6MAN
Internet-Draft
Intended status: Informational
Expires: 21 January 2026

J.F. Guan
BUPT
S. Yao
THU
K.X. Liu
X.L. Hu
J.L. Liu
BUPT

July 2025

Terminal Identity Authentication Based on Address Label
draft-guan-6man-ipv6-id-authentication-03

Abstract

This document proposes an IPv6-based address label terminal identity authentication architecture, which tightly binds identity information to the source address of data packets. This approach enables hop-by-hop identity authentication while ensuring source address verification. The mechanism facilitates user identity verification, ensuring privacy protection, security, and efficient auditing. Additionally, this document details the implementation of address label authentication within the IPv6 extension header.

Table of Contents

1. Introduction 2
2. Address Label Extension Header Format 3
2.1. Additional Validation Data 5
2.2. Integrity Checksum Code 5
3. Address Label Protocol Processing 6
3.1. Assumptions 6
3.2. Network Environment 7
3.3. Address Label Packet Sending 7
3.4. Address Label Packet Forwarding 8
3.5. Address Label Packet Reception 8
4. Security Considerations 8
4.1. Randomness Requirements 9
4.2. Anonymous Address 9
4.3. Unlinkability 9
4.4. Integrity Protection 10
5. Privacy Considerations 10
6. IANA Considerations 10
7. References 11
7.1. Normative References 11
7.2. Informative References 11
Authors' Addresses 12

1. Introduction

In the realm of network communication, the IP address, serving as a relatively stable identifier for the origin of requests, is particularly vulnerable to exploitation by malicious attackers[I-D.ip-address-privacy-considerations]. The ease with which IP addresses can be forged and impersonated complicates the task of ascertaining the legitimacy of data packets for all network participants. Consequently, the implementation of source address verification becomes imperative.

Furthermore, the existing network architecture suffers from a disconnection between the IP address and the terminal identity, rendering the process of tracing a terminal's identity from its IP address exceedingly cumbersome. This separation not only hinders

efficient identity verification but also leaves the network more susceptible to various security threats. Thus, there is a pressing need for a robust mechanism that can effectively bridge this gap, ensuring both the security and integrity of network communications.

Thus this document proposes an identity authentication mechanism based on address tags to protect user privacy and facilitate identity verification. The address label identifies the identity of different terminals. The address label serves as an identifier, created by initializing the multidimensional attribute table of the terminal and encrypting it using a symmetric key. The length of the address label depends on the length of the encryption algorithm. A section of the address label will be incorporated into the IPv6 address, serving as a communication identifier. The remaining part will accompany the data packet to the next hop. At the subsequent hop, the device will utilize this information to acquire the complete address label.

The identity authentication mechanism, relying on address labels, utilizes the user’s multi-dimensional attributes. It designs a unified user identity and employs a distributed consensus infrastructure for consensus and management. The user identity is anonymized and embedded in the data packet to ensure secure data transmission. The cross-domain receiving end verifies the authenticity and trustworthiness of the terminal identity through the dynamic label authentication mechanism.

Given the aforementioned reasons, we employ address label extension headers to transmit terminal identity for authentication and scrutiny within the network. We make full use of the IPv6 address space to establish a robust connection among terminal identity, address, and data. This allows address labels to withstand diverse attacks like tampering, replay, and forgery.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Address Label Extension Header Format

The ALE extension header is encapsulated in the Hop-by-Hop Options header. The (outer) protocol header (IPv6, or Extension) that immediately precedes the ALE header contain the value 0 in its Next Header field[RFC5871] (see IANA web page at http://www.iana.org/assignments/protocol-numbers). Figure 1 illustrates the basic format of the ALE header[RFC6564] [RFC7045].

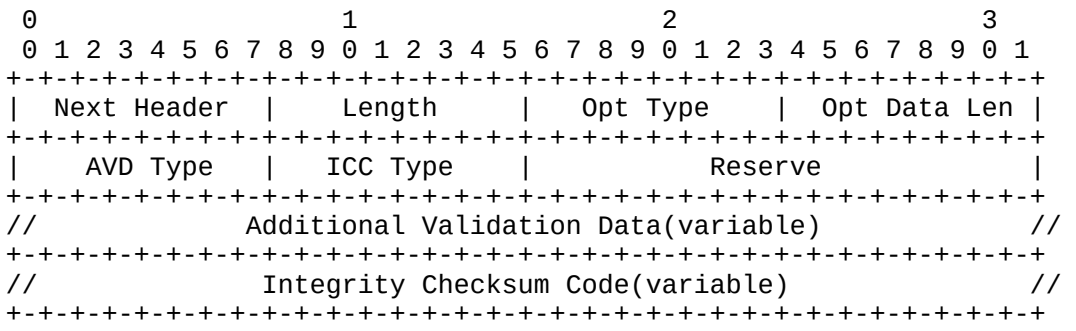

Figure 1: Basic Format of the LAE Header

* The Next Header field identifies the type of header immediately following the Hop-by-Hop Options header[RFC8200].

