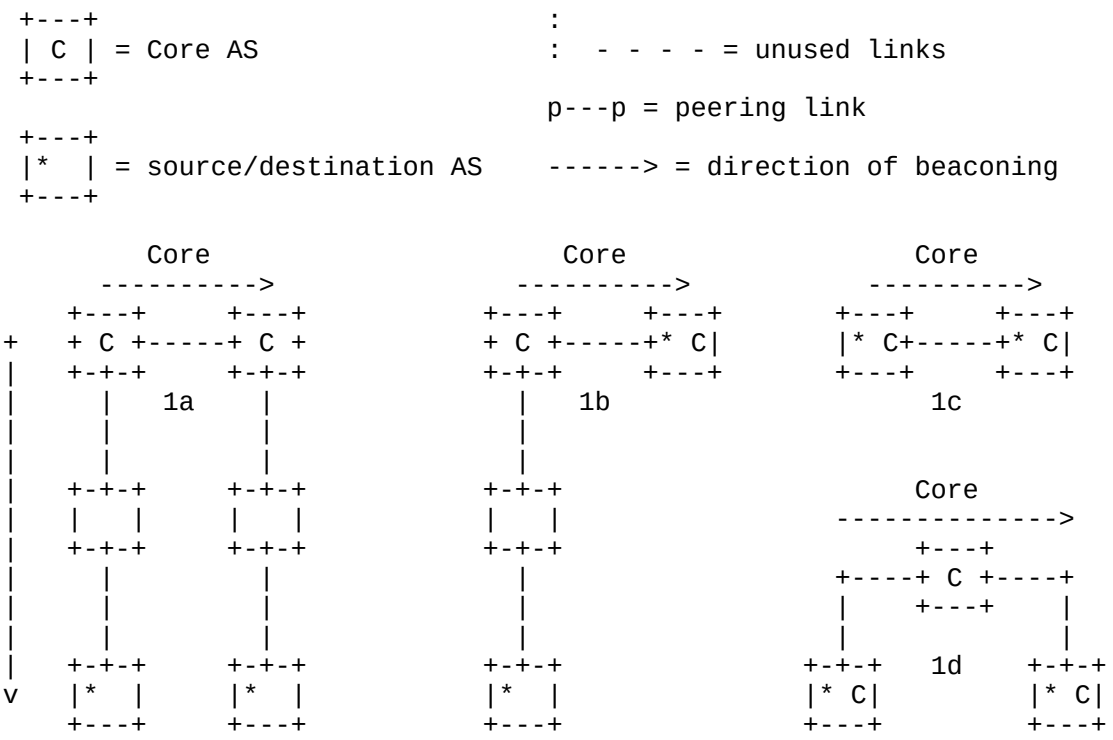

Figure 2: Illustration of valid path segment combinations through multiple core ASes.

Valid path segment combinations:

* *Communication through core ASes*:

  - *Core segment combination* (Cases 1a, 1b, 1c, 1d in Figure 2): The up and down segments of source and destination do not have an AS in common. In this case, a core segment is REQUIRED to connect the source's up segment and the destination's down segment (Case 1a). If either the source or the destination AS is a core AS (Case 1b), or both are core ASes (Cases 1c and 1d), then no up or down segments are REQUIRED to connect the respective ASes to the core segment.

  - *Immediate combination* (Cases 2a, 2b in Figure 3): The last AS on the up segment (which is necessarily a core AS) is the same as the first AS on the down segment. In this case, a simple combination of up and down segments creates a valid forwarding path. In Case 2b, only one segment is required.

* *Peering shortcut* (Cases 3a and 3b): A peering link exists between the up and down segment, and extraneous path segments to the core are cut off. Note that the up and down segments do not need to originate from the same core AS and the peering link could also be traversing to a different ISD.

* *AS shortcut* (Cases 4a and 4b): The up and down segments intersect at a non-core AS below the ISD core, thus creating a shortcut. In this case, a shorter path is made possible by removing the extraneous part of the path to the core. Note that the up and down segments do not need to originate from the same core AS.

* *On-path* (Case 5): In the case where the source’s up segment contains the destination AS or the destination’s down segment contains the source AS, a single segment is sufficient to construct a forwarding path. Again, no core AS is on the final path.

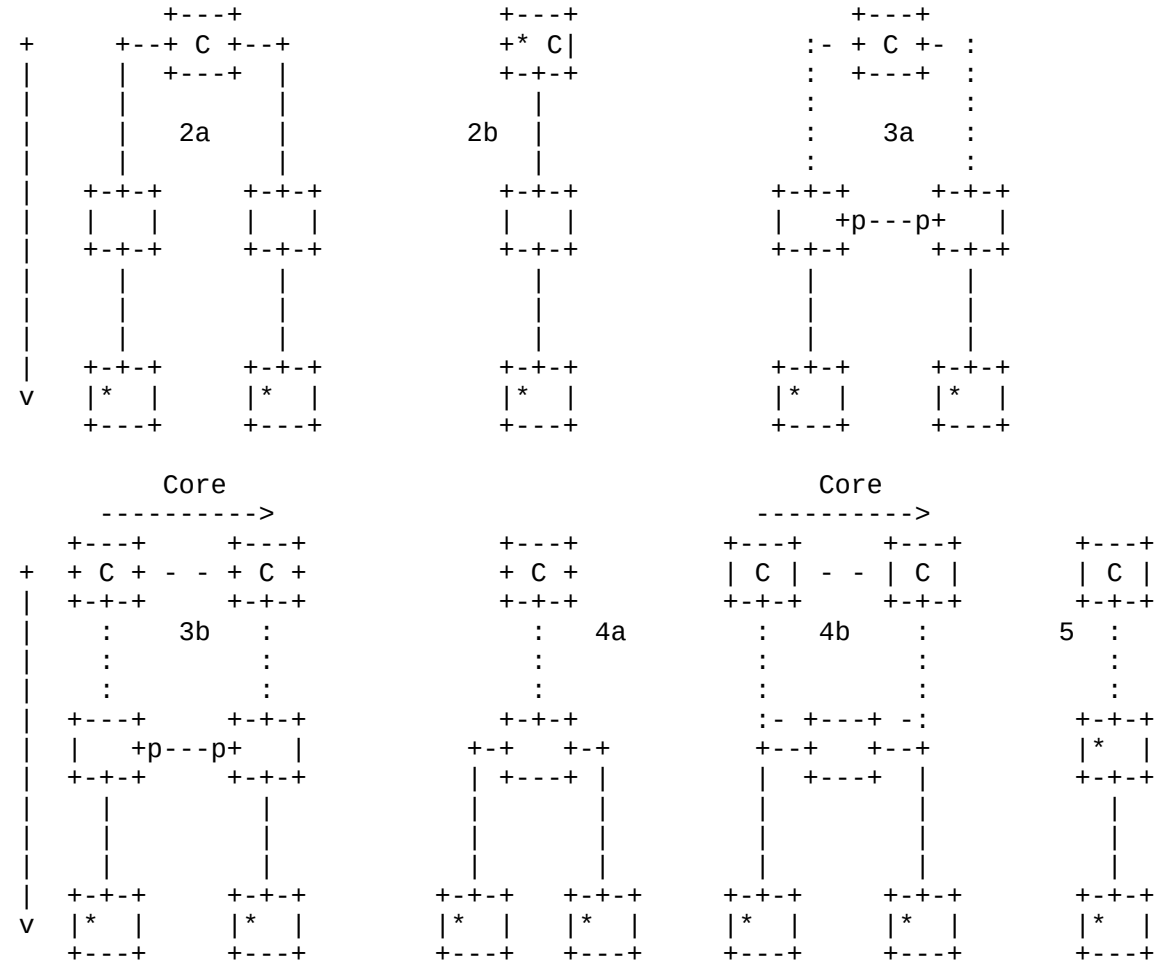

Figure 3: Illustration of valid path segment combinations through one or no core ASes.

1.5. Path Authorization

The SCION Data Plane provides _path authorization_. This ensures that data packets always traverse the network using path segments that were explicitly authorized by the respective ASes and prevents endpoints from constructing unauthorized paths or paths containing loops. SCION uses symmetric cryptography in the form of Message Authentication Codes (MACs) to authenticate the information encoded in Hop Fields and such MACs are verified by routers at forwarding. For a detailed specification, see Section 4.

2. SCION Header Specification

The SCION packet header is aligned to 4 bytes and is composed of a common header, an address header, a path header, and an OPTIONAL extension header, see Figure 4 below. The 4 byte alignment is to allow header length to be computed based on the HdrLen field (see Section 2.1).

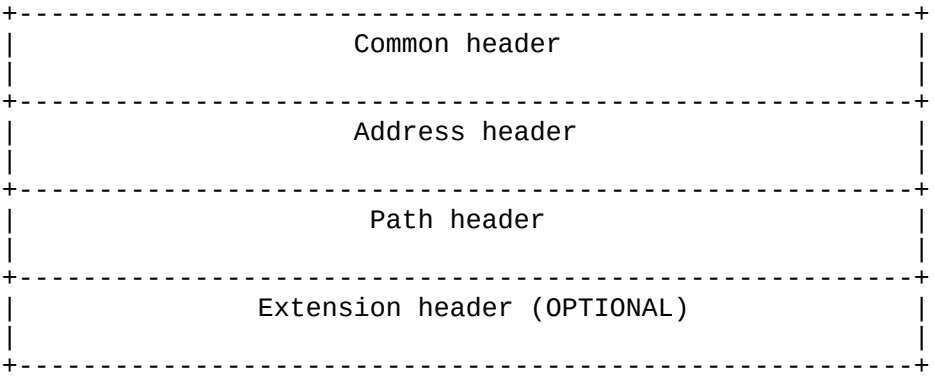

Figure 4: High-level SCION header structure, non-byte aligned

The _common header_ contains important meta information including version number and the lengths of the header and payload. In particular, it contains flags that control the format of subsequent headers such as the address and path headers. For more details, see Section 2.1.

The _address header_ contains the ISD, SCION AS and endpoint addresses of source and destination. The type and length of endpoint addresses are variable and can be set independently using flags in the common header. For more details, see Section 2.2.

The _path header_ contains the full AS-level forwarding path of the packet. A path type field in the common header specifies the path format used in the path header. For more details, see Section 2.3.

The OPTIONAL _extension_ header contains a variable number of hop-by-hop and end-to-end options, similar to extensions in the IPv6 header [RFC8200]. For more details, see Section 2.4.

2.1. Common Header

The SCION common header has the following packet format:

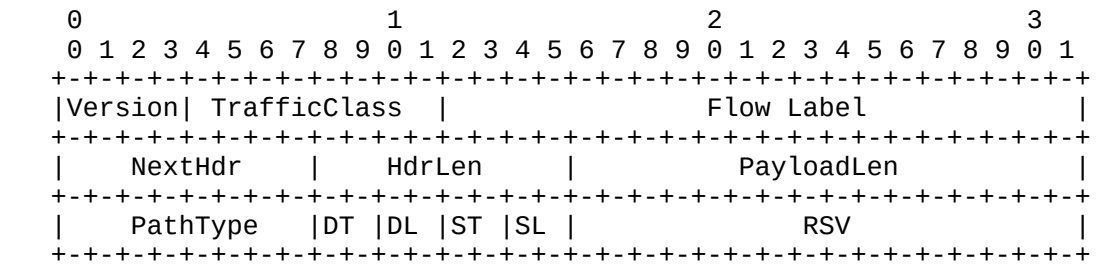

Figure 5: The SCION common header packet format

* Version: The version of the SCION common header. Currently, only version "0" is supported.

* TrafficClass: The 8-bit long identifier of the packet’s class or priority. The value of the traffic class bits in a received packet might differ from the value sent by the packet’s source. The current use of the TrafficClass field for Differentiated Services and Explicit Congestion Notification is specified in [RFC2474] and [RFC3168].

* Flow Label: This 20-bit field labels sequences of packets to be treated in the network as a single flow. Sources MUST set this field which serves the same purpose as what [RFC6437] describes for IPv6 and is used in the same manner. Note that a Flow Label of zero does not imply that packet reordering is acceptable.

* NextHdr: Encodes the type of the first header after the SCION header, which can be either a SCION extension or a Layer 4 protocol such as TCP or UDP. Values of this field respect the Assigned SCION Protocol Numbers (see Appendix "Assigned SCION Protocol Numbers").

* HdrLen: Specifies the entire length of the SCION header in bytes - i.e. the sum of the lengths of the common header, the address header, and the path header. The SCION header is aligned to a multiple of 4 bytes. The SCION header length is computed as HdrLen * 4 bytes. The 8 bits of the HdrLen field limit the SCION header to a maximum of 255 * 4 = 1020 bytes.

