Network Working Group
Internet Draft
Intended status: Standards Track
Expires: August 20, 2025

L. Dunbar
Futurewei
K. Majumdar
Oracle
S. Fluhrer
Cisco

February 20, 2025

Lightweight Authentication Methods for IP Header
draft-dunbar-ipsecme-lightweight-authenticate-00

Abstract

This document describes lightweight authentication methods to prevent malicious actors tampering with the IP encapsulation headers or metadata carried by the UPD Option Header.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on Dec 20, 2025.

Copyright Notice

Table of Contents

1. Introduction

[MULTI-SEG-SDWAN] describes scenarios and methods where an additional header (GENEVE Encapsulation [RFC8926]) is added to the encrypted payload to steer packets through underlay networks. In these scenarios, the underlay network edge nodes do not decrypt and re-encrypt the payloads. The header information is used for optimizing packet forwarding in underlay networks and, therefore, resides outside the IPsec ESP header. Authenticating these additional headers is important in certain environments to prevent malicious actors from tampering with header information.

This document outlines lightweight methods to authenticate encapsulation headers, aiming to reduce the computational resources needed for this process while ensuring security. These methods can be applied to authenticate GENEVE and SRH headers used by [MULTI-SEG-SDWAN], RFC8754, UDP option headers [MEDIA-HDR-WIRELESS], [UDP-OPTION-HDR], and other encapsulation headers.

2. Conventions used in this document

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 BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following acronyms and terms are used in this document:

AES Advanced Encryption Standard

Cloud DC: Off-Premises Data Center, managed by the third party, that hosts applications, services, and workload for different organizations or tenants.

CPE: Customer (Edge) Premises Equipment.

OnPrem: On Premises data centers and branch offices.

RR	Route Reflector.
SD-WAN	An overlay connectivity service that optimizes transport of IP Packets over one or more Underlay Connectivity Services by recognizing applications (Application Flows) and determining forwarding behavior by applying Policies to them. [MEF-70.1]
VPN	Virtual Private Network.

3. Use Cases

3.1. Multi-segment SD-WAN connected by Cloud Backbone

[MULTI-SEG-SDWAN] describes the method of using GENEVE Header to encapsulate the IPsec encrypted packets for Cloud GW to steer the packets through the Cloud backbone without the Cloud GWs to decrypt and re-encrypt the payload, as shown in the figure below. It is necessary to authenticate the GENEVE Header to prevent anyone from tampering with the information in the GENEVE header.

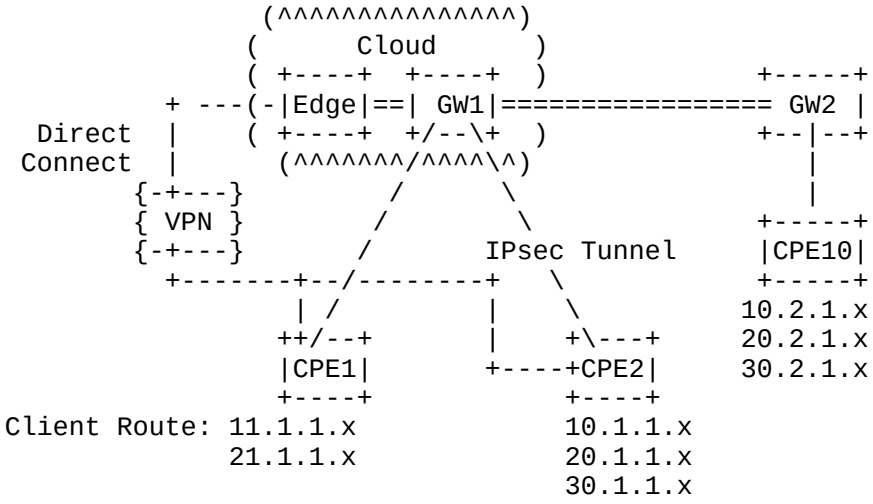

Figure 1 Multi-Segment SD-WAN via Cloud Backbone

3.2. Metadata in UDP Authentication

[MEDIA-HDR-WIRELESS] describes the scenario and method of carrying metadata in a packet's UDP Option Header [UDP-OPTION-HDR] between wireless network nodes and the application servers. The IP packet payload is already encrypted. It is necessary to authenticate the metadata to prevent anyone from tampering with it.

