Network Working Group
Internet Draft
Intended status: Standards Track
Expires: April 20, 2026

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

October 20, 2025

Lightweight Authentication for Encapsulation Header
draft-dunbar-ipsecme-lightweight-authenticate-02

Abstract

This document specifies a lightweight authentication mechanism (KeyID, anti-replay, algorithms, truncation, and keying) intended to be reused by multiple protocol profiles. Concrete profiles define where the authentication data is carried and the exact coverage rules for header fields.

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, 2020.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Many enterprises interconnect sites over the public Internet and provider clouds using encrypted tunnels (typically IPsec ESP). In "multi-segment" deployments, traffic enters a provider’s backbone at one gateway and exits at another. Because ESP hides the payload from the backbone, these intermediate "cloud gateways" still need a small amount of per-packet steering information (e.g., which egress, class, or path) to forward packets correctly and at line rate.

That steering information is conveyed in an outer encapsulation header placed outside ESP. Examples include a GENEVE header (RFC 8926) as described in [MULTI-SEG-SDWAN], an IPv6 Segment Routing Header (SRH, RFC 8754) when used for steering, or UDP options used to carry metadata. Gateways read only this outer header to make a forwarding decision; they do not decrypt or re-encrypt the payload.

Because the outer encapsulation header is not protected by ESP, it is susceptible to on-path modification. Unauthorized or accidental changes can misroute traffic, bypass policy, or degrade service. Ensuring forwarding integrity therefore requires authenticating the relevant fields of this outer header.

This document specifies a lightweight method to authenticate encapsulation headers. The method uses a compact integrity check, keyed via the control plane, that covers only non-mutable steering fields, adds minimal per-packet overhead, and avoids payload decryption. It is applicable to GENEVE, SRH used for multi-segment steering, UDP-option-based headers, and similar encapsulations.

By authenticating only the necessary outer-header fields, networks can enforce policy and protect forwarding integrity while maintaining high performance when steering encrypted traffic across cloud backbones and between segments. The mechanism complements ESP: ESP continues to provide confidentiality and integrity for the payload, while the lightweight authenticator protects the steering information carried outside ESP.

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

In [MULTI-SEG-SDWAN], GENEVE encapsulation is used to carry IPsec-encrypted packets between CPEs via Cloud GWs, enabling the Cloud Backbone to steer traffic without decrypting and re-encrypting payloads. To protect against malicious alteration of routing metadata, the GENEVE header in this

encapsulation SHOULD be authenticated so that only authorized entities can modify or insert header information. This requirement applies symmetrically to both sides. For example, CPE1 to GW1 and CPE10 to GW2 each require authentication when their connectivity to the Cloud GW traverses the Internet. The authentication protects the encapsulation header rather than the already-encrypted payload. However, if the CPE-to-Cloud GW segment is inherently secure, for example, a single Ethernet link or private line, then additional authentication of the encapsulation header is not required.