* PayloadLen: Specifies the length of the payload in bytes. The payload includes (SCION) extension headers and the L4 payload. This field is 16 bits long, supporting a maximum payload size of 65’535 bytes.

* PathType: Specifies the type of the SCION path and is 8 bits long. The format of one path type is independent of all other path types. The currently defined SCION path types are Empty (0), SCION (1), OneHopPath (2), EPIC (3) and COLIBRI (4). This document only specifies the Empty, SCION and OneHopPath path types and the other path types are currently experimental. For more details, see Section 2.3.

Value	Path Type
0	Empty path (EmptyPath)
1	SCION (SCION)
2	One-hop path (OneHopPath)
3	EPIC path (experimental)
4	COLIBRI path (experimental)

Table 1: SCION path types

* DT/DL/ST/SL: These fields define the endpoint address type and endpoint address length for the source and destination endpoint. DT and DL stand for Destination Type and Destination Length, whereas ST and SL stand for Source Type and Source Length. The possible endpoint address length values are 4 bytes, 8 bytes, 12 bytes, and 16 bytes. If an address has a length different from the supported values, the next larger size SHALL be used and the address can be padded with zeros. Table 2 below lists the currently used values for address length. The "type" identifier is only defined in combination with a specific address length - e.g. address type "0" is defined as IPv4 in combination with address length 4, but is defined as IPv6 in combination with address length 16. Per address length, several sub-types are possible and Table 3 shows the currently assigned combinations of lengths and types.

DL/SL Value	Address Length
0	4 bytes
1	8 bytes
2	12 bytes
3	16 bytes

Table 2: Address length values

Type (DT/ST)	Length (DL/SL)	Conventional Use
0	0	IPv4
0	3	IPv6
1	0	Service
other	other	Unassigned

Table 3: Allocations of length and type combinations

A service address designates a set of endpoint addresses rather than a single one. A packet addressed to a service is redirected to any one endpoint address that is known to be part of the set. Table 4 lists the known services.

* RSV: These bits are currently reserved for future use.

2.2. Address Header

The SCION address header has the following format:

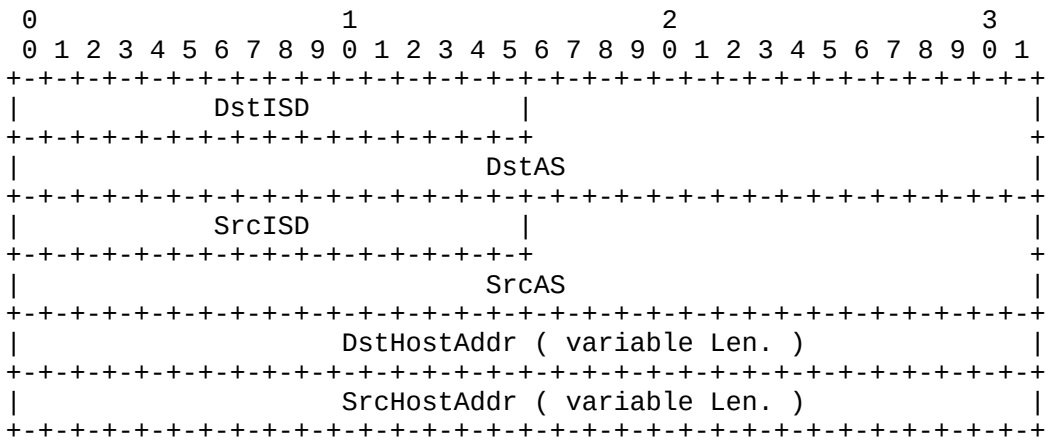

Figure 6: The SCION address header packet format

* DstISD, SrcISD: The 16-bit ISD identifier of the destination/ source.

* DstAS, SrcAS: The 48-bit SCION AS identifier of the destination/ source.

* DstHostAddr, SrcHostAddr: Specifies the variable length endpoint address of the destination/source. The accepted type and length are defined in the DT/DL/ST/SL fields of the common header.

If a service address is implied by the DT/DL or ST/SL field of the common header according to Table 3, the corresponding address field has the following format:

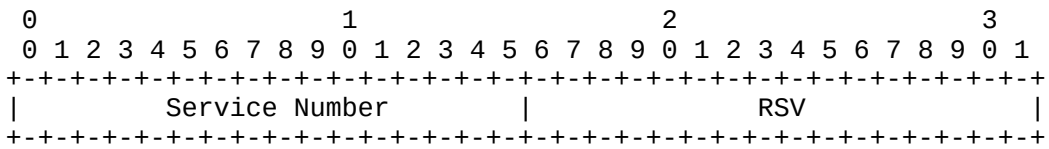

Figure 7: Service address format

* RSV: reserved for future use

The currently known service numbers are:

Service Number (hex)	Short Name	Description
0001	DS	Discovery Service
0002	CS	Control Service
FFFF	None	Reserved invalid value

Table 4: Known Service Numbers

Note: For more information on addressing, see the ([I-D.dekater-scion-controlplane]).

2.3. Path Header

The path header of a SCION packet differs for each SCION path type. The path type is set in the PathType field of the SCION common header.

SCION supports three path types:

2.3.1. Empty Path Type

The Empty path type (PathType=0) is used to send traffic within an AS. It has no additional fields - i.e. it consumes 0 bytes on the wire.

One use case of the Empty path type lies in the context of link-failure detection (Section 6.1).

2.3.2. SCION Path Type

The SCION path type (PathType=1) is the standard path type. A SCION path has the following layout:

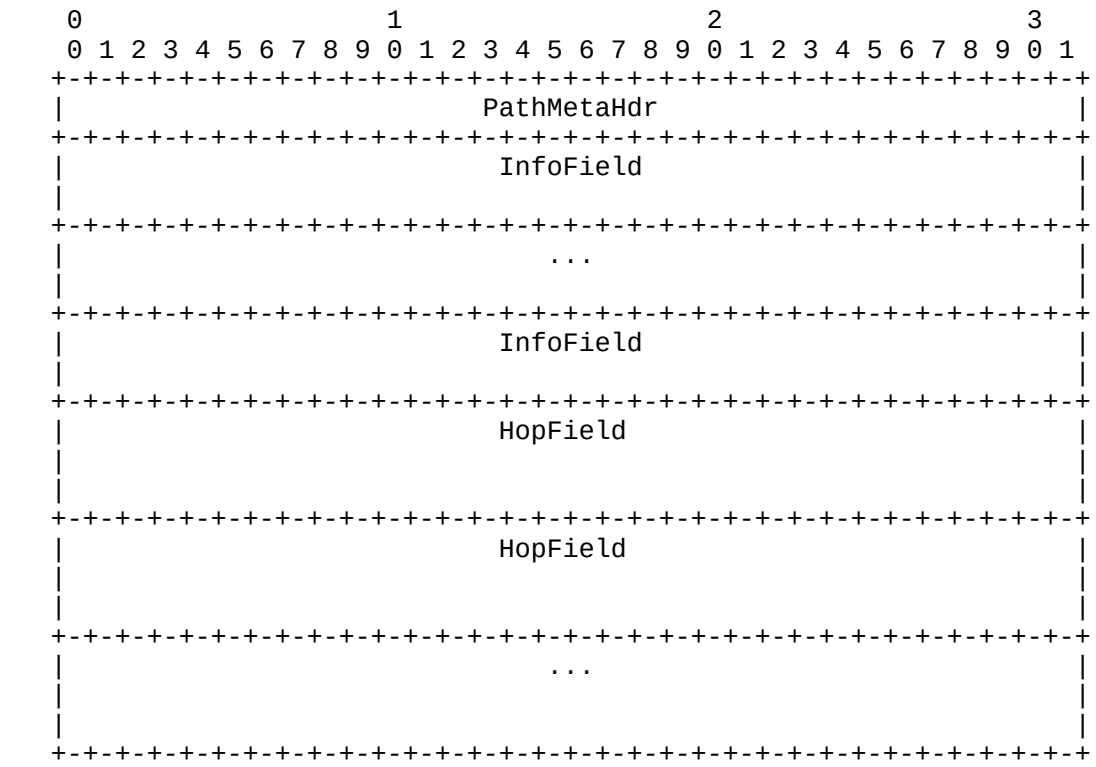

Figure 8: Layout of a standard SCION path

It consists of a path meta header, up to 3 Info Fields and up to 64 Hop Fields.

* PathMetaHdr indicates the currently valid Info Field and Hop Field while the packet is traversing the network along the path, as well as the number of Hop Fields per segment.

* InfoField equals the number of path segments that the path contains - there is one Info Field per path segment. Each Info Field contains basic information about the corresponding segment, such as a timestamp indicating the creation time. There are also two flags: one specifies whether the segment is to be traversed in construction direction, the other whether the first or last Hop Field in the segment represents a peering Hop Field.

* HopField represents a hop through an AS on the path, with the ingress and egress interface identifiers for this AS. This information is authenticated with a Message Authentication Code (MAC) to prevent forgery.

The SCION header is created by extracting the required Info Fields and Hop Fields from the corresponding path segments, and this process is illustrated in Figure 9 below. Note that ASes at the intersection of multiple segments are represented by two Hop Fields. Be aware that these Hop Fields are not equal!

In the Hop Field that represents the last Hop in the first segment (seen in the direction of travel), only the ingress interface will be specified. However, in the hop Field that represents the first hop in the second segment (also in the direction of travel), only the egress interface will be defined. Thus, the two Hop Fields for this one AS build a full hop through the AS, specifying both the ingress and egress interface. As such, they bring the two adjacent segments together.

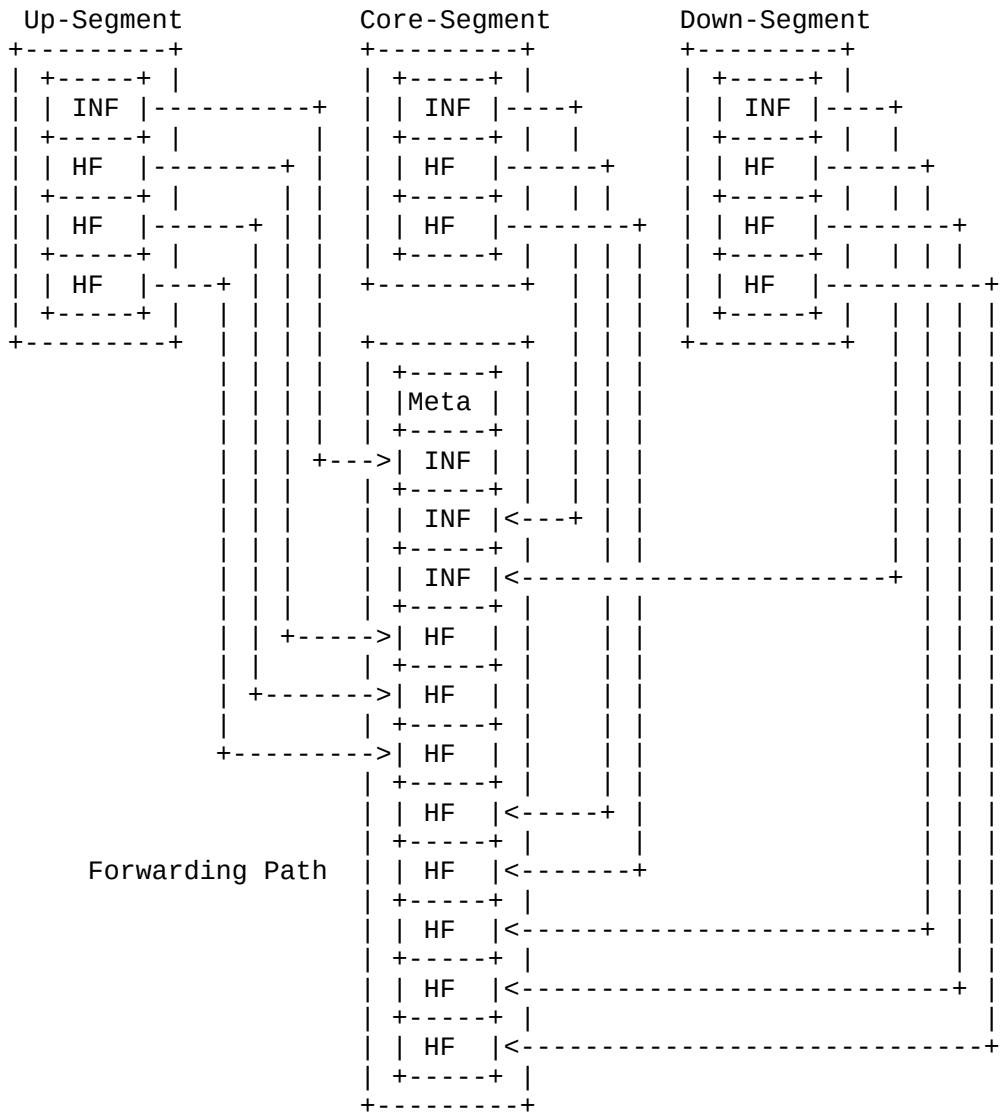

Figure 9: Path construction example

2.3.2.1. Path Meta Header Field

The 4-byte Path Meta Header field (PathMetaHdr) defines meta information about the SCION path that is contained in the path header. It has the following format:

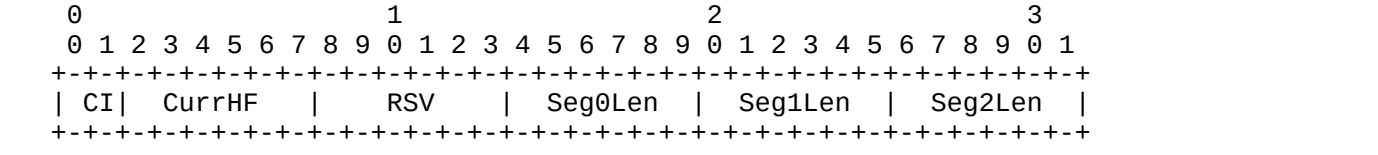

Figure 10: SCION path type - Format of the Path Meta Header field

* CurrINF (shown as CI above): Specifies a 2-bits index (0-based) pointing to the current Info Field for the packet on its way through the network. For details, see Section 2.3.2.2 below.