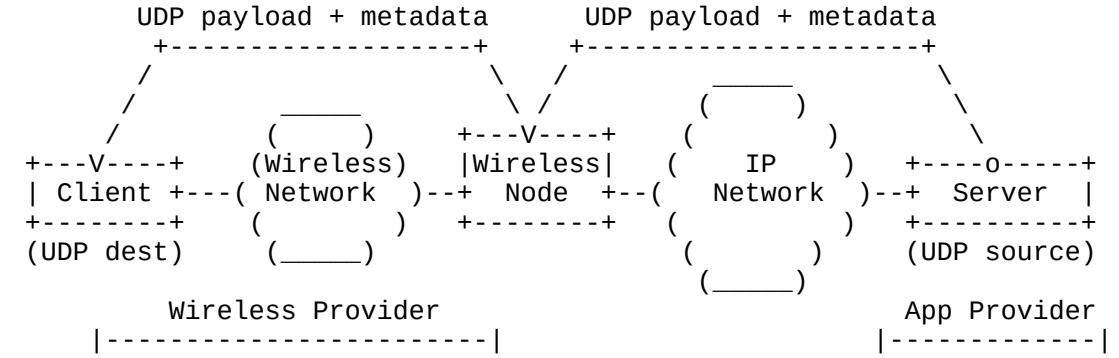

Figure 2: Media Payload and Metadata in UDP Packet

The described header authentication focuses solely on authenticating the appended metadata in the UDP Option Header. This approach differs from Section 11.9 (Authentication) of [UDP-OPTION-HDR], which authenticates the entire payload. It's important to note that the authentication method outlined in this document is designed to be lightweight.

4. Header Authentication Methods Analysis

4.1. Justification for Header Authentication

Authenticating IP packet encapsulation headers is essential for safeguarding against a variety of malicious activities, in particular:

Integrity Assurance:
  Tamper Detection: Any unauthorized modification or tampering of the headers can be detected, indicating potential malicious activity.

Security in Overlay Networks:
Overlay Network Protection: GENEVE or VXLAN are commonly used in overlay networks, where security is paramount. Authenticating encapsulation headers helps secure the overlay network by preventing attackers from manipulating routing information or injecting malicious packets into the network.

Preventing Spoofing and Injection Attacks:
Source Address Verification: Authentication ensures that the source address in the encapsulation header is genuine. This helps prevent address spoofing, ensuring that the sender's identity is verified and not manipulated by malicious actors attempting to inject unauthorized traffic.

Protection Against Man-in-the-Middle Attacks:
Routing Manipulation Prevention: In the case of SRv6, which involves source routing, authenticating headers prevents unauthorized changes to the routing information. This guards against man-in-the-middle attacks where an attacker could alter the routing path of the packet.

Preventing Denial-of-Service (DoS) Attacks:
Header Flooding Protection: Authentication adds a layer of protection against DoS attacks that flood the network with manipulated or maliciously crafted encapsulation headers. By verifying the authenticity of headers, the network can reject unauthorized or suspicious traffic.

4.2. HMAC-Based Authentication Method

HMAC uses hash functions like SHA-256 [RFC4868], which are computationally efficient and can be processed quickly by both software and hardware. This ensures minimal impact on network performance, even at high data rates, making HMAC an efficient choice for authenticating IP encapsulation packet headers, such as GENEVE or UDP option headers. [RFC2104] describes the general construction of HMAC and guides its use.

Here's are the steps:

Key Establishment:
Need a secure channel for network edges to share a secret key. This could be done manual configuration or through a secure key exchange protocol.

Additional Header Field:
Need to add a new field in the packet header, e.g., "HMAC-Auth-Val" field, to store the HMAC value. Before sending the packet, the edge node computes the HMAC of the entire header (excluding the HMAC-Auth-Val field) using the shared secret key.

Authentication Process:
When a packet is received, the recipient recalculates the HMAC of the received header using the shared key. Compare the computed HMAC with the value in the received "HMAC-Auth-Val" field.
If the values match, the header is considered authentic and has not been tampered with.

This method provides a lightweight approach for ensuring the authenticity and integrity of IP encapsulation packet headers without adding significant overhead.

4.3. Digital Signatures

Digital signatures, while effective for providing authenticity and integrity of data, have several limitations when it comes to authenticating IP encapsulation headers or UDP option headers. Here are some key reasons why digital signatures are not ideal for this purpose:

Complexity: Digital signatures involve cryptographic operations such as hashing and asymmetric encryption, which require significant computational resources.

