Network Working Group
Internet-Draft
Intended status: Informational
Expires: 18 October 2026

T. Ahearn
Independent

16 April 2026

Signed Email Authentication Layer (SEAL)
draft-ahearn-seal-01

Abstract

This document defines the Signed Email Authentication Layer (SEAL), a cryptographically signed identity envelope carried within a new message header field, SEAL-Envelope. SEAL provides a stable, forwarding-resilient identity assertion that binds the origin domain to a specific message instance using the SEAL-MSGID header, which contains a SEAL-protected copy of the [RFC5322] Message-ID present at message creation time. After SEAL-MSGID is set, intermediaries may modify or discard the visible [RFC5322] Message-ID header without affecting SEAL validity. SEAL also records the canonical [RFC5322] From header value in the envelope, enabling detection of From rewriting without affecting SEAL validity. SEAL is designed to complement current approaches such as DKIM, DMARC, and ARC by reducing their dependency on mutable message components and by providing a canonical, tamper-evident identity layer that can remain valid across many common transformations.

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 18 October 2026.

Copyright Notice

Table of Contents

1. Introduction

Email authentication today relies primarily on SPF, DKIM, and DMARC. These mechanisms provide meaningful protections but share a fundamental architectural limitation: they bind identity to message components that are inherently mutable. SPF validates only the connecting IP address, not the author identity. DKIM signs selected headers and the message body, both of which are routinely modified by forwarders, mailing lists, and security appliances. DMARC depends on alignment with SPF and DKIM and therefore inherits their fragility.

The Signed Email Authentication Layer (SEAL) introduces a new identity layer for email that reduces dependence on mutable message components. SEAL defines a canonical, tamper-evident identity envelope that is signed by the originating domain and carried in a dedicated header, SEAL-Envelope. The envelope asserts the sender's identity, the intended recipient scope, a validity window, and a stable message identifier derived from the [RFC5322] Message-ID and carried in the SEAL-MSGID header.

SEAL depends on the SEAL-MSGID header and the scope field for message binding. The SEAL-MSGID header is populated by the sender with the [RFC5322] Message-ID value at the time the message is created. After SEAL-MSGID is set, intermediaries may modify or discard the [RFC5322] Message-ID header without affecting SEAL validity. All other [RFC5322] headers and the message body may also be modified in transit without affecting SEAL signature verification. Because SEAL does not sign or depend on mutable headers or the message body, it can remain valid across many forms of forwarding and transformations that commonly break DKIM, subject to the scope and msgid constraints recorded in the envelope.

SEAL is intended to function alongside DKIM and ARC, each addressing a different part of the authentication problem. DKIM and ARC provide content-binding and chain-of-custody information, while SEAL moves the identity assertion into a separate, tamper-evident envelope that remains stable across many forms of forwarding and header rewriting. Together, these mechanisms offer complementary assurances without overlapping responsibilities.

This document is written using the xml2rfc v3 vocabulary [RFC7991].

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

This section defines terminology used throughout this document.

Origin domain -- The domain that generates and signs the SEAL envelope and publishes the corresponding public key in DNS.

SEAL envelope -- The canonical JSON object defined by this

document that carries the identity assertion and associated metadata, and that is signed by the origin domain.

Receiver -- A system that receives an email message and is capable of parsing and verifying the SEAL-Envelope header.

Forwarder -- An intermediary that receives a message and re-sends it, potentially modifying the [RFC5322] header block or message body.

Recipients -- The recipient identifiers the sender addresses the message to at send-time (e.g., mailbox, domain, mailing list address). This term does not include the expanded set of delivery mailboxes produced by intermediaries such as mailing lists, aliases, or forwarding systems.

Scope -- A field in the SEAL envelope that indicates the recipients for whom the SEAL assertion is valid.

Message identifier (msgid) -- The value of the SEAL-MSGID header, which is set by the sender to the [RFC5322] Message-ID header value at message creation time and used to bind the SEAL envelope to a specific message instance.

Author identity (from) -- The canonical [RFC5322] From header value as recorded in the SEAL envelope. This value is signed and immutable; intermediaries may rewrite the visible From header without affecting SEAL verification.

4. Problem Statement

DKIM [RFC6376] attempts to provide message integrity by signing selected headers and the message body. However, email is inherently mutable. Forwarders, mailing lists, and security appliances routinely modify headers and bodies in ways that invalidate DKIM signatures. ARC attempts to preserve authentication results across intermediaries, but it adds complexity and does not address the underlying issue that identity is tied to mutable data.

The core architectural problem is that DKIM binds identity to message components that are not stable. SEAL addresses this by binding identity to a separate, canonicalized envelope that is not part of the mutable header block and is therefore more resilient to common transformations.