* CurrHF: Specifies a 6-bits index (0-based) pointing to the current Hop Field for the packet on its way through the network. For details, see Section 2.3.2.2 below. Note that the CurrHF index MUST point to a Hop Field that is part of the current path segment, as indicated by the CurrINF index.

Both indices are used by SCION routers when forwarding data traffic through the network. The SCION routers also increment the indexes if required. For more details, see Section 4.2.2.

* Seg{0,1,2}Len: The number of Hop Fields in a given segment. Seg{i}Len > 0 implies that segment _i_ contains at least one Hop Field, which means that Info Field _i_ exists. (If Seg{i}Len = 0 then segment _i_ is empty, meaning that this path does not include segment _i_, and therefore there is no Info Field _i_.) The following rules apply:

  - The total number of Hop Fields in an end-to-end path MUST be equal to the sum of all Seg{0,1,2}Len contained in this end-to-end path.

  - It is an error to have Seg{X}Len > 0 AND Seg{Y}Len == 0, where 2 >= _X_ > _Y_ >= 0. That is, if path segment Y is empty, the following path segment X MUST also be empty.

* RSV: Unused and reserved for future use.

2.3.2.2. Path Offset Calculations

Path offset calculations enable SCION border routers to locate the currently active Info Field and Hop Field during packet processing.

The following rules apply when calculating the path offsets:

```
if Seg2Len > 0: NumINF = 3
else if Seg1Len > 0: NumINF = 2
else if Seg0Len > 0: NumINF = 1
else: invalid
```

The offsets of the current Info Field and current Hop Field (relative to the end of the address header) are now calculated as:

```
B = byte
InfoFieldOffset = 4B + 8B * CurrINF
HopFieldOffset = 4B + 8B.NumINF + 12B * CurrHF
```

To check that the current Hop Field is in the segment of the current Info Field, the CurrHF needs to be compared to the SegLen fields of the current and preceding Info Fields.

2.3.2.3. Info Field

The 8-byte Info Field (InfoField) has the following format:

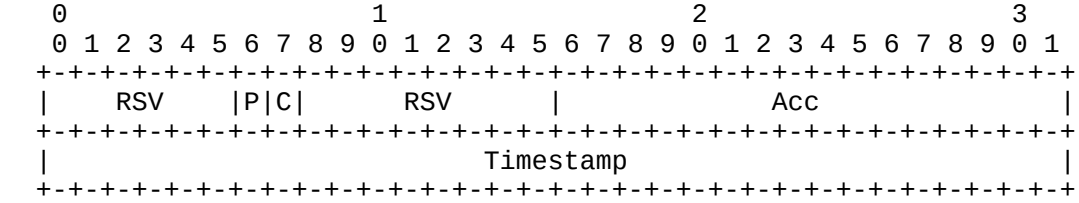

Figure 11: SCION path type - Format of the Info Field

* RSV: Unused and reserved for future use.

* P: Peering flag. If the flag has value "1", the segment represented by this Info Field contains a peering Hop Field, which requires special processing in the data plane. For more details, see Section 4.1.2 and Section 4.2.

* C: Construction direction flag. If the flag has value "1", the Hop Fields in the segment represented by this Info Field are arranged in the direction they have been constructed during beaconing.

* Acc: Accumulator. This updatable field/counter is REQUIRED for calculating the MAC in the data plane. For more details, see Section 4.1.

* Timestamp: Timestamp created by the initiator of the corresponding beacon. The timestamp is defined as the seconds since Epoch according to [POSIX.1-2024] Section 4.19, encoded as a 32-bit unsigned integer. This timestamp enables the validation of a Hop Field in the segment represented by this Info Field, by verifying the expiration time and MAC set in the Hop Field - the expiration time of a Hop Field is calculated relative to the timestamp. An Info field with a timestamp in the future is invalid, and for the purpose of validation, a timestamp is considered "future" if it is later than the locally available current time plus 337.5 seconds (i.e. the minimum time to live of a hop). This timestamp wraps around every 2^32 seconds (roughly 136 years) with the next wraparound occurring in year 2106. Care should be taken by implementations while computing validity during a wraparound.

2.3.2.4. Hop Field

The 12-byte Hop Field (HopField) has the following format:

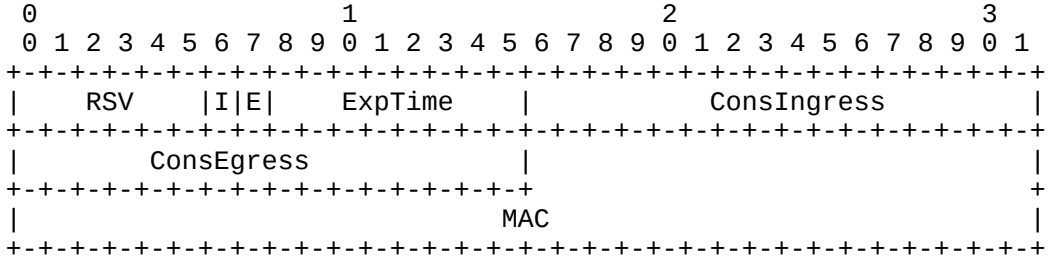

Figure 12: SCION path type - Format of the Hop Field

* RSV: Unused and reserved for future use.

* I: The Ingress Router Alert flag. If this has value "1" and the packet is received on the interface with ID corresponding to the value of ConsIngress, the router SHOULD process the L4 payload in the packet.

* E: The Egress Router Alert flag. If this has value "1" and the packet is received on the interface with ID corresponding to the value of ConsEgress, the router SHOULD process the L4 payload in the packet.

* ExpTime: Expiration time of a Hop Field. This field is 1-byte long, and the expiration time specified in this field is relative and expressed in units of 256th of a day. An absolute expiration time in seconds is computed in combination with the Timestamp field (from the corresponding Info Field), as follows:

- Timestamp + (1 + ExpTime) * (86400/256)

* ConsIngress, ConsEgress: The 16-bits ingress/egress Interface IDs in construction direction, that is, the direction of beaconing.

* MAC: The 6-byte Message Authentication Code to authenticate the Hop Field. For details on how this MAC is calculated, see Section 4.1.1.

The Ingress Router (respectively Egress Router) is the router owning the Ingress interface (respectively Egress interface) when the packet is traveling in the _construction direction_ of the path segment (i.e. the direction of beaconing). When the packet is traveling in the opposite direction, the meanings are reversed.

Router alert flags work similarly to [RFC2711] and allow a sender to address a specific router on the path without knowing its address. Processing the Layer 4 payload in the packet means that the router will treat the payload of the packet as a message to itself and parse it according to the value of the NextHdr field. Such messages include Traceroute Requests (see 'SCMP/Traceroute request' in [I-D.dekater-scion-controlplane]).

Setting multiple router alert flags on a path SHOULD be avoided. This is because the router for which the corresponding Router Alert flag is set to "1" may process the request without further forwarding it along the path. Use cases that require multiple routers/hops on the path to process a packet SHOULD rely on a hop-by-hop extension (see Section 2.4).

2.3.3. One-Hop Path Type

Bootstrapping beaconing between neighboring ASes relies on the OneHopPath path type (PathType=2). This is necessary as neighbor ASes do not have a forwarding path before beaconing is started.

A one-hop path has exactly one Info Field and two Hop Fields. Any entity with access to the AS forwarding key can create a valid info and Hop Field as described in Section 2.3.2.3 and Section 2.3.2.4 respectively. The second Hop Field is created by the ingress SCION border router of the neighboring AS while processing the one-hop path. The appropriate Hop Field can be processed by a border router based on the source and destination address. In this context, the following rules apply:

* At the source endpoint AS, _CurrHF := 0_.

* At the destination endpoint AS, _CurrHF := 1_.

Upon receiving a packet containing a one-hop path, the ingress border router of the destination AS fills in the ConsIngress field in the second Hop Field of the one-hop path with the ingress interface ID. It sets the ConsEgress field to the unspecified value 0, ensuring the path cannot be used beyond the destination AS. Then it calculates and appends the appropriate MAC for the Hop Field.

2.3.4. Path Reversal

When a destination endpoint receives a SCION packet, it MAY use the path information in the SCION header for sending the reply packets. To reverse a path, the destination endpoint MUST perform the following steps:

1. Reverse the order of the Info Fields;

2. Reverse the order of the Hop Fields;

3. For each Info Field, negate the construction direction flag C; do not change the accumulator field Acc.

4. In the PathMetaHdr field:

    * Set the CurrINF and CurrHF to "0".

    * Reverse the order of the non-zero SegLen fields.

Note that the destination endpoint, upon receiving a first packet, is not aware of the path MTU. When using a reversed path, it should use a mechanism to estimate its MTU (e.g. MTU discovery or estimate MTU from the largest packet received).

2.4. Extension Headers

SCION provides two types of extension headers:

* The Hop-by-Hop Options header is used to carry OPTIONAL information that MAY be examined and processed by every SCION router along a packet’s delivery path. The Hop-by-Hop Options header is identified by value "200" in the NextHdr field of the SCION common header (see Section 2.1).

* The End-to-End Options header is used to carry OPTIONAL information that MAY be examined and processed by the sender and/or the receiving endpoints of the packet. The End-to-End Options header is identified by value "201" in the NextHdr field of the SCION common header (see Section 2.1).

If both headers are present, the Hop-by-Hop Options header MUST come before the End-to-End Options header.

Note: The SCION extension headers are defined and used based on and similar to the IPv6 extensions as specified in Section 4 of [RFC8200]. The SCION Hop-by-Hop Options header and End-to-End Options header resemble the IPv6 Hop-by-Hop Options Header (section 4.3 in the RFC) and Destination Options Header (section 4.6) respectively.

The SCION Hop-by-Hop Options and End-to-End Options headers are aligned to 4 bytes and have the following format:

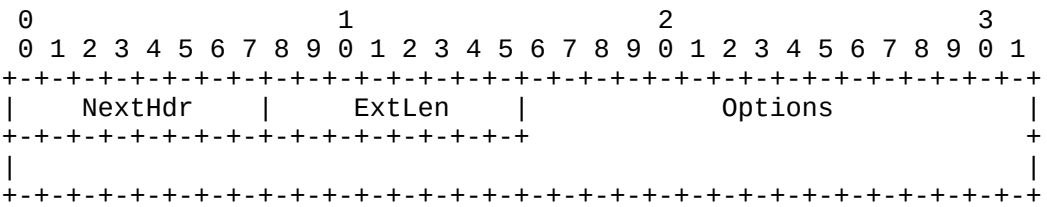

Figure 13: Extension headers: Options header

* NextHdr: Unsigned 8-bit integer. Identifies the type of header immediately following the Hop-by-Hop/End-to-End Options header. Values of this field respect the Assigned SCION Protocol Numbers (see also Appendix "Assigned SCION Protocol Numbers").