Frequent Changes: IP encapsulation headers and UDP option headers often contain fields that change frequently as the packet traverses the network. Digital signatures would need to be recalculated for each change, which is impractical.

4.4. Other Authentication Methods

There are several alternatives to HMAC for message authentication, including CMAC (Cipher-based Message Authentication Code) [RFC4493], GMAC (Galois/Counter Mode Message Authentication Code) [RFC4106], Poly1305 [RFC8439], SIPHASH [RFC8968], BLAKE2 [RFC7693], and KMAC (Keccak Message Authentication Code) [RFC8702]. CMAC is based on AES .Advanced Encryption Standard. [RFC3826] and offers strong security in hardware-accelerated environments, while

GMAC provides high performance and is often used with AES-GCM for authenticated encryption. Poly1305, known for its speed, pairs well with ChaCha20, and SIPHASH is optimized for short inputs, making it ideal for hash table lookups. BLAKE2 is faster than traditional hash functions and offers both hashing and keyed MAC capabilities. KMAC, based on the SHA-3 sponge function, provides flexibility with variable-length keys and tags. Despite these alternatives, HMAC remains a preferred choice due to its widespread support, robust security, and efficient performance across diverse hardware and software environments, making it a reliable and well-understood option for authenticating IP encapsulation packet headers.

5. Encoding of Header Authentication Value

5.1. Analysis of HMAC Value

While the ideal HMAC value size might be 32 bytes for robust security [NIST-SP-800-107], adding an extra 32 bytes to each IP packet can significantly impact the overall packet size, potentially leading to exceeding the underlay network' MTU (Maximum Transmission Unit), fragmentation, and increased transmission overhead.

To address this challenge, a judicious compromise can be made by employing a smaller, yet still secure, shorter HMAC size, such as 4 bytes (32 bits) or 8 bytes (64 bits can be considered to append the packet header (or UDP Option Header). Shortening the HMAC reduces the packet size, conserving bandwidth and reducing processing time, which is crucial in high-speed networks and resource-constrained environments. While a truncated HMAC offers less security than a full-length HMAC, the shorter length still provides sufficient protection against common threats in scenarios where the primary goal is to ensure data integrity and authenticity without imposing significant performance penalties. Additionally, the use of a shorter HMAC is often a pragmatic choice in environments where packets are frequently modified, as it minimizes the need for recalculating the authentication code.

As stated in the [RFC2104], the authentication tag may be truncated (e.g., to 128 bits or less) as a means of reducing the packet overhead and still provide adequate security, depending on the application.

5.2. Consideration in Generating the Authentication Value

HMAC-SHA-256 [RFC4868] [RFC6234] produces a 32-byte (256-bit) output by default. Here are some methods to generate a 4-byte (32-bit) or 8-byte (64-bits) HMAC value,

- Direct Truncation: Simply take the first 4 bytes of the HMAC output.
- Hash function output: apply another hash function on the 32 bytes value from the HMAC-SHA-256 to derive 4 bytes value.

[RFC2104] suggests using the Direct Truncation method.

5.3. Authentication Value Encoding

HMAC-Auth-VAL Sub-TLV specified in this section can be appended to the IP header for the header authentication purpose.

The HMAC-Auth-VAL Sub-TLV is to carry the HMAC authentication value. The HMAC Sub-TLV must be appended as the last Sub-TLV in the encapsulation header. The entire encapsulation header, excluding the HMAC Sub-TLV, is included in computing the HMAC authentication value based on the pre-configured algorithm. The detailed Sub-TLV is specified below:

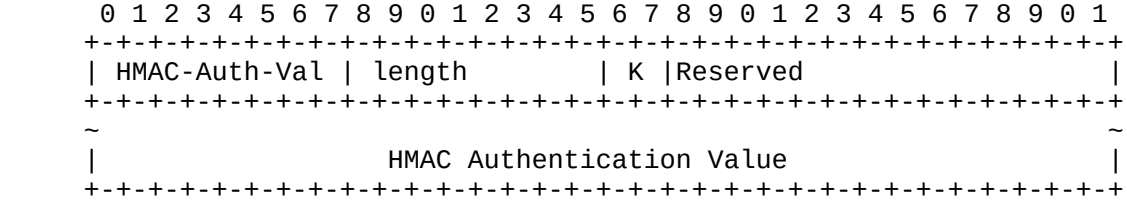

Figure 1 HMAC Sub-TLV

HMAC-Auth-Val (8 bits): HMAC Authentication Value Type = 6 (assigned by [MULTI-SEG-SDWAN]).