* The Length field indicates the length of the Hop-by-Hop Options header in 8-octet units, not including the first 8 octets.

* The Opt Type field identifies address label data with a value of 0x73[RFC8200].

* The Opt Data Len field indicates the length of the entire ALE header in 8 bytes, including the variable length Additional Validation Data and Integrity Checksum Code sections.

* The AVD Type field represent the method of encryption used in the AVD field and the length of it. The value of this field MUST NOT be 0.

* The ICC Type field represent the method of hash used in the ICC field and the length of it. The value of this field MUST NOT be 0.

* The Reserve field provides room for our future work. We aim to incorporate path verification functionality into our protocol if possible.

* The Additional Validation Data is a variable length field used to store partially encrypted data with a symmetric key (see Section 2.1).

* The Integrity Checksum Code is a variable length field used to store the hash results of partial terminal identity information and the entire transport layer data (see Section 2.2).

2.1. Additional Validation Data

The Additional Validation Data (AVD) represents a partial value derived from the encryption of identity information using a symmetric key.

The initial 64 bits of the encryption result will be inserted into the trailing 64 bits of the IPv6 packet source address (referred to as the Implicit Identifier IID). The remaining portion will be stored in the AVD field, ensuring data authenticity and the inviolability of identity information.

The length of this field MAY vary depending on the selected encryption algorithm. The data requiring encryption encompasses the anonymous identity, timestamp, and serial number of the terminal. The verifier must decrypt these outcomes to authenticate the identity.

The encryption algorithm type used is represented by the AVD Type field, and specific values refer to Table 1:

AVD Type field	Encryption Algorithm	AVD Length/bits
0	Reserve	
1	SM4	64
2	AES128	64
3	AES256	192
4	DES	64
5	3DES	64

Table 1: Category of Symmetric Key Encryption Algorithm

Values not listed in the table are considered reserved values.

2.2. Integrity Checksum Code

The Integrity Checksum Code field (ICC) is a hash result containing partial identity information of the terminal and the complete transport layer data.

The ICC field guarantees data integrity during transmission. The data subject to hash verification encompasses the IPv6 source address, destination address, AVD, anonymous terminal identity AID, timestamp, serial number, and the transport layer data (including transport layer headers) of the message. In the verification process, the identical hash operation is applied to these data, and subsequently, the ICC is compared. Transmission correctness is confirmed only when the two match; otherwise, this packet should be discarded.

Different hash algorithms MAY result in different lengths of ICC. We use the ICC Type field to represent the current hash algorithm being used, and specific values refer to Table 2:

ICC Type field	Encryption Algorithm	ICC Length/bits
0	Reserve	
1	SHA256	256
2	SHA384	384
3	SHA512	512
4	MD5	128

Table 2: Category of Hash Algorithm

Values not listed in the table are considered reserved values.

3. Address Label Protocol Processing

3.1. Assumptions

The process description for the terminal identity authentication based on address label will be based on the following assumptions:

(a) All entities can verify the routing prefix generated by AS.

(b) Each host and router in the protocol negotiates a symmetric key through a secure method for the subsequent protocol service. The source AS and destination AS also share the corresponding symmetric key in secret.

(c) The encryption method is assumed to be secure, i.e., encryption cannot be broken and MACs cannot be forged.