5. SEAL Overview

SEAL defines a new identity envelope containing:

* *origin* -- the domain asserting identity
* *scope* -- the intended recipients
* *iat* -- issued-at timestamp
* *exp* -- expiration timestamp
* *msgid* -- the value of the SEAL-MSGID header, which is set to the [RFC5322] Message-ID at message creation time
* *from* -- the canonical [RFC5322] From header value at message creation time
* *alg* -- signature algorithm identifier
* *eh* -- hash of the canonical envelope excluding the "sig" field
* *sig* -- signature over the canonical envelope

The envelope is serialized using a strict canonical JSON serialization. Before signing, the implementation computes the "eh" value as a hash of the canonical envelope with all required fields except "sig". The signature is then computed over this same canonical form, including "eh" but excluding "sig". The resulting signed envelope is base64-encoded and placed in the SEAL-Envelope header field.

SEAL depends on the SEAL-MSGID header and the scope field for message binding. The SEAL-MSGID header is populated by the sender with the [RFC5322] Message-ID value at the time the message is created. All other [RFC5322] headers, including the [RFC5322] Message-ID header itself, and the message body may be modified in transit without affecting SEAL signature verification. The envelope provides a stable, tamper-evident identity assertion that can survive many forms of forwarding and common message transformations, subject to the scope and msgid constraints recorded in the envelope.

6. The SEAL-Envelope Header

The SEAL-Envelope header carries the signed identity envelope. Its ABNF is:

```
SEAL-Envelope = "SEAL-Envelope:" OWS 1*(VCHAR / WSP)
```

The header value is a base64-encoded representation of the signed envelope. Intermediaries MUST NOT modify the header value. Any modification will invalidate the signature.

7. Envelope Canonicalization

The SEAL envelope is serialized using a strict canonical JSON serialization to ensure that all implementations produce an identical byte sequence for signing and verification. The canonicalization rules are:

* *Top-level structure* -- The envelope MUST be a single JSON object. Arrays MUST NOT appear at the top level.

* *Key ordering* -- All keys MUST be serialized in lexicographic order based on their UTF-8 byte values.

* *String escaping* -- Strings MUST use the minimal JSON escaping required by [RFC8259]. Characters MUST NOT be escaped unless required. Unicode escapes, if used, MUST use lowercase hexadecimal.

* *Timestamps* -- The iat and exp fields MUST use full RFC 3339 / ISO 8601 timestamps in UTC with seconds included (for example, "2026-03-03T23:17:00Z").

* *Whitespace* -- No whitespace is permitted outside of string literals.

* *Field presence* -- All defined fields MUST appear; optional omission is not permitted.

8. Signature Model

SEAL uses modern digital signature algorithms such as Ed25519 [RFC8032] or RSA-PSS. To compute the signature, the implementation MUST construct the complete envelope object with all required fields except "sig", serialize it according to the canonicalization rules in this document, and compute the "eh" field as a hash of this canonical form.

The signature is then computed over the same canonicalized envelope (including "eh" but excluding "sig"). The "sig" field is populated with the base64-encoded signature value. The public key is published in DNS using records defined in this document. Receivers verify the signature using the public key retrieved from DNS. If verification fails, the SEAL assertion is invalid.

The msgid field in the SEAL envelope MUST contain the exact value of the SEAL-MSGID header. The SEAL-MSGID header MUST be set by the sender at message creation time to the [RFC5322] Message-ID header value of the message. This binds the envelope to the specific message instance and prevents transplanting the envelope onto a different message. After SEAL-MSGID is set, intermediaries MAY modify or discard the [RFC5322] Message-ID header without affecting SEAL verification.

The scope field indicates the recipients for whom the SEAL assertion is valid. A receiver MUST treat a SEAL envelope as invalid if the message is delivered to a recipient that does not match the value in scope. Receivers MAY apply additional local policy based on scope, but a scope mismatch constitutes a SEAL verification failure.

9. Forwarding Semantics

SEAL is designed to remain valid across many forms of forwarding that do not change the effective recipients or the message identity as recorded in the envelope. Because the envelope depends on the SEAL-MSGID header and the scope field for message binding, intermediaries may modify other headers, including the [RFC5322] Message-ID header, and the message body without affecting SEAL signature verification, provided that the message is delivered only to recipients that are consistent with the scope value.

Receivers MAY compare visible header values in the received message to the corresponding values recorded in the SEAL envelope to detect header rewriting, but such comparisons do not affect SEAL signature validity.

Receivers MUST treat the value of the msgid field inside the SEAL envelope as the authoritative message identifier for SEAL verification. Differences between the SEAL-MSGID header and the visible [RFC5322] Message-ID header MUST NOT cause SEAL verification to fail.

Receivers MAY compare the from field in the SEAL envelope to the visible [RFC5322] From header to detect From rewriting. A mismatch MUST NOT cause SEAL verification to fail, but MAY be used as input to local policy or user interface signaling.

Forwarders MAY add their own SEAL-related headers in future extensions, but this document defines only the origin signature carried in the SEAL-Envelope header.