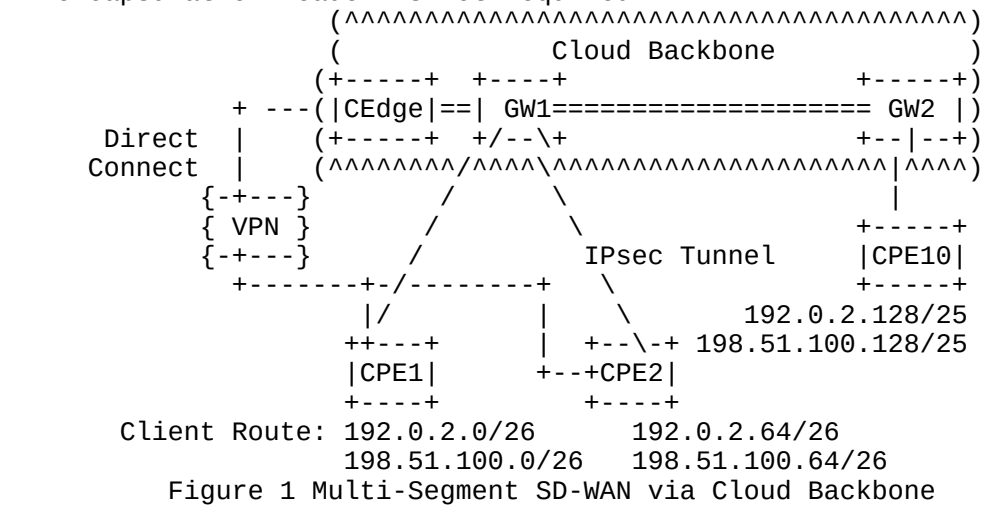

Figure 1 Multi-Segment SD-WAN via Cloud Backbone

3.2. Metadata in UDP Authentication

[MEDIA-HDR-WIRELESS] describes how metadata can be carried in a packet's UDP Option Header [UDP-OPTION-HDR] between wireless network nodes and application servers. While the IP packet payload is encrypted, the metadata in the UDP Option Header remains exposed in transit. Authenticating this metadata is essential to ensure its integrity and prevent unauthorized modification that could mislead application logic or disrupt service behavior.

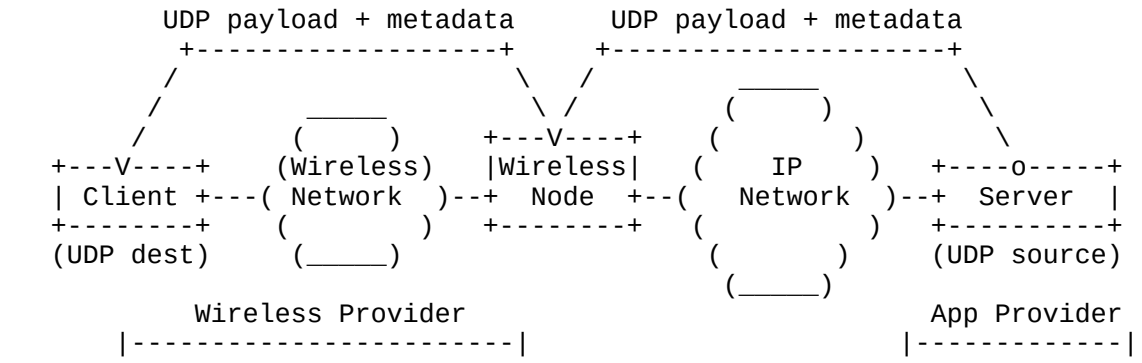

Figure 2: Media Payload and Metadata in UDP Packet

The authentication described here focuses specifically on the metadata carried in the UDP Option Header, rather than the entire packet payload. This differs from Section 11.9 (Authentication) of [UDP-OPTION-HDR], which applies authentication to the full payload. Since the user payload is already encrypted, authenticating it again is unnecessary; protecting the appended metadata alone is sufficient to ensure integrity. The method proposed in this document is intentionally lightweight, targeting only the appended metadata to reduce processing overhead while still ensuring its integrity.

4. Header Authentication Methods Analysis

Several methods can be used to authenticate encapsulation headers without processing the entire packet payload. The choice depends on balancing security strength, processing overhead, and deployment complexity.

- HMAC-based authentication provides strong integrity protection and is widely supported. It uses shared keys between endpoints and remains effective even when outer IP addresses or encapsulation headers change in transit.

- Authentication Header (AH) [RFC 4302] authenticates IP headers but fails when any address or mutable field is modified en route, such as by NAT, load balancers, or re-encapsulation. This limitation makes AH unsuitable for multi-segment or cloud-backbone environments. HMAC, by protecting only non-mutable steering fields and

tolerating address changes, provides a more flexible and robust solution.

- Lightweight checksums (e.g., CRC or Fletcher) offer minimal overhead but do not protect against intentional tampering and are only suitable for environments with low security risk.