3.2. Network Environment

This document describes a protocol process that includes a Source AS and a Destination AS, both of which contain a Border Router for packet forwarding, and each of which contains a User/Host. The network environment is illustrated in the figure 2 below:

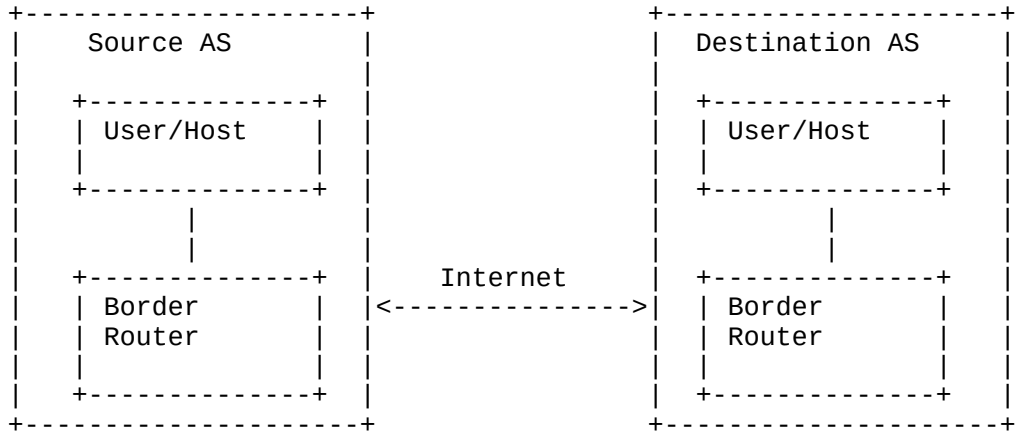

Figure 2: Process Description Topology

3.3. Address Label Packet Sending

Each packet sent from the source host MUST insert the extension header described above and encrypt the Anonymous Identifier (AID) of its own IPv6 address using a symmetric key. The first 64 bits of the encrypted result are used as the IID(implicit Identifier) of the address, while the remaining bits are used as the AVD in the extension header. By following these steps, the non-linkability between the sender and receiver is improved to counter third-party observers on the same LAN segment and maintain anonymity.

To ensure that there are no errors during data transmission, we also use ICC fields as a means of verifying data correctness. We use specific hash algorithms to calculate the checksum of partial terminal identity information and all transport layer data. In addition, the original checksum check of the transport layer is also retained. But due to the modification of the IPv6 address, the checksum of the transport layer MAY need to be recalculated.

3.4. Address Label Packet Forwarding

For outgoing packets, when the border router of the source AS receives an outgoing packet, it uses the corresponding symmetric key to decrypt the IID and AVD of the address, thereby obtaining the original AID. Next, the border router calculates the ICC using the IP header and payload and then compares it with the ICC in the packet to verify the integrity of the packet. If the verification fails, the border router discards the packet. Otherwise, it uses the corresponding symmetric key shared with the destination AS to re-encrypt the AID and forwards the packet.

For incoming packets, when the border router of the destination AS receives an incoming packet, it requests and reproduces the symmetric key of the terminal from the domain server based on the information in the packet extension header, and decrypts the address label to verify it. If the verification is successful, it encrypts the original source address's AID using the symmetric key and forwards the packet.

According to [I-D.ietf-6man-hbh-processing] and [PROC-HBH-OPT-HEADER], new options should be defined with the Action type set to 00 to skip over this option. However, due to our modification of the packet's source address, skipping the hop-by-hop option header can lead to more serious issues, such as ICMPv6's inability to notify the source address of certain network errors. Therefore, we decided to adopt a design that directly discards the data packet.

3.5. Address Label Packet Reception

The packets forwarded by the border router are received by the User/Host in the destination AS, which decrypts the AID of the source address with its own symmetric key, and verifies the real source address of the host with which it communicates. Similarly, the host needs to verify the correctness of data transmission through ICC field.

Through the above steps, the protocol in this document can guarantee that only verified packets can leave the source AS and successfully reach the destination host.

4. Security Considerations

This section contains security considerations for the protocol described in this document.

4.1. Randomness Requirements

All random values in the protocol and symmetric key MUST be generated using a cryptographically secure source of randomness [RFC4086].

4.2. Anonymous Address

Attackers can track and identify the sender’s activity patterns and history by using the source address in network traffic to conduct tracking attacks. In the network environment, the source address is usually a fixed or stable identifier, such as an IP address, a MAC address, or other types of identifiers. These identifiers can be collected and correlated by attackers to construct the sender’s activity patterns and history.

This document protocol ensures that the sender can hide their identity from the source AS, the transit ASes, the destination AS, and even the receiver, making it difficult for the source address information in network traffic to be exposed or identified. This is because the attacker does not know the symmetric key of the AS, so they cannot decrypt IDD and AVD to obtain user identifiers and extract user identities, thus fully protecting the sender’s privacy and security.

However, it is important to note that this document protocol does not maintain sender anonymity for observers in the LAN segment because they already know the identity (link layer address) of the sender.