Length (8 bits): total length of the value field, which is the length of the HMAC Authentication Value in bytes plus 2 Reserved bytes. It is 6 bytes by default. However, in some deployments where security requirements are high, a longer authentication value can be considered.

K Flag (2 bits): indicates the index of the key used for computing the HMAC Authentication Value present in the TLV. This flag allows the recipient to identify which key from a predefined set is used to generate the HMAC, facilitating key rotation and management. The key set is assumed to be shared and agreed upon by both the sender and the receiver prior to communication.

HMAC Authentication Value: is computed including the entire encapsulation header, excluding the HMAC-auth-Val Sub-TLV, based on a pre-configured algorithm.

5.4. Selective Packet Header Authentication

Selective Packet Header Authentication (SPHA) employs a strategic method, where receiving nodes selectively authenticate a subset of packets, guided by control messages from their controller or management systems. Each packet features an Authentication Type-Length-Value (TLV), but only specific packets or flows undergo full authentication. Authenticated packets contain a genuine authentication value in the TLV Value field of the encapsulation header, while others hold dummy authentication values. However, it should be noted that the metadata itself in each GENEVE or other encapsulation is actual/new information. The control plane determines the frequency of authentication or selective flows for authentication, and the presence of an Authentication TLV on every packet conceals which packets are authentically validated. This obfuscation enhances security by preventing intermediaries from tampering with or identifying the authentic authentication values.

On the receiving end, nodes verify the authentication of headers based on a pre-configured policy, choosing only specific packets or flows for authentication. This method of selectively applying authentication effectively balances security with operational efficiency. It optimizes resource usage while maintaining robust protection against header manipulation, making it an economical choice for securing network traffic against man-in-the-middle attacks.

Furthermore, for packets that are not authenticated, the receiver can still monitor for integrity by comparing their headers, such as the flow identifier, with those of authenticated packets. This comparison helps detect unauthorized modifications, even if the packet itself was not authenticated.

This selective authentication method is particularly beneficial in environments where the computational cost of authenticating every packet header is prohibitively high, such as on small IoT devices. Moreover, in contexts where packet payloads are already encrypted, the main concern shifts to ensuring the integrity of the packet headers. The use of HMAC or Digital-Sig Sub-TLVs in SPHA provides a robust measure against header tampering by potential malicious intermediaries, thereby maintaining the integrity of packet headers.

Overall, SPHA offers a sophisticated and dynamic approach to packet authentication, effectively balancing security with efficiency and scalability.

6. Authentication Key Distribution

The lightweight authentication methods proposed in this document are for environments where there are IPsec tunnels between SD-WAN CPEs and the Cloud GW. For traffic originating from SD-WAN CPEs and terminating within the Cloud Data Center (DC), the Cloud GW decrypts the IPsec traffic. For traffic that needs to be routed via the Cloud Backbone to remote CPEs, the proposed lightweight authentication method is used.

6.1. Key Distribution Via Secure Control Plane Channel

The proposed deployment environment assumes the presence of a secure channel between the two organizations for key exchange. It also assumes a secure channel exists between the network controller and the CPEs. Other scenarios are out of the scope of this document.

Pre-shared keys are preferred over dynamic key exchange for simplicity and efficiency when possible. In scenarios demanding a higher security posture, authentication keys can be generated using a cryptographically secure random number generator to mitigate predictability and opt for shorter key lifetimes.

In the case of [MULTI-SEG-SDWAN], the Cloud Controller can own the authentication key and securely distribute it to the enterprise’s SD-WAN controller through a secure channel, such as TLS. In the case of [MEDIA-HDR-WIRELESS], the application controller can own the authentication key and securely distribute it to the wireless provider’s controller (or management system).

To enhance security, it is imperative to periodically rotate the authentication keys and incorporate key versioning information. This ensures that both the Cloud Operator's Controller and the enterprise's SD-WAN controller utilize the correct and up-to-date keys.

6.2. Key Distribution Via Secure Data Plane Tunnel

For environments with IPsec tunnels between SD-WAN CPEs and the Cloud GW, the IPsec tunnel provides a secure channel for transmitting authentication keys, ensuring protection against eavesdropping or tampering during distribution.

Additionally, the existing IPsec keys can be used as input to a key derivation function (KDF). The KDF generates unique authentication keys that are cryptographically linked to the IPsec keys but not directly exposed.