9.1. Mailing List Behavior

Mailing lists are not forwarders; they generate new messages with new recipients, new Message-IDs, and often modified content. Because SEAL binds identity to a specific message instance via the msgid and scope fields, the original SEAL envelope cannot survive mailing list redistribution. Mailing lists that wish to provide SEAL assurances may generate a new SEAL envelope for the redistributed message, asserting their own domain identity. This behavior is consistent with existing email authentication mechanisms and reflects the architectural reality that mailing lists create new messages rather than forwarding existing ones.

10. Deployment Considerations

SEAL is designed for incremental and non-disruptive deployment. Domains may adopt SEAL without requiring any changes from receivers, intermediaries, or other senders. SEAL operates in parallel with existing authentication mechanisms such as SPF, DKIM [RFC6376], and DMARC, and does not interfere with their operation.

A domain may begin signing messages with SEAL at any time by publishing one or more SEAL keys in DNS and adding the SEAL-Envelope header to outbound mail. Receivers that understand SEAL can verify the signature and use the resulting identity assertion as an additional signal. Receivers that do not implement SEAL will simply ignore the SEAL-Envelope header without impact.

SEAL does not require modifications to MTAs, message transfer pipelines, or intermediary systems, and does not require any changes to SMTP protocol behavior. Because the SEAL envelope is independent of mutable [RFC5322] headers and the message body, SEAL signatures can remain valid across many forms of forwarding and common message transformations that frequently invalidate DKIM signatures. However, SEAL verification depends on the scope and msgid values recorded in the envelope, and SEAL signatures might not survive transformations (such as some mailing list expansions) that change the effective recipients or message identity. In scenarios where an intermediary such as a mailing list redistributes a message, that intermediary might generate a new SEAL envelope for the redistributed message, but the behavior of such intermediaries is out of scope for this document and is left for future work. Such intermediaries may also continue to rely on DKIM to provide content-binding or message-integrity assurances for the redistributed message.

Domains MUST publish at least two SEAL keys if they intend to rotate keys without interrupting SEAL verification. One key is used for current signing and at least one additional key remains available to

validate previously signed messages. Additional keys MAY be published to support delegation to third-party sending services or to facilitate algorithm agility. This document does not define an upper limit on the number of SEAL keys a domain may publish. Receivers MUST attempt verification with any SEAL key published at the _seal.<domain> label.

SEAL does not define receiver policy. Receivers determine how SEAL verification results are incorporated into local authentication, reputation, or filtering decisions. Future work may explore integration of SEAL with DMARC or related mechanisms, but such policy considerations are out of scope for this document.

A dedicated SEAL DNS resource record type may be defined in a future revision of this specification. For initial deployment, SEAL keys are published using TXT records to ensure broad compatibility with existing DNS infrastructure.

11. DNS Requirements

Origin domains publish SEAL public keys in DNS under the _seal label using TXT records. TXT is the initial publication mechanism for SEAL keys. A dedicated SEAL-specific DNS resource record type MAY be defined in a future revision of this specification, but this document does not define such a type.

A SEAL key record is published at:

_seal.<domain>. TXT ("v=SEAL1; k=ed25519; p=<base64-public-key>")

Domains publish SEAL keys as TXT records at the _seal.<domain> label.

Operators SHOULD choose DNS TTL values that balance responsiveness to key rotation with caching efficiency. TTLs on the order of one hour to one day are RECOMMENDED for SEAL key records. Receivers SHOULD respect published TTLs but MAY cache keys for shorter periods according to local policy.

12. Security Considerations

SEAL provides a cryptographically signed identity assertion that is independent of mutable message components. Receivers MUST NOT assume that the message body or headers other than SEAL-Envelope and SEAL-MSGID are unmodified simply because the SEAL signature verifies. SEAL is complementary to existing mechanisms such as DKIM [RFC6376]. Receivers and operators SHOULD handle SEAL verification results according to local policy.

13. Privacy Considerations

SEAL asserts domain-level identity only and does not introduce new personal data fields. SEAL does not expose user identifiers beyond those already present in the email message.

SEAL does not introduce new privacy considerations beyond those already present in DKIM [RFC6376] or other domain-based authentication mechanisms.

14. IANA Considerations

This document has no IANA actions.

15. Future Work

Future work may define:

* SEAL extensions for intermediaries.
* Integration with DMARC.
* A dedicated DNS RR type.
* Post-quantum signature algorithms.

16. Normative References

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

[RFC5322] Resnick, P., "Internet Message Format", October 2008, <https://www.rfc-editor.org/rfc/rfc5322>.

[RFC6376] Crocker, D. and T. Hansen, "DomainKeys Identified Mail (DKIM) Signatures", September 2011, <https://www.rfc-editor.org/rfc/rfc6376>.

[RFC8032] Josefsson, S., "Edwards-Curve Digital Signature Algorithm (EdDSA)", January 2017, <https://www.rfc-editor.org/rfc/rfc8032>.

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

[RFC7991] Hoffman, P., "The "xml2rfc" Version 3 Vocabulary", December 2016, <https://www.rfc-editor.org/rfc/rfc7991>.

[RFC8259] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

Author’s Address

Tim Ahearn
Independent
Email: tim.ahearn@outlook.com