SPRING
Internet-Draft
Intended status: Standards Track
Expires: 19 June 2026

F. Yang
X. Zhang
China Mobile
C. Lin
New H3C Technologies
H. Zhang
Tsinghua University

16 December 2025

SRv6 Path Verification
draft-yang-spring-srv6-verification-02

Abstract

SRv6 is being rapidly deployed and is currently primarily used in trusted-domain backbone networks. However, we have also observed that SRv6 is beginning to extend toward end-user devices, e.g., in SD-WAN deployments. SD-WAN can be deployed in third-party clouds or at customer sites, causing the physical boundary of SRv6 to become blurred. This introduces certain security risks, such as packet injection and path manipulation attacks. Section 6 of [I-D.draft-ietf-spring-srv6-security] identifies these risks as well, including Section 6.2.1 on Modification Attacks and Section 6.2.3 on Packet Insertion. This proposal mitigates these risks by enhancing the HMAC mechanism defined in [RFC8754].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 19 June 2026.

Copyright Notice

Table of Contents

1. Introduction

SRv6 is being rapidly deployed and is currently primarily used in trusted-domain backbone networks. However, we have also observed that SRv6 is beginning to extend toward end-user devices, e.g., in SD-WAN deployments. SD-WAN can be deployed in third-party clouds or at customer sites, causing the physical boundary of SRv6 to become blurred. This introduces certain security risks, such as packet injection and path manipulation attacks. Section 6 of [I-D.draft-ietf-spring-srv6-security] identifies these risks as well, including Section 6.2.1 on Modification Attacks and Section 6.2.3 on Packet Insertion. This proposal mitigates these risks by enhancing the HMAC mechanism defined in [RFC8754].

[RFC8754] describes how to use the HMAC TLV to verify the integrity and authenticity of the SRH(Segment Routing Header) during the transmission process, and to prevent the SRH from being maliciously tampered with or forged. Although the HMAC mechanism specified in RFC 8754 can verify the integrity of the entire SID List, if we want to force the SRv6 endpoints the packet must pass through during forwarding, it is necessary to retain some information each time the packet passes through an SRv6 endpoint. This draft proposes an enhancement to HMAC specificed by RFC 8754 that provides the capability to enforce the packet's forwarding path to go through all

or certain SRv6 endpoints in the SID List. Meanwhile, the SRv6 HMAC mechanism performs end-to-end cryptographic verification of the entire IPv6 header and SRH header, which significantly increases the processing performance and storage overhead of forwarding chips, making it challenging to implement in practical commercial deployments.

This document proposes a path verification mechanism for SRv6, which adopts a hop-by-hop cryptographic computation on the destination segment identifier at each node, combined with an end-to-end verification at the last hop. Although the HMAC mechanism specified in RFC 8754 can verify the integrity of the entire SID List, if we want to force the SRv6 endpoints the packet must pass through during forwarding, it is necessary to retain some information each time the packet passes through an SRv6 endpoint. This draft proposes an enhancement to HMAC specificed by RFC 8754 that provides the capability to enforce the packet's forwarding path to go through all or certain SRv6 endpoints in the SID List. And this approach also significantly reduces the processing overhead associated with hop-by-hop path verification.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Process

The improved SRv6 path verification mechanism proposed in this document follows the processing flow at the head node, intermediate nodes, and tail nodes as described below:

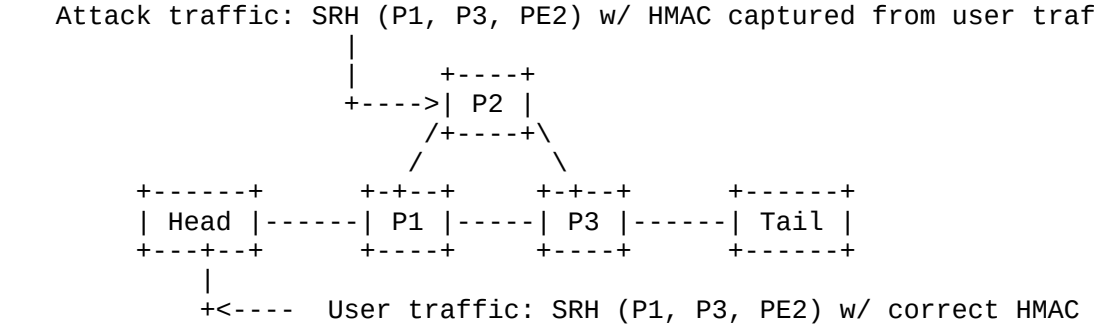


Figure 1: Example topo

Head Node:

The head node sends an IPv6/SRv6 packet. It encrypts the destination segment identifier (i.e., the SID of the first intermediate node) using a predefined encryption algorithm (e.g., HMAC, CRC, or other generic algorithms) and a pre-shared key, generating verification information 1. This verification information 1 is then inserted into a specified field of the packet (e.g., the Segment Routing Header (SRH) label field, SRH TLV field, path segment field, or IPv6 extension header), In this document, it is assumed that the mechanism is implemented by extending the "SRv6 SID Verify TLV" and incorporating it into the SRH (Segment Routing Header). The packet, now containing verification information 1, is forwarded to the first intermediate node.