* ExtLen: 8-bit unsigned integer. The length of the Hop-by-hop or End-to-end options header in 4-octet units, not including the first 4 octets. That is: ExtLen = uint8(((L + 2) / 4) - 1), where L is the size of the header in bytes, assuming that L + 2 is a multiple of 4.

* Options: This is a variable-length field. The length of this field MUST be such that the complete length of the Hop-by-Hop/End-to-End Options header is an integer multiple of 4 bytes. This can be achieved by using options of type 0 or 1 (see Table 4). The Options field contains one or more Type-Length-Value (TLV) encoded options. For details, see Section 2.4.1.

2.4.1. Options Field

The Options field of the Hop-by-Hop Options and the End-to-End Options headers carries a variable number of options that are type-length-value (TLV) encoded. Each TLV-encoded option has the following format:

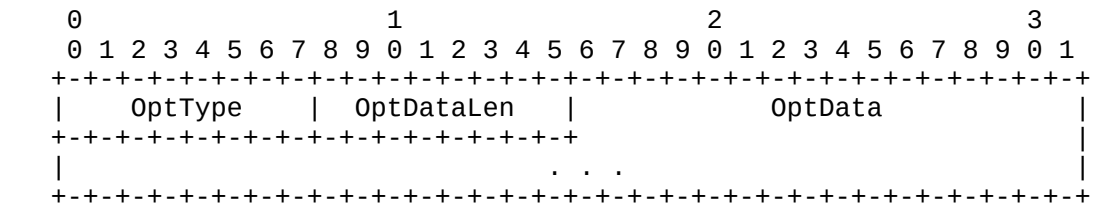

Figure 14: Options field: TLV-encoded options

* OptType: 8-bit identifier of the type of option. The following option types are assigned to the SCION HBH/E2E Options header:

Decimal	Option Type
0	Pad1 (see Section 2.4.1.1)
1	PadN (see Section 2.4.1.2)
2	SCION Packet Authenticator Option. Only used by the End-to-End Options header (experimental).
253	Used for experimentation and testing
254	Used for experimentation and testing
255	Reserved

Table 5: Option types of SCION Options header

* OptDataLen: Unsigned 8-bit integer denoting the length of the OptData field of this option in bytes.

* OptData: Variable-length field. Option-type specific data.

The options within a header MUST be processed strictly in the order they appear in the header. This is to prevent a receiver from scanning through the header looking for a specific option and processing this option prior to all preceding ones.

Individual options may have specific alignment requirements to ensure that multibyte values within the OptData fields have natural boundaries. The alignment requirement of an option is specified

using the notation "xn+y". This means that the OptType MUST appear at an integer multiple of x bytes from the start of the header, plus y bytes. For example:

* 2n: means any 2-bytes offset from the start of the header.

* 4n+2: means any 4-bytes offset from the start of the header, plus 2 bytes.

There are two padding options to align subsequent options and to pad out the containing header to a multiple of 4 bytes in length. For details, see below. All SCION implementations MUST recognize these padding options.

2.4.1.1. Pad1 Option

Alignment requirement: none.

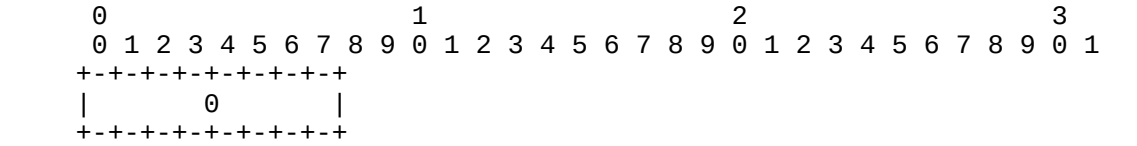

Figure 15: TLV-encoded options - Pad1 option

Note: The format of the Pad1 option is a special case - it does not have length and value fields.

The Pad1 option is used to insert 1 byte of padding into the Options field of an extension header. If more than one byte of padding is required, the PadN option MUST be used.

2.4.1.2. PadN Option

Alignment requirement: none.

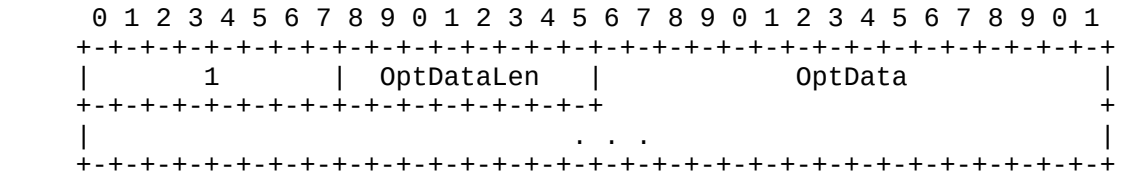

Figure 16: TLV-encoded options - PadN option

The PadN option is used to insert two or more bytes of padding into the Options field of an extension header. For N bytes of padding, the OptDataLen field contains the value N-2, and the OptData consists of N-2 zero-valued bytes.

2.5. Pseudo Header for Upper-Layer Checksum

The SCION Data Plane does not provide payload integrity protection, as further clarified in Section 7.2.2. Should any transport or other upper-layer protocols compute a checksum of the SCION header, then they SHOULD use the following pseudo header:

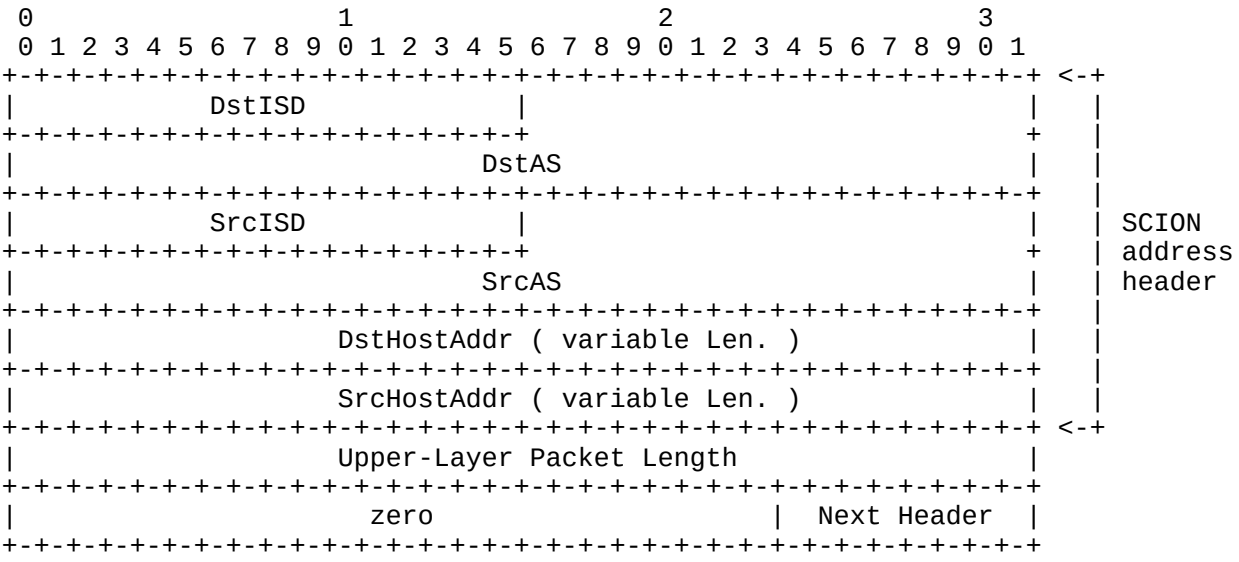

Figure 17: Layout of the pseudo header for the upper-layer checksum

* DstISD, SrcISD, DstAS, SrcAS, DstHostAddr, SrcHostAddr: These values are taken from the SCION address header.

* Upper-Layer Packet Length: The length of the upper-layer header and data. Some upper-layer protocols define headers that carry the length information explicitly (e.g. UDP) and this information is used as the upper-layer packet length in the pseudo header for these protocols. The remaining protocols, which do not carry the length information directly, use the value from the PayloadLen field in the SCION common header, minus the sum of the extension header lengths.

* Next Header: The protocol identifier associated with the upper-layer protocol (e.g. 17 for UDP - see also Appendix "Assigned SCION Protocol Numbers"). This field can differ from the NextHdr field in the SCION common header, if extensions are present.

This pseudo-header is used in current implementations of UDP on top of SCION. However, as checksums across layers are not recommended their use is discouraged in future revisions.

3. Life of a SCION Data Packet

This section describes the life of a SCION packet: how it is created at its source endpoint, passes through a number of SCION routers, and finally reaches its destination endpoint. It is assumed that both source and destination are native SCION endpoints (i.e. they both run a native SCION network stack).

This example illustrates an intra-ISD case - i.e. all communication happening within a single ISD. As the sample ISD only consists of one core AS, the end-to-end path only includes an up-path and down-path segment. The forwarding logic is uniform across intra- and inter-ISD scenarios. An inter-ISD scenario would use an additional core path segment or a peering link.

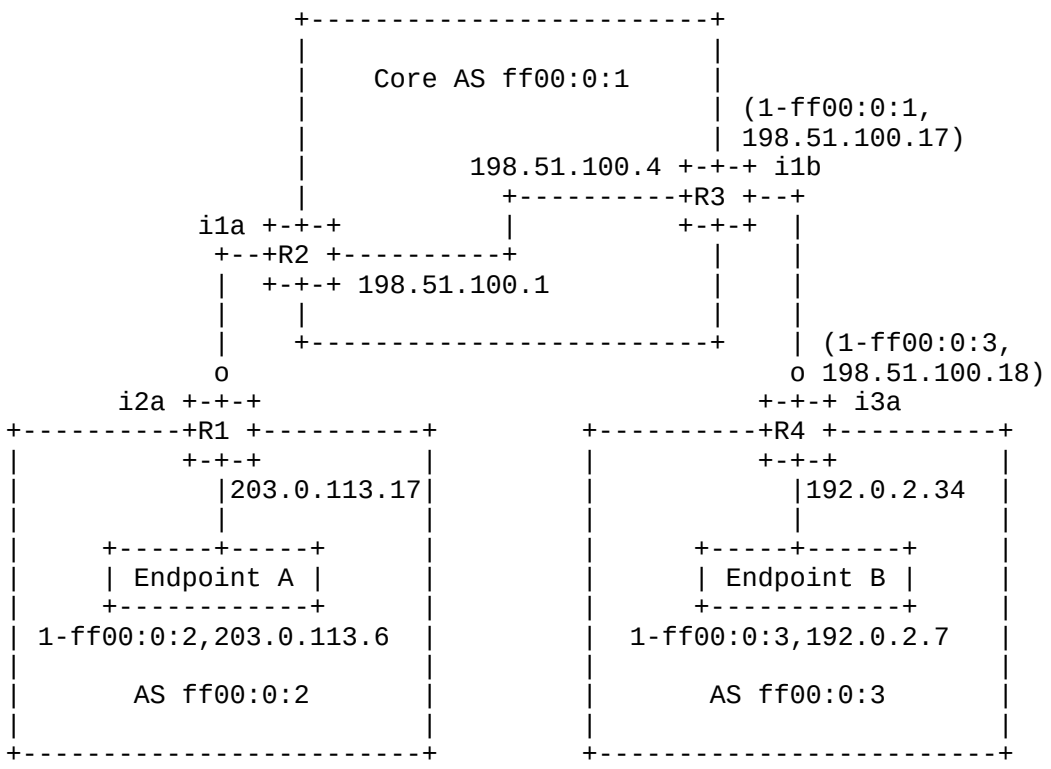

Figure 18: Example topology. AS ff00:0:1 is the core AS of ISD 1, and AS ff00:0:2 and AS ff00:0:3 are non-core ASes of ISD 1.

Based on the above topology, this example shows the life of a SCION packet sent from source at Endpoint A to destination at Endpoint B. It also shows simplified snapshots of the packet header after each on-path router.

3.1. Path Lookup and Segment Combination at Source

In this example, Endpoint A in AS ff00:0:2 wants to send a data packet to Endpoint B in AS ff00:0:3 where both are part of ISD 1. To create an end-to-end SCION forwarding path, Endpoint A first queries its own AS ff00:0:2 control service for up segments to the core AS in its ISD. The AS ff00:0:2 control service returns up segments from AS ff00:0:2 to the ISD core AS ff00:0:1. Endpoint A also queries its AS ff00:0:2 control service for a down segment from its ISD core AS ff00:0:1 to AS ff00:0:3, in which Endpoint B is located. The AS ff00:0:2 control service will return down segments from the ISD core down to AS ff00:0:3. The path segments consist of Hop Fields that carry the ingress and egress interfaces of each AS (e.g. i2a, i1a,

...), as described in detail in Section 2 - (x,y) represents one Hop Field.