- Digital signatures ensure strong, non-repudiable integrity but add significant computational cost and packet size, making them impractical for high-throughput scenarios.

The methods described here are intended for authenticating only the encapsulation header (e.g., GENEVE, SRH, UDP Option) while leaving the already-encrypted payload untouched. This targeted authentication reduces processing overhead compared to full-packet authentication, while still preventing tampering with critical steering or metadata information.

5. Encoding of Header Authentication Value

5.1. Analysis of HMAC Value

While a 32-byte HMAC value is recommended for strong security [NIST-SP-800-107], appending this to every packet can increase the packet size enough to cause MTU exceedance, fragmentation, and higher transmission overhead.

A practical compromise is to use a shorter HMAC, such as 4 bytes (32 bits) or 8 bytes (64 bits), when the primary goal is integrity and authenticity verification without heavy performance impact. Truncating the HMAC conserves bandwidth, reduces processing time, and is especially beneficial in high-speed or resource-constrained environments.

As noted in [RFC2104], authentication tags may be truncated (e.g., to 128 bits or less) to balance security with efficiency. While shorter tags reduce the cryptographic strength compared to full-length values, they can still provide adequate protection against common threats in the intended use cases.

5.2. Consideration in Generating the Authentication Value

HMAC-SHA-256 [RFC4868] [RFC6234] produces a 32-byte (256-bit) output by default. For scenarios requiring shorter authentication values, such as 4 bytes (32 bits) or 8 bytes (64 bits), the following methods can be used:

- Direct Truncation: Select the first n bytes from the HMAC output. This is simple, efficient, and the approach recommended by [RFC2104].
- Secondary Hash: Apply a separate hash function to the full 32-byte HMAC output to derive the desired shorter value.

Direct truncation is generally preferred for its simplicity and minimal processing overhead, especially in high-speed or resource-constrained environments.

5.3. Authentication Value Encoding

The HMAC-Auth-VAL Sub-TLV carries the HMAC authentication value used to validate the encapsulation header. It MUST be the last Sub-TLV in the encapsulation header. The HMAC is computed over the entire encapsulation header excluding the HMAC-Auth-VAL Sub-TLV itself, using a pre-configured algorithm:

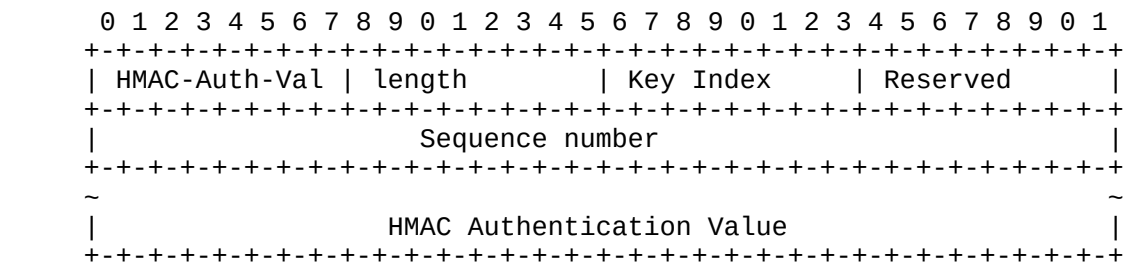

Figure 3 HMAC Sub-TLV

- HMAC-Auth-Val (8 bits): Type value = 6 (assigned by [MULTI-SEG-SDWAN]).
- Length (8 bits): Total length of the Value field, excluding the Type and Length fields. This is equal to the HMAC length plus 4 bytes for sequence number, 1 byte Key Index, and 1 reserved byte. Default is 10 bytes, but

longer values may be used in deployments with higher security requirements.
- Key Index (8 bits): Index of the key used for HMAC calculation, enabling key rotation and management. The key set is pre-shared and agreed upon between sender and receiver.
- Sequence number (32 bits): Unsigned, network-byte-order counter scoped per {sender, receiver, Key Index}. The sender MUST initialize it on key activation and increment by 1 for each packet; reuse within the same Key Index is forbidden. The receiver MUST enforce anti-replay with a sliding window and drop stale/duplicate values.
- HMAC Authentication Value: The computed HMAC output, based on the encapsulation header (excluding this Sub-TLV).