Intermediate Nodes:

The first intermediate node receives the IPv6/SRv6 packet from the head node, which includes verification information 1 and the destination segment identifier of the next hop (i.e., the SID of the second intermediate node). The intermediate node reads verification information 1 and the segment identifier of the next hop from the packet, and then encrypts the verification information 1 and the segment identifier of the next hop using the same predefined encryption algorithm and pre-shared key, respectively. It then sums up verification information 2 through a predefined operation (e.g., weighted summation), generating verification information 2, which will be inserted into the same specified field of the packet, which is then forwarded to the second intermediate node. Subsequent intermediate nodes repeat this process, sequentially propagating the combined results of their own and all preceding nodes' calculations.

Tail Node:

The tail node receives the packet from the last intermediate node, which carries the combined verification information. It will compare the combined verification information with pre-calculated path verification value. If they do not match, the packet is considered routed by unexpected path and can be discarded. If they match, the packet strictly follows the SID List carried in the packet. In case of a mismatch, tail node can compare these results with its own calculations to identify the specific node where the verification failed, enabling traceability of the verification anomaly.

In summary, the algorithm works in the following way. Define ALG_n(x) = ALG(kn, x), kn is the key for node n, and x is the SID in the destination address, and Yn is the path verification information carried by the packet and updated on each hop. Suppose the SRv6 path

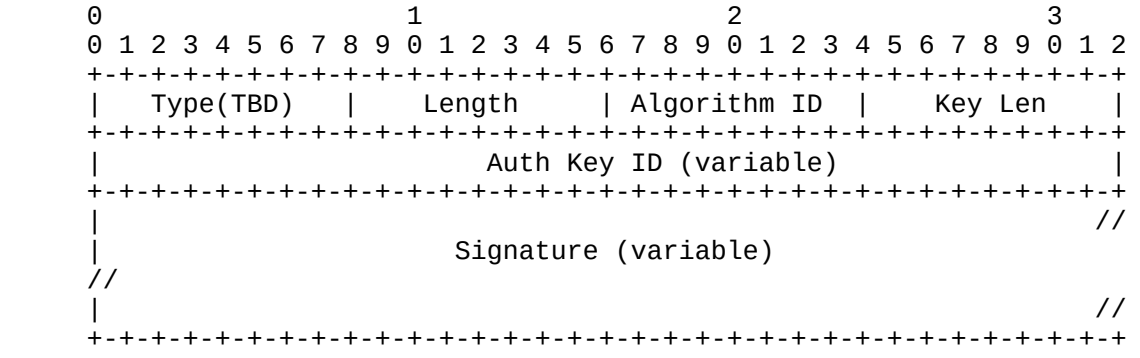
starts from Node1 and ends on Node4, the path verification information would be computed as below on each node. Node1: Y1 = ALG_1(SID_2); Node2: Y2 = ALG_2(SID_3) + ALG_2(Y1); Node3: Y3 = ALG_3(SID_4) + ALG_3(Y2); Node4: Y4 = ALG_3(SID_4) + ALG_4(y3). Optionally, on last hop node, if the verication failed it can send the packet to the SDN controller. Because Yn and ALG_n(x) is known to SDN controller, it can identify which nodes has been bypassed.

In this way, the intermediate nodes specified by in the SID list will not be allowed to be bypassed since every hop will have fingerpint in the Yn.

3. Extensions

3.1. SRv6 SID Verify TLV

A new SRv6 SID Verify TLV is requested from "Segment Routing Header TLVs" in this document.



Type (1 octets): TBD, SRv6 SID Verify TLV

Length (1 octets): The length of the variable-length data in bytes.

Algorithm ID(1 octets): The ID of encryption Algorithm.

Key Len(1 octet): Length of pre-shared

Auth Key ID: pre-shared key to encrypt the SID.

Signature: encrypted SID data, variable, in multiples of 8 octets.

Figure 2: SRv6 SID Verify TLV

4. IANA Considerations

4.1. SRv6 SID Verify TLV

A new SRv6 SID Verify TLV is requested from "Segment Routing Header TLVs".

Value	Description	Reference
0	SRv6 SID Verify TLV	This document

Table 1: Code Point

5. Security Considerations

This document should not affect the security of the Internet.

6. References

6.1. Normative References

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/rfc/rfc8754>.

[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>.

[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>.

6.2. Informative References

[I-D.draft-ietf-spring-srv6-security] Buraglio, N., Mizrahi, T., tongtian124, Contreras, L. M., and F. Gont, "Segment Routing IPv6 Security Considerations", Work in Progress, Internet-Draft, draft-ietf-spring-srv6-security-09, 6 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-srv6-security-09>.

Authors' Addresses

Feng Yang
China Mobile
China
Email: yangfeng@chinamobile.com

Xiaoqiu Zhang
China Mobile
China
Email: zhangxiaoqiu@chinamobile.com

Changwang Lin
New H3C Technologies
China
Email: linchangwang.04414@h3c.com

Han Zhang
Tsinghua University
China
Email: zhhan@tsinghua.edu.cn