Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 2 October 2026

T. Ahearn
Independent

31 March 2026

Signed Email Authentication Layer (SEAL)
draft-ahearn-seal-00

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 is independent of the mutable RFC 5322 header block and message body. SEAL is designed to complement or replace DKIM and ARC by eliminating their dependency on mutable message components and by providing a canonical, tamper-evident identity layer that survives transit through intermediaries.

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 2 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 avoids this dependency on mutable data. SEAL defines a canonical, immutable identity envelope that is signed by the originating domain and carried in a dedicated header field, SEAL-Envelope. The envelope asserts the sender's identity, the intended recipient scope, a validity window, and a stable message identifier. Because SEAL does not sign or depend on the RFC 5322 header block or message body, it remains valid across forwarding and transformations that commonly break DKIM.

SEAL is designed to operate alongside, or in some deployments instead of, DKIM and ARC. By relocating identity into a separate, tamper-evident envelope, SEAL provides a more stable foundation for domain-level authentication.

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 RFC 5322 [RFC5322] header block or message body.

* *Scope* -- A field in the SEAL envelope that indicates the intended recipient domain or domains for the message.

* *Message identifier (msgid)* -- A stable, opaque identifier for the message within the SEAL envelope, intended to remain unchanged across forwarding.

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 resilient to common transformations.

5. SEAL Overview

SEAL defines a new identity envelope containing:

* *origin* -- the domain asserting identity

* *scope* -- the intended recipient domain or domains

* *iat* -- issued-at timestamp

* *exp* -- expiration timestamp

* *msgid* -- stable message identifier

* *alg* -- signature algorithm identifier

* *sig* -- signature over the canonical envelope

The envelope is serialized in a strict canonical JSON serialization and signed using the private key of the origin domain. The resulting signed envelope is base64-encoded and placed in the SEAL-Envelope header field. Because the envelope is independent of the mutable RFC 5322 [RFC5322] header block and message body, it remains valid across forwarding and other transformations that commonly disrupt DKIM.

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 RFC 8259. 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. The signature is computed over the canonicalized envelope excluding the "sig" field. 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 sign the resulting byte sequence.

The "sig" field is then 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 is an opaque, stable identifier for the message within the SEAL envelope. It MUST remain unchanged across forwarding. The _msgid_ MAY be derived from, but is not required to match, the RFC 5322 [RFC5322] Message-ID header field. Implementations SHOULD generate _msgid_ values that are globally unique, such as UUIDs.

The _scope_ field indicates the intended recipient domain or domains for the message. A receiver MAY choose to enforce scope by treating a message as out-of-scope if the recipient domain does not match the value in _scope_. This document does not require any particular enforcement behavior; scope is an assertion by the origin domain that receivers MAY use as an input to local policy.

9. Forwarding Semantics

SEAL is designed to survive forwarding. Because the envelope is independent of the mutable header block and message body, intermediaries may modify those components without affecting SEAL validity.

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.

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 two 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. SEAL does not require any changes to SMTP protocol behavior. Because the SEAL envelope is independent of the RFC 5322 [RFC5322] header block and message body, SEAL signatures remain valid across forwarding, mailing list expansion, and common message transformations that frequently invalidate DKIM signatures.

Domains typically publish one SEAL key, but MAY publish a second key to support key rotation or delegation to a third-party sending service. 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 MAY publish up to two SEAL TXT records at the _seal.<domain> label. Receivers MUST retrieve all SEAL TXT records published at _seal.<domain> and MUST attempt verification with any key whose parameters match the envelope.

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 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>.

Author’s Address

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