Note: For more details on the lookup of path segments, see 'Path Lookup' in [I-D.dekater-scion-controlplane].

Based on its own selection criteria, Endpoint A selects the up segment (0,i2a)(i1a,0) and the down segment (0,i1b)(i3a,0) from the path segments returned by its own AS ff00:0:2 control service.

To obtain an end-to-end forwarding path from the source AS to the destination AS, Endpoint A combines the two path segments into the resulting SCION forwarding path, which contains the two Info Fields _IF1_ and _IF2_ and the Hop Fields (0,i2a), (i1a,0), (0,i1b), and (i3a,0).

Note: As this brief sample path does not contain a core segment, the end-to-end path only consists of two path segments.

Endpoint A now adds this end-to-end forwarding path to the header of the packet that it wants to send to Endpoint B, and starts transferring the packet.

3.2. Steps at Intermediate Routers

This section contains simplified snapshots of the packet header at each hop. These snapshots are depicted in tables and they show the most relevant information of the header, including the SCION path and underlay IP encapsulation for local communication.

The current Info Field (with metadata on the current path segment) in the SCION header is depicted as _italic_ in the tables. The current Hop Field, representing the current AS, is shown *bold*. The snapshot tables also include references to IP/UDP addresses. In this context, words "ingress" and "egress" refer to the direction of travel the SCION packet.

* _Step 1 -_ *A->R1*:
  The SCION Endpoint A in AS ff00:0:2 creates a new SCION packet destined for destination Endpoint B in AS ff00:0:3. Endpoint A sends the packet (for the chosen forwarding path) to the next SCION router as provided by its control service, which is in this case router R1. Endpoint A encapsulates the SCION packet into an underlay UDP/IPv4 header for the local delivery to router R1, utilizing AS ff00:0:2's internal routing protocol. The current Info Field is _IF1_. Upon receiving the packet, router R1 will forward the packet on the egress interface that Endpoint A has included into the first Hop Field of the SCION header.

Field	Value	Description
SCION addr.	SRC = 1-ff00:0:2,203.0.113.6 DST = 1-ff00:0:3,192.0.2.7	Endpoint A Endpoint B
SCION path	- _IF1_ *(0,i2a)* (i1a,0) - IF2 (0,i1b) (i3a,0)	
UDP port	SRC = 30041 DST = 30041	
IP	SRC = 203.0.113.6 DST = 203.0.113.17	Endpoint A Router R1
Link layer	SRC=A DST=R1	

Table 6: Snapshot header - step 1 - A->R1

* _Step 2 -_ *R1->R2*:
  Router R1 inspects the SCION header and considers the relevant Info Field of the specified SCION path, which is the Info Field indicated by the current Info Field pointer. In this case, it is the first Info Field _IF1_. The current Hop Field is the first Hop Field (0,i2a), which instructs router R1 to forward the packet on its interface i2a. After reading the current Hop Field, router R1 moves the pointer forward by one position to the second Hop Field (i1a,0).

  The link shown here is an example of not using a UDP/IP underlay. Although most implementations use such an encapsulation, SCION only requires link-layer connectivity. What is used for one given inter-AS link is a function of the available implementations at each end, the available infrastructure, and the joint preference of the two ASes administrators.

Field	Value	Description
SCION addr.	SRC = 1-ff00:0:2,203.0.113.6 DST = 1-ff00:0:3,192.0.2.7	Endpoint A Endpoint B
SCION path	- _IF1_ (0,i2a) *(i1a,0)* - IF2 (0,i1b) (i3a,0)	
Link layer	SRC=R1 DST=R2	

Table 7: Snapshot header - step 2 - R1 -> R2

* _Step 3 -_ *R2->R3*:
  When receiving the packet, router R2 of Core AS ff00:0:1 checks whether the packet has been received through the ingress interface i1a as specified by the current Hop Field, otherwise the packet is dropped by router R2. The router notices that it has consumed the last Hop Field of the current path segment and then moves the pointer from the current Info Field to the next Info Field _IF2_. The corresponding current Hop Field is (0,i1b), which contains egress interface i1b. The router maps the i1b interface ID to egress router R3, and encapsulates the SCION packet inside an intra-AS underlay IP packet with the address of router R3 as the underlay destination.

Field	Value	Description
SCION addr.	SRC = 1-ff00:0:2,203.0.113.6 DST = 1-ff00:0:3,192.0.2.7	Endpoint A Endpoint B
SCION path	- IF1 (0,i2a) (i1a,0) - _IF2_ *(0,i1b)* (i3a,0)	
UDP port	SRC = 30041 DST = 30041	
IP	SRC = 198.51.100.1 DST = 198.51.100.4	Router R2 Router R3
Link layer	SRC=R2 DST=R3	

Table 8: Snapshot header - step 3 - R2 -> R3

* _Step 4 -_ *R3->R4*:
  router R3 inspects the current Hop Field in the SCION header, uses interface i1b to forward the packet to its neighbor SCION router R4 of AS ff00:0:3, and moves the current hop-field pointer forward. It adds an IP header to reach router R4.

Field	Value	Description
SCION addr.	SRC = 1-ff00:0:2,203.0.113.6	Endpoint A
	DST = 1-ff00:0:3,192.0.2.7	Endpoint B
SCION path	- IF1 (0,i2a) (i1a,0)	
	- _IF2_ (0,i1b) *(i3a,0)*	
UDP port	SRC = 30041	
	DST = 30041	
IP	SRC = 198.51.100.17	Router R3
	DST = 198.51.100.18	Router R4
Link layer	SRC=R3	
	DST=R4	

Table 9: Snapshot header - step 4 - R3 -> R4

* _Step 5 -_ *R4->B*:
  SCION router R4 first checks whether the packet has been received through the ingress interface i3a as specified by the current Hop Field. Router R4 will then also realize, based on the fields CurrHF and SegLen in the SCION header, that the packet has reached the last hop in its SCION path. Therefore, instead of stepping up the pointers to the next Info Field or Hop Field, router R4 inspects the SCION destination address and extracts the endpoint address 192.0.2.7. It creates a fresh underlay UDP/IP header with this address as destination and with itself as source. The intra-domain forwarding can now deliver the packet to its destination at Endpoint B.

Field	Value	Description
SCION addr.	SRC = 1-ff00:0:2,203.0.113.6 DST = 1-ff00:0:3,192.0.2.7	Endpoint A Endpoint B
SCION path	- IF1 (0,i2a) (i1a,0) - _IF2_ (0,i1b) *(i3a,0)*	
UDP port	SRC = 30041 DST = 30041	
IP	SRC = 192.0.2.34 DST = 192.0.2.7	Router R4 Endpoint B
Link layer	SRC=R4 DST=B	

Table 10: Snapshot header - step 5 - R4 -> B

When destination Endpoint B wants to respond to source Endpoint A, it can just swap the source and destination addresses in the SCION header, reverse the SCION path, and set the pointers to the Info Fields and Hop Fields at the beginning of the reversed path (see also Section 2.3.4).

4. Path Authorization

Path authorization guarantees that data packets always traverse the network along path segments authorized by all on-path ASes in the control plane. In contrast to the IP-based Internet where forwarding decisions are made by routers based on locally stored information, SCION routers base their forwarding decisions purely on the forwarding information carried in the packet header and set by endpoints.

SCION uses cryptographic mechanisms to efficiently provide path authorization. The mechanisms are based on _symmetric_ cryptography in the form of Message Authentication Codes (MACs) in the data plane to secure forwarding information encoded in Hop Fields. This section first explains how Hop Field MACs are computed, then how they are validated as they traverse the network.

4.1. Authorizing Segments through Chained MACs

When authorizing SCION PCBs and path segments in the control plane and forwarding information in the data plane, an AS authenticates not only its own hop information but also an aggregation of all upstream hops. This section describes how this works.

4.1.1. Hop Field MAC Overview

The MAC in the Hop Fields of a SCION path has two purposes:

* Preventing malicious endpoints from adding, removing or reordering hops within a path segment created during beaconing by the control plane. In particular, preventing path splicing, i.e. the combination of parts of different valid path segments into a new and unauthorized path segment.

* Authentication of the information contained in the Hop Field itself, in particular the ExpTime, ConsIngress, and ConsEgress.

To fulfil these purposes, the MAC for the Hop Field of AS_i includes both the components of the current Hop Field HF_i and an aggregation of the path segment identifier and all preceding Hop Fields/entries in the path segment. The aggregation is a 16-bit XOR-sum of the path segment identifier and the Hop Field MACs.

When originating a PCB in the control plane, a core AS chooses a random 16-bit value as segment identifier SegID for the path segment and includes it in the PCB's Segment Info component. In the control plane, each AS_i on the path segment computes the MAC for the current HF_i, based on the value of SegID and the MACs of the preceding hop entries. Here, the full XOR-sum is computed explicitly.

For high-speed packet processing in the data plane, computing even cheap operations such as the XOR-sum over a variable number of inputs is complicated, in particular for hardware router implementations. To avoid this overhead for the MAC chaining in path authorization in the data plane, the XOR-sum is tracked incrementally for each of the path segments in a path as a separate, updatable Accumulator Field Acc. Routers update Acc by adding/subtracting only a single 16-bit value each.

When combining path segments to create a path to the destination endpoint, the source endpoint MUST also initialize the value of accumulator field Acc for each path segment. The Acc field MUST contain the correct XOR-sum of the path segment identifier and preceding Hop Field MACs expected by the first router that is traversed.

The aggregated 16-bit path segment identifier and preceding MACs prevent splicing of different path segments unless there is a collision of the Acc value among compatible path segments in an AS. See Section 7.1.3 for more details.

4.1.1.1. Hop Field MAC - Calculation

The Hop Field MAC is generally calculated at a current AS_i as follows:

* Consider a path segment with "n" hops, containing ASes AS_0, ... , AS_(n-1), with forwarding keys K0, ... , K(n-1) in this order.

* AS_0 is the core AS that created the PCB representing the path segment and that added a random initial 16-bit segment identifier SegID to the Segment Info field of the PCB.

MAC_i =
C_ki (SegID XOR MAC_0 [:2] ... XOR MAC_(i-1) [:2], Timestamp, ExpTime_i, ConsIngress_i, ConsEgress_i)

where

* ki = The forwarding key k of the current AS_i

* C_ki (...) = Cryptographic checksum C over (...) computed with forwarding key ki

* SegID = The random initial 16-bit segment identifier set by the core AS when creating the corresponding PCB

* XOR = The bitwise "exclusive or" operation

* MAC_i [:2] = The Hop Field MAC for AS_i, truncated to 2 bytes

* Timestamp = The timestamp set by the core AS when creating the corresponding PCB

* ExpTime_i, ConsIngress_i, ConsEgress_i = The content of the Hop Field HF_i

Thus, the current MAC is based on the XOR-sum of the truncated MACs of all preceding Hop Fields in the path segment as well as the path segment’s SegID - i.e. the current MAC is _chained_ to all preceding MACs. In order to effectively prevent path-splicing, the cryptographic checksum function used MUST ensure that the truncation of the MACs is non-degenerate and roughly uniformly distributed (see Section 4.1.1.3.2).

4.1.1.2. Accumulator Acc - Definition

The Accumulator Acc_i is an updatable counter introduced in the data plane to avoid the overhead caused by MAC-chaining for path authorization. This is achieved by incrementally tracking the XOR-sum of the previous MACs as a separate, updatable accumulator field Acc which is part of the path segment's Info Field InfoField in the packet header (see also Section 2.3.2.3). Routers update this field by adding/subtracting only a single 16-bit value each.

Section 4.1.1.1 provides a general formula to compute MAC_i, but at each SCION router the expression SegID XOR MAC_0 [:2] ... XOR MAC_i-1 [:2] is replaced by Acc_i. This results in the following alternative procedure for the computation of MAC_i at SCION routers:

MAC_i = C_ki (Acc_i, Timestamp, ExpTime_i, ConsIngress_i, ConsEgress_i)

During forwarding, SCION routers at each AS_i update the Acc field in the packet header so that it contains the correct input value of Acc for the next AS in the path segment to be able to calculate the MAC over its Hop Field. Note that the correct input value of the Acc field depends on the direction of travel.