4.3. Unlinkability

Unlinkability in this document refers to the ability of the sender’s different actions or activities to be uncorrelated. This way, the sender can prevent their actions or activity from being linked by adversaries or other third parties, thereby avoiding the leakage of their information or intentions.

Through this protocol, adversaries cannot obtain more information about the source correlation of traffic by observing any number of flows from the same Autonomous System (AS). The source correlation of traffic refers to the possibility of two flows coming from the same sender. The meaning of traffic here is the same for the sender and receiver as it is in traditional networks, but it is different for other devices and observers in the network. This is because the protocol in this document changes the source or destination address.

In the network environment, adversaries may invade hosts in the same LAN segment as the sender and obtain clues about the sender’s identity, which leads to a decrease in sender-receiver unlinkability.

When the sender sends a data packet, the invaded host in the LAN segment can know the source and destination addresses. However, in this document protocol, since the sender encrypts the AID of the destination address in each data packet, the invaded host cannot know the true destination address.

4.4. Integrity Protection

Integrity protection ensures that the information in the extended header has not been tampered with or modified during packet transmission.

In this document protocol, if an adversary sends packets with incorrect ICCs, the border router will concatenate and decrypt the IID and AVD in the packet and calculate a new ICC using SN, timestamps, and other data. If the new ICC calculated by the border router does not match the one in the packet header, the border router identifies the packet as bogus and discards it. Through the above analysis, data packets that do not pass the ICC integrity check in this protocol will not be forwarded, thus ensuring data integrity.

5. Privacy Considerations

According to the design outlined in this document, the IPv6 source addresses of each packet will change at each hop, making it difficult to trace the source based on the address label. This effectively protects the sender’s privacy information.

Although the MAC address is not within the scope of this document, it is crucial to note that the MAC address changes with each hop, making it difficult to trace back to the original sender of the data packet.

6. IANA Considerations

IANA is asked to assign the Option Type in the "Destination Options and Hop-by-Hop Options" subregistry of the "Internet Protocol Version 6 (IPv6) Parameters" registry as follows:

Hex Value	Binary Value			Description	Reference
	act	chg	rest		
0x73	01	1	10011	Address Label	This document

Table 3: Destination Options and Hop-by-Hop Options Registry

7. References

7.1. Normative References

[I-D.ietf-6man-hbh-processing]
Hinden, R. M. and G. Fairhurst, "IPv6 Hop-by-Hop Options Processing Procedures", Work in Progress, Internet-Draft, draft-ietf-6man-hbh-processing-20, 5 June 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-6man-hbh-processing-20>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC5871] Arkko, J. and S. Bradner, "IANA Allocation Guidelines for the IPv6 Routing Header", RFC 5871, DOI 10.17487/RFC5871, May 2010, <https://www.rfc-editor.org/info/rfc5871>.

[RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <https://www.rfc-editor.org/info/rfc6564>.

[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/info/rfc8200>.

7.2. Informative References

[I-D.ip-address-privacy-considerations]
Finkel, M., Lassey, B., Iannone, L., and B. Chen, "IP Address Privacy Considerations", Work in Progress, Internet-Draft, draft-ip-address-privacy-considerations-03, 10 January 2022, <https://datatracker.ietf.org/doc/html/draft-ip-address-privacy-considerations-03>.

[PROC-HBH-OPT-HEADER]
Peng, S., Li, Z., Xie, C., Qin, Z., and G. Mishra, "Operational Issues with Processing of the Hop-by-Hop

Options Header", Work in Progress, Internet-Draft, draft-ietf-v6ops-hbh-10, 17 February 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-v6ops-hbh-10>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <https://www.rfc-editor.org/info/rfc7045>.

Authors' Addresses

Jianfeng Guan
BUPT
No.10 Xitucheng Road, Haidian District
Beijing
100876
China
Email: jfguan@bupt.edu.cn

Su Yao
THU
No.30 Shuangqing Road, Haidian District
Beijing
100084
China
Email: yaosu@tsinghua.edu.cn

Kexian Liu
BUPT
No.10 Xitucheng Road, Haidian District
Beijing
100876
China
Email: kxliu@bupt.edu.cn

Xiaolong Hu
BUPT
No.10 Xitucheng Road, Haidian District

Beijing
100876
China
Email: hxl814446051@bupt.edu.cn

Jianli Liu
BUPT
No.10 Xitucheng Road, Haidian District
Beijing
100876
China
Email: kuohao233@bupt.edu.cn