5.4. Selective Packet Header Authentication

Selective Packet Header Authentication (SPHA) enables receiving nodes to authenticate only a subset of flows, as determined by control-plane instructions from a controller or management system. Every packet includes an Authentication TLV in its encapsulation header, but only packets of selective flows carry a genuine authentication value; others contain dummy values. This design ensures that an observer cannot distinguish which packets are actually authenticated, reducing the risk of targeted tampering.

The metadata in each GENEVE or other encapsulation header is always valid and current. The control plane determines the authentication frequency, or which flows to be authenticated. On receipt, nodes verify only the packets of designated flows according to pre-configured policy, thereby reducing processing load while maintaining protection against header manipulation.

SPHA is particularly valuable in environments where authenticating every packet header is computationally expensive, such as on resource-constrained devices. It optimizes the balance between security and performance, ensuring that the integrity of critical flows is protected without imposing unnecessary overhead on all traffic.

6. Authentication Key Distribution

The lightweight authentication methods in this document apply to environments where IPsec tunnels connect SD-WAN CPEs to Cloud GWs. When traffic from a CPE is destined for services in the Cloud Data Center, the Cloud GW decrypts the IPsec traffic. When traffic is routed through the Cloud Backbone to reach remote CPEs, the lightweight authentication method is applied without decryption at the Cloud GWs.

6.1. Key Distribution Via Secure Control Plane

When a secure control channel exists-such as between two organizations for interconnection, or between a network controller and its CPEs-authentication keys can be exchanged over this channel.

Pre-shared keys are preferred for their simplicity and efficiency. For deployments requiring stronger security, keys should be generated using a cryptographically secure random number generator to avoid predictability, and key lifetimes should be kept short.

In a [MULTI-SEG-SDWAN] environment, the Cloud Controller can own the authentication keys and securely distribute them, such as via TLS, to the enterprise’s SD-WAN controller. In a [MEDIA-HDR-WIRELESS] environment, the application controller can similarly distribute keys securely to the wireless provider’s controller or management system.

To maintain ongoing security, keys should be rotated periodically, and versioning information should be included so both ends always use the correct and current keys.

6.2. Key Distribution Via Secure Data Plane Tunnel

In environments where IPsec tunnels connect SD-WAN CPEs to Cloud GWs, the IPsec tunnel itself provides a secure channel for transmitting authentication keys, protecting them from eavesdropping or tampering during distribution.

The existing IPsec session keys can also serve as input to a key derivation function (KDF), producing dedicated authentication keys that are cryptographically linked to the IPsec keys but never directly exposed. This approach ensures both strong security and operational efficiency by leveraging already-established secure channels.

7. Dynamic Authentication Policy Control

The selection and frequency of flows to be fully authenticated are determined by the network controller through a secure management channel with the edge nodes. This policy can target specific flows or, for example, only the first packet of those flows.

When a network segment is deemed at higher risk of security threats, such as man-in-the-middle (MITM) attacks, the controller can dynamically adjust the policy to:

- Increase the proportion of flows subject to full authentication.

- Apply additional header encryption between CPEs and Cloud GWs.

- Use stronger encryption algorithms (e.g., AES-256).

The secure management channel enables real-time adaptation to changing network conditions and threat levels, ensuring that authentication remains both efficient and effective. By adjusting authentication scope and strength as needed, the system can detect and deter malicious attempts to inject or manipulate traffic, maintaining the integrity of the data flow.

8. Packet Loss Handling

In environments using Selective Packet Header Authentication (SPHA), packet loss can reduce the number of authenticated packets available for integrity verification, which may temporarily weaken header-tampering detection. In non-SPHA deployments where every packet is authenticated, loss directly impacts both integrity verification and delivery reliability.