The value of Acc_(i+1) is calculated based on the following definition (in the direction of beaconing):

Acc_(i+1) = Acc_i XOR MAC_i [:2]

* XOR = The bitwise "exclusive or" operation

* MAC_i [:2] = The Hop Field MAC for the current AS_i, truncated to 2 bytes

4.1.1.3. Hop Field MAC Algorithm

The algorithm used to compute the Hop Field MAC is an AS-specific choice, although the Control Services and border routers within an AS MUST use the same algorithm. Implementations MUST also support the Default Hop Field MAC algorithm as described below.

4.1.1.3.1. Default Hop Field MAC Algorithm

The default MAC algorithm is AES-CMAC ([RFC4493]) truncated to 48-bits, computed over the Info Field and the first 6 bytes of the Hop Field with flags and reserved fields zeroed out. The input is padded to 16 bytes. The _first_ 6 bytes of the AES-CMAC output are used as resulting Hop Field MAC.

Figure 19 below shows the layout of the input data to calculate the Hop Field MAC.

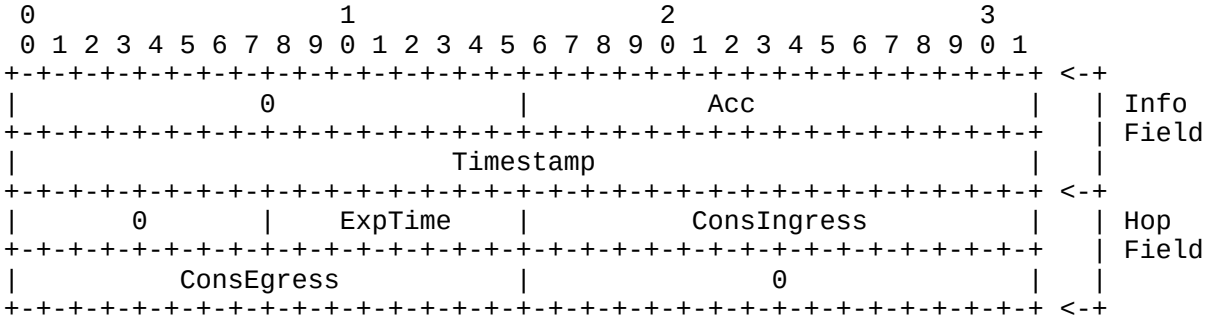

Figure 19: Input data to calculate the Hop Field MAC for the default hop-field MAC algorithm

4.1.1.3.2. Alternative Hop Field MAC Algorithms

For alternative MAC algorithms, the following requirements MUST all be met:

* The Hop Field MAC field is computed as a function of the secret forwarding key, the Acc and Timestamp fields of the Info Field, and the ExpTime, ConsIngress and ConsEgress fields of the Hop Field. Function is used in the mathematical sense that for for any values of these inputs there is exactly one result.

* The algorithm returns an unforgable 48-bit value. Unforgable specifically means "existentially unforgable under a chosen message attack" ([CRYPTOBOOK]). Informally, this means an attacker without access to the secret key has no computationally efficient means to create a valid MAC for some attacker chosen input values, even if it has access to an "oracle" providing a valid MAC for any other input values.

* The truncation of the result value to the first 16 bits of the result value:

  - is not degenerate - i.e. any small change in any input value SHOULD have an "avalanche effect" on these bits, and;

  - is roughly uniformly distributed when considering all possible input values.

This additional requirement is naturally satisfied for MAC algorithms based on typical block ciphers or hash algorithms. It ensures that the MAC chaining via the Acc field is not degenerate.

4.1.2. Peering Link MAC Computation

The Hop Field MAC computation described in Section 4.1.1.1 does not apply to a peering Hop Field - i.e. to a Hop Field that allows transiting from a child interface/link to a peering interface/link.

The reason for this is that the MACs of the Hop Fields "after" the peering Hop Field (in beaconing direction) are not chained to the MAC of the peering Hop Field but to the MAC of the main Hop Field in the corresponding AS entry. To make this work, the MAC of the peering Hop Field is also chained to the MAC of the main Hop Field. This allows for the validation of the chained MAC for both the peering Hop Field and the following Hop Fields by using the same Acc field value.

The peering Hop Field is defined as follows:

Hop Field^Peer_i = (ExpTime^Peer_i, ConsIngress^Peer_i, ConsEgress^Peer_i, MAC^Peer_i)

See [I-D.dekater-scion-controlplane] for more information.

This results in the calculation of the MAC for the peering Hop Field^Peer_i as follows:

MAC^Peer_i =
C^Peer_ki (SegID XOR MAC_0 [:2] ... XOR MAC_i [:2], Timestamp, ExpTime^Peer_i, ConsIngress^Peer_i, ConsEgress^Peer_i)

Note: The XOR-sum of the MACs in the formula of the peering Hop Field *also includes* the MAC of the main Hop Field (whereas for the calculation of the MAC for the main Hop Field itself only the XOR-sum of the _previous_ MACs is used).

4.2. Path Initialization and Packet Processing

As described in Section 2, the path header of the data plane packets only contains a sequence of Info Fields and Hop Fields without any additional data from the corresponding PCBs. The SCION path also does not contain any AS numbers - except for the source and destination ASes - and there is no field explicitly defining the type of each segment (up, core, or down). This chapter describes the required steps for the source endpoint and SCION routers to respectively craft and forward a data packet.

4.2.1. Initialization at Source Endpoint

The source endpoint MUST perform the following steps to correctly initialize a path:

1. Combine the preferred end-to-end path from the path segments obtained during path lookup.

2. Extract the Info Fields and Hop Fields from the different path segments that together build the end-to-end path to the destination endpoint. Then insert the relevant information from the Info Fields and Hop Fields into the corresponding InfoFields and Hopfields in the data packet header.

3. Each 8-byte InfoField in the packet header contains the updatable Acc field as well as a Peering flag P and a Construction Direction flag C (see also Section 2.3.2.3). As a next step in the path initialization process, the source MUST correctly set the flags and the Acc field of all InfoFields included in the path, according to the following rules:

   * The Construction Direction flag C MUST be set to "1" whenever the corresponding segment is traversed in construction direction, i.e., for down-path segments and potentially for core segments. It MUST be set to "0" for up-path segments and "reversed" core segments.

   * The Peering flag P MUST be set to "1" for up-segments and down-segments if the path contains a peering Hop Field.

   The following InfoField settings are possible, based on the following cases:

   * *Case 1*
     The path segment is traversed in construction direction and includes no peering Hop Field. It starts at the _i_-th AS of the full segment discovered in beaconing. In this case:

     - The Peering flag P = "0"

     - The Construction Direction flag C = "1"

     - The value of the Acc = Acc_i. For more details, see Section 4.1.1.2.

* *Case 2*
  The path segment is traversed in construction direction and includes a peering Hop Field (which is the first Hop Field of the segment). It starts at the _i_-th AS of the full segment discovered in beaconing. In this case:

  - The Peering flag P = "1"

  - The Construction Direction flag C = "1"

  - The value of the Acc = Acc_(i+1). For more details, see Section 4.1.1.2.

* *Case 3*
  The path segment is traversed against construction direction. The full segment has "n" hops. In this case:

  - The Peering flag P = "0" or "1" (depending on whether the last Hop Field in the up-segment is a peering Hop Field)

  - The Construction Direction flag C = "0"

  - The value of the Acc = Acc_(n-1). This is because seen from the direction of beaconing, the source endpoint is the last AS in the path segment. For more details, see Section 4.1.1.1 and Section 4.1.1.2.

4. Besides setting the flags and the Acc field, the source endpoint MUST also set the pointers in the CurrInf and CurrHF fields of the Path Meta Header PathMetaHdr (see Section 2.3.2.1). As the source endpoint builds the starting point of the forwarding, both pointers MUST be set to "0".

4.2.2. Processing at Routers

During forwarding, each SCION router verifies the path contained in the packet header. Each SCION router also MUST correctly verify or set the value of the Accumulator in the Acc field for the next AS to be able to verify its Hop Field. The exact operations differ based on the location of the AS on the path.

The processing of SCION packets for ASes where a peering link is crossed between path segments is a special case. A path containing a peering link contains exactly two path segments, one against construction direction (up) and one in construction direction (down). On the path segment against construction direction (up), the peering Hop Field is the last hop of the segment. In construction direction (down), the peering Hop Field is the first hop of the segment.

The following sections describe the tasks to be performed by the ingress and egress border routers of each on-path AS. Each operation is described from the perspective of AS_i, where i belongs to [0 ... n-1], and n == the number of ASes in the path segment (counted from the first AS in the beaconing direction).

4.2.2.1. Steps at Ingress Border Router

A SCION ingress border router MUST perform the following steps when it receives a SCION packet:

1. Check that the interface through which the packet was received is equal to the ingress interface in the current Hop Field. If not, the router MUST drop the packet.

2. If there is a segment switch at the current router, check that the ingress and egress interface links are either:

   * Both core

   * Parent-child or vice-versa

   * Peering-child or vice-versa

   Link types above are defined in ’Path and Links’ in [I-D.dekater-scion-controlplane]. This check prevents valley use of peering links or hair-pin segments.

3. Check if the current Hop Field is expired or originated in the future - i.e. the current Info Field MUST NOT have a timestamp in the future, as defined in Section 2.3.2.3. If either is true, the router MUST drop the packet.

4. If the packet traverses the path segment *against construction direction* (Construction Direction flag C = "0") perform this step:

   * *Case 1*
     The path segment includes *no peering Hop Field* (Peering flag P = "0"). In this case, the ingress border router MUST take the following step(s):

     - Compute the value of the Accumulator Acc as follows:

Acc = Acc_(i+1) XOR MAC_i
where
Acc_(i+1) = the current value of the field Acc in the current Info Field
MAC_i = the value of MAC_i in the current Hop Field representing AS_i

Note: In the case described here, the packet travels against direction of beaconing, i.e. the packet comes from AS_(i+1) and will enter AS_i. This means that the Acc field of this incoming packet represents the value of Acc_(i+1), but to compute the MAC_i for the current AS_i, we need the value of Acc_i (see Section 4.1.1.2). As the border router knows that the formula for Acc_(i+1) = Acc_i XOR MAC_i [:2] (see also Section 4.1.1.2), and because the values of Acc_(i+1) and MAC_i are known, the router will be able to recover the value Acc_i based on the aforementioned formula for Acc.

- Replace the current value of the field Acc in the current Info Field with the newly calculated value of Acc.

- Compute the MAC^Verify_i over the Hop Field of the current AS_i. For this, use the formula in Section 4.1.1.1 but replace SegID XOR MAC_0[:2] ... XOR MAC_i-1 [:2] in the formula with the value of Acc as just set in the Acc field in the current Info Field.

- If the MAC_i in the current Hop Field does not match the just calculated MAC^Verify_i, drop the packet.

- If the current Hop Field is the last Hop Field in the path segment as determined by the value of the current SegLen and other metadata in the path meta header, increment both CurrInf and CurrHF in the path meta header. Proceed with step 5.

* *Case 2*
The path segment includes a *peering Hop Field* (P = "1"), but the current hop is *not* the peering hop (i.e. the current hop is *neither* the last hop of the first segment *nor* the first hop of the second segment). In this case, the ingress border router needs to perform the steps previously described for the path segment without peering Hop Field, but the border router MUST NOT increment CurrInf and MUST NOT increment CurrHF in the path meta header. Proceed with step 5.

* *Case 3*
  The path segment includes a *peering Hop Field* (P = "1"), and the current Hop Field _is_ the peering Hop Field (i.e. the current hop is *either* the last hop of the first segment *or* the first hop of the second segment). In this case, the ingress border router MUST take the following step(s):

  - Compute MAC^Peer_i. For this, use the formula in Section 4.1.2, but replace SegID XOR MAC_0[:2] ... XOR MAC_i [:2] in the formula with the value of Acc as set in the Acc field in the current Info Field (this is the value of Acc as it comes with the packet).

  - If the MAC_i in the current Hop Field does not match the just calculated MAC^Peer_i, drop the packet.

  - Increment both CurrInf and CurrHF in the path meta header. Proceed with step 5.

5. Forward the packet to the egress border router (based on the egress Interface ID in the current Hop Field) or to the destination endpoint, if this is the destination AS.

4.2.2.2. Steps at Egress Border Router

A SCION egress border router MUST perform the following steps when it receives a SCION packet:

1. Check the settings of the Construction Direction flag C and the Peering flag P in the currently valid Info Field. The following cases are possible:

   * *Case 1*
     The packet traverses the path segment in *construction direction* (C = "1"). The path segment either includes *no peering Hop Field* (P = "0") or the path segment does include a *peering Hop Field* (P = "1"), but the current hop is not the peering hop (i.e. the current hop is *neither* the last hop of the first segment *nor* the first hop of the second segment). In this case, the egress border router MUST take the following step(s):

     - Compute MAC^Verify_i over the Hop Field of the current AS_i. For this, use the formula in Section 4.1.1.1, but replace SegID XOR MAC_0[:2] ... XOR MAC_i-1 [:2] in the formula with the value of Acc as set in the Acc field in the current Info Field.