7. Control Plane Mechanism

The configuration for the frequency and selection of packets or flows that undergo real authentication is managed through a secure management channel between edge nodes and the network controller.

Options include:

- Specific flows,

- the first packet of specific flows or every 'N' packets.

When a network segment is detected to have a higher than usual probability of security risks, such as man-in-the-middle (MITM) attacks, several actions can be taken to mitigate these risks. For example:

- Increase the frequency of packets or flows to be fully authenticated,

- Move towards comprehensive authentication where all packet headers are authenticated, rather than just selective flows.

- Adding another layer of encryption for the header between CPEs and Cloud GWs.

- Use Stronger Encryption Algorithms: Ensure that the encryption algorithms used for securing data are strong (e.g., AES-256).

The secure management channel facilitates dynamic adjustments to the authentication process, accommodating varying network conditions and security needs efficiently.

This approach also enhances security by enabling the detection of malicious actors injecting traffic into the flow. This capability not only ensures the integrity and security of data flow but also acts as a deterrent against man-in-the-middle attacks, providing a robust defense mechanism against unauthorized data manipulation.

8. Frames Loss Handling

Here are some methods that can be effectively integrated with SPHA to mitigate the impact of packet loss:

- Retransmission Requests: Implement a mechanism where receiving nodes can request the retransmission of lost packets. This is particularly useful for packets that were supposed to carry valid authentication but were lost in transit. Ensuring that these packets can be retransmitted securely is crucial.

- Forward Error Correction (FEC): Use FEC techniques to send additional error-correcting code with the packets. This allows the receiver to reconstruct lost packets without needing a retransmission, which is especially useful in high-latency or unreliable network environments.

- Sequence Numbering: Assign sequence numbers to each packet as they are sent. This allows receiving nodes to detect missing packets (gaps in the sequence numbers) and can be used to trigger alerts or retransmission requests.

- Heartbeat Messages: Regularly send heartbeat or status messages that can help in identifying packet loss quickly. These messages can carry summary information about the sequence of authenticated packets, allowing for faster detection and recovery from packet loss.

- Multi-Path Transmission: Use multiple network paths to send duplicates of critical packets. This redundancy increases the likelihood that at least one copy of the packet

reaches its destination, which is useful in networks where packet loss is frequent.

- Adaptive Authentication Thresholds: Dynamically adjust the frequency of authentication based on network conditions. If packet loss is detected, the system could increase the authentication frequency to ensure that enough authenticated packets are received to maintain security.

- Time-based Reconciliation: Implement a system where packets are periodically reconciled based on their timestamps. This can help in identifying missing packets over a specific interval and ensure that data integrity is maintained over time.

Each of these methods can be tailored to fit the specific needs and constraints of the network, allowing for an effective balance between security, performance, and reliability in the face of packet loss challenges.

9. Mechanism to Handle Replay

Handling replay attacks, where a man-in-the-middle resends previously captured packet frames to disrupt or mislead the network, is crucial in maintaining the security of a network using Selective Packet Header Authentication (SPHA). Here are several strategies that can be employed to mitigate the risk of replay attacks:

- Timestamps: Include precise timestamps in each packet’s header. Receiving nodes can then check these timestamps against a permissible time window to ensure that packets are not being replayed outside of this window. This approach requires synchronized clocks between the sender and the receiver.

- Sequence Numbers: Use unique sequence numbers for each packet. This allows the receiving nodes to detect out-of-order or repeated sequences, which are indicative of replay attacks. Sequence numbers should be sufficiently large to prevent rollover during a session.

- Nonce Values: Incorporate a random or pseudo-random number (nonce) in each packet that cannot be predicted by attackers. This nonce can be verified by the receiver to ensure that each packet is fresh and not a replay.

- Challenge-Response Mechanisms: Implement a challenge-response system where the receiver sends a challenge to the sender, and the sender must include a response in subsequent packets. If a packet lacks the correct response, it can be assumed to be a replay.

- Session Keys: Use session-specific keys that change periodically or each time a new connection is established. Even if a packet is captured, it will become useless once the session key changes.

- Packet Expiry Information: Embed expiry information within packets to define how long they are valid. This can prevent older packets from being accepted by the receiver long after their intended lifespan.

- Stateful Inspection: Utilize stateful inspection at the receiving end to track the state of network connections and validate that incoming packets conform to the expected state. This can help detect and block packets that are anomalies or replays based on previous communications.