Therefore, mechanisms to mitigate the effects of packet loss are important in both SPHA and non-SPHA environments, but are especially critical in SPHA when authentication frequency is already reduced.

Mitigation Techniques

Several complementary methods can be applied to minimize the security and operational impact of lost packets:

- Retransmission Requests: Allow receivers to securely request retransmission of lost packets that were expected to carry valid authentication data.

- Forward Error Correction (FEC): Send additional error-correcting codes to enable reconstruction of lost packets without retransmission, which is useful in high-latency or unreliable networks.

- Sequence Numbering: Assign sequence numbers to all packets so that missing packets can be detected quickly and trigger alerts or retransmissions.

- Heartbeat Messages: Periodically send control or status packets summarizing authenticated packet sequences to speed loss detection.

- Multi-Path Transmission: Transmit duplicates of critical packets over multiple network paths to increase delivery success probability.

- Adaptive Authentication Thresholds: Dynamically increase authentication frequency when packet loss is detected, ensuring enough authenticated packets reach the receiver to maintain integrity guarantees.

- Time-based Reconciliation: Periodically compare packet headers received over a given interval to identify gaps and detect possible tampering.

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

Replay attacks-where an attacker resends previously captured packets to disrupt or mislead the network-must be addressed in both SPHA and non-SPHA environments, but the mitigation approach differs.

In non-SPHA environments, where every packet is authenticated, replay protection is typically implemented using per-packet identifiers such as sequence numbers, nonces, or timestamps. This makes detecting and rejecting

replays straightforward, provided the anti-replay window and state tracking are properly managed.

In SPHA environments, where authentication is applied to selected flows rather than every packet, replay protection must still ensure that packets within authenticated flows are uniquely identifiable and time-bound. Without this, attackers could replay unauthenticated packets from an existing flow or reuse authenticated packets within the replay window. Flow-level authentication therefore requires supplemental measures, such as per-packet sequence numbers or timestamps, to maintain protection against replays.

Common techniques for mitigating replay attacks include:

- Sequence Numbers: Ensure each packet has a monotonically increasing identifier to detect duplicates or out-of-order arrivals.

- Nonce Values: Add unpredictable values to each packet to guarantee freshness.

- Session Keys with Rotation: Change keys periodically so captured packets cannot be reused after a key update.

- Packet Expiry Information: Set validity windows so old packets are automatically rejected.

- Stateful Inspection: Maintain connection state to identify packets that fall outside expected patterns.

These measures, used individually or in combination, ensure that replay attempts are detected and blocked, preserving the integrity of both SPHA and non-SPHA deployments.

10. Security Considerations

The effectiveness of HMAC-based authentication depends on the strength of the shared key, the robustness of the hash algorithm, and sound key management practices. Deployments should select algorithms and key lifetimes that match their specific security requirements.

While a 32-bit signature can reduce processing and bandwidth overhead, especially valuable in resource-constrained environments, it provides lower cryptographic strength.

Operators must evaluate whether this trade-off meets their risk tolerance.

For environments concerned about possible HMAC key compromise, an additional integrity layer such as IPsec AH [RFC4301] or ESP-NULL [RFC2410][RFC6071] may be applied on top of existing IPsec encryption between CPEs. These methods require pairwise IPsec key management between Cloud GWs and CPEs and introduce additional processing load. AH also has the limitation of being incompatible with NAT traversal due to changes in outer IP headers.

11. Manageability Considerations

Effective management of HMAC key configurations between SD-WAN edges and Cloud GWs is critical for maintaining authentication consistency. The management system must support secure key generation, protected distribution, and periodic key rotation. It should also include clear procedures for rapid key revocation and replacement in the event of compromise or other security incidents, ensuring minimal disruption to operations.

12. IANA Considerations

This document makes no IANA requests. It reuses the HMAC-Auth-Val Sub-TLV Type defined in [MULTI.SEG.SDWAN].

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