- If the just calculated MAC^Verify_i does not match the MAC_i in the Hop Field of the current AS_i, drop the packet.

- Compute the value of Acc_(i+1). For this, use the formula in Section 4.1.1.2. Replace Acc_i in the formula with the current value of Acc as set in the Acc field of the current Info Field.

- Replace the value of the Acc field in the current Info Field with the just calculated value of Acc_(i+1).

* *Case 2*
The packet traverses the path segment in *construction direction* (C = "1") where the path segment includes a *peering Hop Field* (P = "1") and the current Hop Field _is_ the peering Hop Field (i.e. the current hop is *either* the last hop of the first segment *or* the first hop of the second segment). In this case, the egress border router MUST take the following steps:

- Compute MAC^Peer_i. For this, use the formula in Section 4.1.2, but replace SegID XOR MAC_0 [:2] ... XOR MAC_i [:2] with the value in the Acc field of the current Info Field.

- If the MAC_i in the Hop Field of the current AS_i does not match the just calculated MAC^Peer_i, drop the packet.

* *Case 3*
The packet traverses the path segment *against construction direction* (C = "0" and P = "0" or "1"). In this case, proceed with Step 3.

2. Increment CurrHF in the path meta header.

3. Forward the packet to the neighbor AS.

4.2.2.3. Effects of Clock Inaccuracy

Coarse time synchronization between an AS' s control service and its SCION routers is necessary because path segments are generated by control service instances and later used to construct data plane paths. Specifically, the timestamp in the Info Field and the expiration time of Hop Fields are used for Hop Field MAC computation at each on-path SCION router, see Section 4.1.1.1.

A segment’s originating control service and the routers that the segment refers to all have different clocks. Their differences affect the validation process:

* A fast clock at origination or a slow clock at validation will yield a lengthened expiration time for hops, and possibly an origination time in the future.

* A slow clock at origination or a fast clock at validation will yield a shortened expiration time for hops, and possibly an expiration time in the past.

This bias comes in addition to a structural delay: PCBs are propagated at a configurable interval (typically, one minute). As a result of this and the way they are iteratively constructed, PCBs with N hops may become available for path construction up to N intervals (so typically N minutes) after origination which creates a constraint on the expiration of hops. Hops of the minimal expiration time (337.5 seconds - see Section 2.3.2.4) would render useless any path segment longer than 5 hops. For this reason, it is unadvisable to create hops with a short expiration time, and the norm is 6 hours.

In comparison to these time scales, clock offsets in the order of minutes are immaterial.

Care should be taken to ensure that control plane instances and routers maintain coarse time synchronization. The specific methods used to achieve this synchronization are outside the scope of this document.

Security considerations related to this issue are discussed in [I-D.dekater-scion-controlplane].

5. Deployment Considerations

5.1. MTU

SCION requires its underlay protocol to provide a minimum MTU of 1232 bytes. This number results from 1280, the minimum IPv6 MTU as of [RFC8200]), minus 48, assuming UDP/IPv6 as underlay. Higher layer protocols such as SCMP rely only on such minimum MTU.

The MTU of a SCION path is defined as the minimum of the MTUs of the intra-AS and inter-AS links traversed by that path. The control plane disseminates such values and makes them available to the source endpoint (see ’Path MTU in [I-D.dekater-scion-controlplane]).

The MTU of each link may be discovered or administratively configured (current practice is for it to be configured). It must be less than or equal to the MTU of the link's underlay encapsulation or native link-layer in either direction.

SCION assumes that the MTUs of a path segment remains correct for the life time of that segment. This is generally a safe assumption because:

* Intra-AS network MTUs are a result of the network configuration of each AS and therefore predictable.

* Inter-AS links MTU are normally under the joint control of the administrators of the two ASes involved and therefore equally predictable.

Should the inter-AS link MTU be unpredictable (e.g. because the inter-AS link is deployed as an overlay), then the link's MTU MUST be configured statically to a conservative value. For a UDP/IP underlay, 1232 is a safe value.

5.2. Packet Fragmentation

The SCION network layer does not support packet fragmentation; not even at the source endpoint. Upper layer protocols and applications MUST comply with the MTU of the paths that they use.

SCION is agnostic to datagram fragmentation by the underlay network layer (e.g. used for intra-AS communication). Implementations SHOULD allow MTU discovery mechanisms such as [RFC4821] to be enabled in the underlay and avoid fragmentation. For inter-AS links, using a different configuration is the joint decision of the administrators of the two ASes involved. For intra-AS interfaces using a different configuration is the choice of that AS's administrator alone.

5.3. SCION IP Gateway

The SCION IP Gateway (SIG) enables IP packets to be tunneled over SCION to support communication between hosts that do not run a SCION implementation. A SIG acts as a router from the perspective of IP, whilst acting as SCION endpoint from the perspective of the SCION network. It is typically deployed inside the same AS internal network as its non-SCION hosts, or at the edge of an enterprise network. Tunneling IP traffic over SCION requires a pair of SIGs: at the ingress and egress points of the SCION network.

IP tunneling over SCION is an application from the perspective of the Data Plane and is outside the scope of this document.

More information about the reference open source SCION IP Gateway implementation can be found at [SIG].

6. Handling Link Failures

6.1. Link Failure Detection - BFD

To detect link failures quickly and reliably, SCION uses the Bidirectional Forwarding Detection (BFD) protocol ([RFC5880]) on links between SCION routers. If a router does not receive a BFD message from its peer at some regular interval, it considers the link to be down (in both directions) until messages are received again.

A SCION BFD message consists of a SCION packet with a NextHdr value of 203 (BFD/SCION) and a path type of either 00 (Empty - used on intra-AS links) or 2 (OneHopPath - used on inter-AS links). The BFD header itself is a BFD Control Header as described in [RFC5880]. More information on one-hop and empty paths is available in Section 2.3.3 and Section 2.3.1.

A SCION router SHOULD accept BFD connections from its peers and SHOULD attempt to establish BFD connections to its peers. While a link is considered to be down, a SCION router should drop packets destined to that link. In that case, it SHOULD send a notification (Section 6.2) to the originator.

6.2. Link Failure Notification - SCMP

In SCION, an intermediate router cannot change the path followed by a packet, only the source endpoint can chose a different path. Therefore, to enable fast recovery, a router SHOULD signal forwarding failures to the source, via a SCMP notification (see ’SCMP/Error messages’ in [I-D.dekater-scion-controlplane]). This allows the source endpoint to quickly switch to a different path, and the source end-point SHOULD give lower preference to the broken path. Current implementations use a negative cache with entries retained for 10s.

Sending an SCMP error notification is OPTIONAL. Endpoints should therefore implement additional mechanisms to validate or detect link down signals. To reduce exposure to denial-of-service attacks, SCION routers SHOULD employ rate limiting when sending recommended SCMP notifications (especially identical ones). Rate limit policies are up to each AS’s administrator.

7. Security Considerations

This section describes the possible security risks and attacks that the SCION Data Plane may be prone to, and how these attacks may be mitigated. It first discusses security risks that pertain to path authorization, followed by a section on other forwarding related security considerations.

7.1. Path Authorization

A central property of the SCION path-aware data plane is path authorization. Path authorization guarantees that data packets always traverse the network along path segments authorized in the control plane by all on-path ASes. This section discusses how an adversary may attempt to violate the path authorization property, as well as SCION's prevention mechanisms. Either an attacker can attempt to create unauthorized Hop Fields, or they can attempt to create illegitimate paths assembled from authentic individual Hop Fields.

The main protection mechanism is the Hop Field MAC (see Section 4.1) that authenticates the Hop Field content comprised of ingress/egress interface identifiers, creation and expiration timestamp and the preceding Hop Field MACs in the path segment. Each Hop Field MAC is computed using the secret forwarding key of the respective AS, which is shared across the SCION routers and control plane services within each AS.

7.1.1. Forwarding key compromise

For the current default MAC algorithm - AES-CMAC truncated to 48 bits - key recovery attacks from (any number of) known plaintext/MAC combinations is computationally infeasible as far as publicly known. In addition, the MAC algorithm can be freely chosen by each AS, enabling algorithmic agility for MAC computations. Should a MAC algorithm be discovered to be weak or insecure, each AS can quickly switch to a secure algorithm without the need for coordination with other ASes.

A more realistic risk to the secrecy of the forwarding key is exfiltration from a compromised router or control plane service. An AS can optionally rotate its forwarding key at regular intervals to limit the exposure after a temporary device compromise. However, such a key rotation scheme cannot mitigate the impact of an undiscovered compromise of a device.

When an AS's forwarding key is compromised, an attacker can forge Hop Field MACs and undermine path authorization. As path segments are checked for validity and policy compliance during the path discovery phase and during forwarding, routers only validate the MAC and basic validity of the current the Hop Field. Consequently, creating fraudulent Hop Fields with valid MACs allows an attacker to bypass most path segment validity checks and to create path segments that violate the AS's local policy and/or general path segment validity requirements. In particular, an attacker could create paths that include loops (limited by the maximum number of Hop Fields of a path).

Unless an attacker has access to the forwarding keys of all ASes on the illegitimate path it wants to fabricate, it will need to splice fragments of two legitimate path segments with an illegitimate Hop Field. For this, it needs to create a Hop Field with a MAC that fits into the MAC chain expected by the second path segment fragment. The only input that the attacker can vary relatively freely is the 8-bit ExpTime, but the resulting MAC needs to match a specific 16 bit Acc value. While there is a low probability of this working for a specific attempt (1/256), the attack will succeed eventually if the attacker can keep retrying over a longer time period or with many different path segment fragments.

While a forwarding key compromise and the resulting loss of path authorization is a serious degradation of SCION's routing security properties, this does not affect access control or data security for the hosts in the affected AS. Unauthorized paths are available to the attacker, but the routing of packets from legitimate senders is not affected. Such compromise can be mitigated with a forwarding key rotation.

7.1.2. Forging Hop Field MAC

Another method to break path authorization is to directly forge a Hop Field in an online attack, using the router as an oracle to determine the validity of the Hop Field MAC. The adversary needs to send one packet per guess for verification and for 6-byte MAC, an adversary would need an expected 2^47 (~140 trillion) tries to successfully forge the MAC of a single Hop Field.

As the router only checks MACs during the encoded validity period of the Hop Field, which is limited by the packet header format to at most 24 hours, these tries need to occur in a limited time period. This results in a seemingly infeasible number of ~1.6e9 guesses per second.

In the unlikely case that an online brute-force attack succeeds, the obtained Hop Field can be used until its inevitable expiration after the just mentioned 24 hour limit.

7.1.3. Path Splicing

In a path splicing attack, an adversary source endpoint takes valid Hop Fields of multiple path segments and splices them together to obtain a new unauthorized path.

Candidate path segments for splicing must have at least one AS interface in common as a connection point, and the same origination timestamp as this is directly protected by the Hop Field MAC. This can occur by chance or if the two candidate path segments were originated as the same segment that diverged and converged back.

The Hop Field MAC protects the 16-bit aggregation of path segment identifier and preceding MACs, see Section 4.1. This MAC chaining prevents splicing even in the case that the AS interface and segment timestamp match.

As the segment identifier and aggregation of preceding MACs is only 16-bits wide, a chance collision among compatible path segments can occur rarely. Successful path splicing would allow an attacker to briefly use a path that violates an ASes path policy - e.g. making a special transit link available to a customer AS that is not billed accordingly, or violate general path segment validity requirements.

In particular, a spliced path segment could traverse one or multiple links twice. However, creating a loop traversing a link an arbitrary number of times would involve multiple path splices and therefore multiple random collisions happening simultaneously, which is exceedingly unlikely. A wider security margin against path splicing could be obtained by increasing the width of the segment identifier / Acc field, e.g. by extending it into the 8-bit reserved field next to it in the Info Field.

7.2. On-Path Attacks

When an adversary sits on the path between the source and destination endpoint, it is able to intercept the data packets that are being forwarded and would allow the adversary to hijack traffic onto a path that is different from the intended one selected by the source endpoint. Possible on-path attacks in the data plane are modifications of the SCION path header and SCION address header, or by simply dropping packets. This kind of attack generally cannot be prevented, although an endpoint can use SCION’s path awareness to immediately select an alternate path if available.