Each of these methods can be used alone or in combination to provide robust defense mechanisms against replay attacks, enhancing the security of communications in networks using SPHA.

10. Security Considerations

The HMAC provided security relies on the strength of the shared key and the effectiveness of the algorithms. Each deployment must adjust the hash algorithm and key management based on specific security requirements and considerations.

While a 32-bit signature offers efficiency advantages, especially in resource-constrained environments, it comes with security trade-offs. It is essential to carefully consider the specific security requirements of the deployment to assess whether the reduced security strength is acceptable.

10.1. AH based Integrity and Authentication

For enterprises or Cloud providers worrying about secret HMAC keys being compromised, they can add another layer of AH encryption [RFC4301] or ESP-NULL [RFC2410] [RFC6071] on top

of the IPsec encryption between the two CPEs. Both AH and ESP-NULL IPsec encryption require pairwise IPsec key management between Cloud GWs and the CPEs, therefore requiring more processing on Cloud GWs and CPEs. In addition, the AH encrypted packets can’t traverse NAT because of outer IP address changes.

11. Manageability Considerations

A robust management system is essential for handling HMAC key configurations between SD-WAN edges and Cloud GW to ensure consistency and streamline the management of authentication settings. It is imperative to guarantee the secure generation, distribution, and periodic updating of HMAC keys. Additionally, implementing a well-defined process for promptly revoking and replacing HMAC keys in response to compromises or other security incidents is necessary.

12. IANA Considerations

IANA is requested to assign the values for the following Sub-TLV types that can append to IP Header for Authentication Purpose:

- HMAC-Auth-Val Sub-TLV Type: The HMAC Sub-TLV is to carry the HMAC authentication value.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2403] C. Madson, R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH", RFC2403, Nov. 1998.

[RFC2404] C. Madson, R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC2404, Nov. 1998.

[RFC4301] S. Kent and K. Seo, "Security Architecture for the Internet Protocol", RFC4301, Dec. 2005.

[RFC4303] S. Kent, "IP Encapsulating Security Payload (ESP)". RFC4303, Dec. 2005.

[RFC4868] S. Kelly , S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC4868, May 2007.

[RFC5424] R. Gerhards, "The Syslog Protocol", RFC5424, March 2009.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8926] J. Gross, et al, "Geneve: Generic Network Virtualization Encapsulation", RFC8926, Nov 2020.

13.2. Informative References

[RFC2104] H. Krawczyk, et al, "HMAC: Keyed-Hashing for Message Authentication", RFC2104, Feb. 1997.

[MULTI-SEG-SDWAN] K. Majumdar, et al, "Multi-segment SD-WAN via Cloud DCs", draft-ietf-rtgwg-multisegment-sdwan-02, Feb, 2025.

[NIST-SP-800-107] National Institute of Standards and Technology (NIST) Special Publication 800-107 Revision 1, "Recommendation for Applications Using Approved Hash Algorithms".

[NIST-AES-GCM] National Institute of Standards and Technology (NIST) Special Publication 800-38D, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", Nov. 2007.

[RFC2410] R. Glenn and S. Kent, "The NULL encryption Algorithm and Its Use with IPsec", RFC2310, Nov. 1998.

[RFC4493] T, Iwata, et al, "The AES-CMAC Algorithm", RFC4493, June 2006.

[RFC6071] S. Frankel and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", Feb. 2011.

[RFC6234] D. Eastlake and T. Hansen, "US Secure Hash Algorithms", RFC6234, May 2011.

[MEDIA-HDR-WIRELESS] J. Kaippallimalil, et al, "Media Header Extensions for Wireless Networks", draft-kaippallimalil-tsvwg-media-hdr-wireless-05, Aug 2024.

[MEF-70.1] MEF 70.1 SD-WAN Service Attributes and Service Framework. Nov. 2021.

[UDP-OPTION-HDR] J Touch, "Transport Options for UDP", draft-ietf-tsvwg-udp-options-28, Nov. 2023.

14. Acknowledgments

Acknowledgements to Russ Housley, Paul Wouters, and Scott Fluhrer for their review, questions, and suggestions.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Linda Dunbar
Futurewei
Email: ldunbar@futurewei.com

Kausik Majumdar
Oracle
Email: kausik.majumdar@oracle.com

Scott Fluhrer
Cisco
Email: sfluhrer@cisco.com

Contributors' Addresses