7.2.1. Modification of the Path Header

An on-path adversary could modify the SCION path header and replace the remaining part of path segments to the destination with different segments. Such replaced segments must include authorized segments as otherwise the packet would be simply dropped on its way to the destination.

The already traversed portion of the current segment and past segments can also be modified by the adversary (e.g. by deleting and adding valid and invalid Hop Fields). On reply packets from the destination, the adversary can transparently revert the changes to the path header again. For example, if an adversary M is an intermediate AS on the path of a packet from A to B, then M can replace the packet's past path (leading up to, but not including M) where the new path may not be a valid end-to-end path. However, when B reverses the path and sends a reply packet, that packet would go via M which can then transparently change the invalid path back to the valid path to A. In addition, the endpoint address header can also be modified.

Modifications of the SCION path and address header can be discovered by the destination endpoint by a data integrity protection system. Such a data integrity protection system, loosely analogous to the IPsec Authentication Header, exists for SCION but is out of scope for this document. This is described as the SCION Packet Authentication Option (SPAO) in [CHUAT22].

Moreover, packet integrity protection is not enough if there are two colluding adversaries on the path who can forward the packet between them using a different path than selected by the source endpoint. The first on-path attacker remodels the packet header arbitrarily, and the second on-path attacker changes the path back to the original source-selected path, such that the integrity check by the destination endpoint succeeds. However, such an attack is of little value. An on-path adversary may inspect/copy/disrupt its traffic without diverting it away from the sender-chosen path. For this reason proof-of-transit, which would be required to detect such an attack, has marginal benefit in the context of SCION and it is not in scope for this document.

7.2.2. Payload Integrity

An on-path attacker can modify the payload of a SCION packet. Existing higher layer protocols can easily defend against such an attack without any cooperation by the SCION network. For that reason, payload integrity is not in scope for this specification. However, there exists a proposal for an experimental extension (SPAO) to authenticate addresses, provide integrity protection for payloads, and replay protection. This is still very experimental and it not used in the production network.

7.3. Off-Path Attacks

SCION's path awareness limits the abilities of an off-path adversary to influence forwarding in the data plane. Once a packet is en-route it will follow its determined path regardless of the actions of the adversary. An adversary can attempt to disrupt the connectivity of the path by flooding a link with excessive traffic (see Section 7.4 below), but after detecting congestion, the endpoint can switch to another non-congested path for subsequent packets.

7.4. Volumetric Denial of Service Attacks

An adversary can attempt to disrupt the connectivity of a network path by flooding a link with excessive traffic. In this case, the endpoint can switch to another non-congested path for subsequent packets.

SCION provides protection against certain reflection-based DoS attacks. Here, the adversary sends requests to a server with the source address set to the address of the victim, and the server will send a reply that is typically larger than the request to the victim. This can be prevented in SCION as long as the attacker and the victim are located in different ASes as the reply packets are simply returned along reversed path to the actual sender regardless of the source address information. Thus, the reply packets will be forwarded to the attacker's AS where they will be discarded because the destination AS does not match.

However, the path choice of the endpoint may possibly be exploited by an attacker to create intermittent congestion with a relatively low send rate. The attacker can exploit the latency differences of the available paths, sending at precisely timed intervals to cause short, synchronized bursts of packets near the victim.

Note SCION does not protect against two other types of DoS attacks, namely transport protocol attacks and application layer attacks. Such attacks are out of SCION's scope although additional information contained in the SCION header enables more targeted filtering - e.g. by ISD, AS or path length.

8. IANA Considerations

This document has no IANA actions.

The ISD and SCION AS number are SCION-specific numbers. They are allocated by the SCION Association (see [ISD-AS-assignments]).

9. References

9.1. Normative References

[I-D.dekater-scion-controlplane]
de Kater, C., Rustignoli, N., and S. Hitz, "SCION Control Plane", Work in Progress, Internet-Draft, draft-dekater-scion-controlplane-13, 4 December 2025, <https://datatracker.ietf.org/doc/html/draft-dekater-scion-controlplane-13>.

[I-D.dekater-scion-pki]
de Kater, C., Rustignoli, N., and S. Hitz, "SCION Control Plane PKI", Work in Progress, Internet-Draft, draft-dekater-scion-pki-10, 6 September 2025, <https://datatracker.ietf.org/doc/html/draft-dekater-scion-pki-10>.

[POSIX.1-2024]
"Standard for Information Technology--Portable Operating System Interface (POSIX) Base Specifications, Issue 8", 2024, <https://pubs.opengroup.org/onlinepubs/9799919799/basedefs/V1_chap04.html>.

[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>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/rfc/rfc2474>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/rfc/rfc3168>.

[RFC4493] Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The AES-CMAC Algorithm", RFC 4493, DOI 10.17487/RFC4493, June 2006, <https://www.rfc-editor.org/rfc/rfc4493>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/rfc/rfc5880>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/rfc/rfc6437>.

[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>.

9.2. Informative References

[CHUAT22] Chuat, L., Legner, M., Basin, D., Hausheer, D., Hitz, S., Mueller, P., and A. Perrig, "The Complete Guide to SCION", ISBN 978-3-031-05287-3, 2022, <https://doi.org/10.1007/978-3-031-05288-0>.

[CRYPTOBOOK] Boneh, D. and V. Shoup, "A Graduate Course in Applied Cryptography", 2023, <https://toc.cryptobook.us/>.

[ISD-AS-assignments] "SCION Registry", 2026, <http://scion.org/registry/>.

[PEREIRA2025] Pereira, J., Klenze, T., Giampietro, S., Limbeck, M., Dionysios Spiliopoulos, Wolf, F., Eilers, M., Sprenger, C., Basin, D., Mller, P., and A. Perrig, "Protocols to Code: Formal Verification of a Secure Next-Generation Internet Router", 2025.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/rfc/rfc1918>.

[RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, DOI 10.17487/RFC2711, October 1999, <https://www.rfc-editor.org/rfc/rfc2711>.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <https://www.rfc-editor.org/rfc/rfc4821>.

[RFC9217] Trammell, B., "Current Open Questions in Path-Aware Networking", RFC 9217, DOI 10.17487/RFC9217, March 2022, <https://www.rfc-editor.org/rfc/rfc9217>.

[RFC9473] Enghardt, R. and C. Krhenbhl, "A Vocabulary of Path Properties", RFC 9473, DOI 10.17487/RFC9473, September 2023, <https://www.rfc-editor.org/rfc/rfc9473>.

[SCIONLAB] Kown, J., Garca-Pardo, J., Legner, M., Wirz, F., Frei, M., Hausheer, D., and A. Perrig, "SCIONLAB - A Next-Generation Internet Testbed", 2020, <https://ieeexplore.ieee.org/abstract/document/9259355>.

[SCIONLAB_WEBSITE] "SCIONLab website", 2024, <https://www.scionlab.org/>.

[SIG] Anapaya, ETH, and SCION, "SCION IP Gateway Documentation", 2024, <https://docs.scion.org/en/latest/sig.html>.

Acknowledgments

Many thanks go to Harald Alvestrand (Google), Joel Halpern (Ericsson), Michael McBride (Futurewei), Ron Bonica (Juniper), Brian Trammel (Google) for reviewing this document. We also thank Matthias Frei (SCION Association), Juan A. Garcia Prado (ETH Zurich) and Kevin Meynell (SCION Association), Adrian Perrig (ETH Zurich) for providing inputs to this document. We also thank the Information Security Group at ETH Zurich for their inputs based on their formal verification work of the SCION open source router implementation [PEREIRA2025]. Finally, we are indebted to the SCION development teams of Anapaya, ETH Zurich, and SCION Association for their practical knowledge and for the documentation about the SCION Data Plane, as well as to the authors of [CHUAT22] - the book is an important source of input and inspiration for this draft.

Deployment Testing: SCIONLab

SCIONLab is a global research network that is available to test the SCION architecture. You can create and use your ASes using your own computation resources which allows you to gain real-world experience of deploying and managing a SCION network.

More information can be found at [SCIONLAB_WEBSITE] and in the [SCIONLAB] paper.

Assigned SCION Protocol Numbers

This appendix lists the assigned SCION protocol numbers.

Considerations

SCION attempts to take the IANA's assigned Internet protocol numbers into consideration. Widely employed protocols have the same protocol number as the one assigned by IANA. SCION specific protocol numbers start at 200.

The protocol numbers are used in the SCION header to identify the upper layer protocol.

Assignment

Decimal	Keyword	Protocol
0-5		Unassigned
6	TCP/SCION	Transmission Control Protocol over SCION
7-16		Unassigned
17	UDP/SCION	User Datagram Protocol over SCION
18-199		Unassigned
200	HBH	SCION Hop-by-Hop Options
201	E2E	SCION End-to-End Options
202	SCMP	SCION Control Message Protocol
203	BFD/SCION	BFD over SCION
204-252		Unassigned

253		Use for experimentation and testing
254		Use for experimentation and testing
255		Reserved

Table 11: The assigned SCION protocol numbers

Change Log

Changes made to drafts since ISE submission. This section is to be removed before publication.

draft-dekater-scion-dataplane-10

* Add normative reference to POSIX time and clarify timestamp behavior at wraparound
* Clarify distinction between SCION ASes and BGP ASes through the text
* Figure 1: split into two smaller figures to fit in a single page
* Figure 9 (Path construction example): shorten and remove superfluous AS chain
* Configuration: clarify text on intra vs inter-domain interface id mappings
* Remove unused informative reference to I-D.dekater-panrg-scion-overview, to RFC5280, and to Anapaya's ISD assignments, since they are taken over by SCION Association in 2026
* Overall review and wording polish

draft-dekater-scion-dataplane-09

* Intro: remove duplicated motivation and component description and add a reference to the same text in -controlplane
* Clarify coarse time synchronization requirement between routers and control services and add reference to -controlplane security considerations

draft-dekater-scion-dataplane-08

* Small clarifications and nits (e.g, replace RFC2460 reference with more recent RFC8200)
* Life of a SCION Data Packet: improve clarity in text and tables
* Remove use of decimal notation in tables 3 and 4

draft-dekater-scion-dataplane-07

* Clarify MTU of reversed paths and MAC algorithm
* Fix and reduce nested indentations in "Steps at Ingress Border Router"
* Reference formal verification work and acknowledge reviewers
* Nits, improve figure 2

draft-dekater-scion-dataplane-06

* Figures: redraw and add aasvg version when possible
* Clarify 0 as "unspecified" Interface ID
* Use ASes within the documentation range in examples
* Remove one-hop path type figure

draft-dekater-scion-dataplane-05

* Abstract: mention goal and that document is not an Internet Standard

draft-dekater-scion-dataplane-04

* Moved SCMP specification to draft-dekater-scion-controlplane

draft-dekater-scion-dataplane-03

Major changes:

* Introduction: clarified document goal and added Figure showing SCION Header within the stack
* Added section with SCMP specification

* Added section on Handling Link Failures and BFD

* Added sections on MTU and fragmentation

* Clarified router checks in Processing at Routers

* Security Considerations: add section on Payload Modifications

Minor changes:

* Added short section mentioning SCION IP Gateway

* Clarified the router alert flags and relationship to the ConsIngress/Egress fields.

* Clarifications in the SCION Header Specification section (router alert flags, service addresses, one-hop paths, text clarifications, validity of peering links)

* Added mention of why proof of transit is not needed.

* Rename flow ID to Flow Label and document by reference to [RFC6437].

* Added reference to SCIONLab as a testbed for implementors

* Added J. C. Hugly as author.

* Introduced this change log

* Clarify addressing and avoid confusing claim of communication between two endpoints with the same IP (section 1.3.1)

draft-dekater-scion-dataplane-02

Major changes:

* Added overview of SCION components to Introduction section.

* Introduced AES-CMAC as default MAC algorithm and elaborated on MAC chaining and path splicing.

* Added section to describe Effects of Clock Inaccuracy / time synchronization requirements

* Added section to describe required router Configuration

* Added service field table.

Minor changes:

* Removed forward references.

* General edits to make terminology consistent, remove duplication and rationalize text.

* Added and capitalized RFC2119 compliant terminology.

* Clarified implications of AS forwarding key compromise and path splicing in security considerations

* Clarified the computation of ExtLen.

Authors' Addresses

Corine de Kater
Independent
Email: c_de_kater@gmx.ch

Nicola Rustignoli
SCION Association
Email: nic@scion.org

Jean-Christophe Hugly
Independent
Email: jice@vwaty.com

Samuel Hitz
Anapaya Systems
Email: hitz@anapaya.net