Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 15 August 2026

T. Herr
GreenArrow Email

11 February 2026

DKIM2 Best Practices
draft-herr-dkim2-bcp-00

Abstract

[DKIM2] and associated documents describe the DomainKeys Identified Mail v2 (DKIM2) email authentication protocol. DKIM2 is designed to address shortcomings in email authentication protocols and mechanisms released prior to DKIM2, specifically SPF [RFC7208], DKIM [RFC6376], DMARC [RFC7489], and ARC [RFC8617]. Those shortcomings most commonly manifested themselves in email messages that passed through one or more intermediary hosts (e.g., forwarders or mailing list servers) while transiting from origination to final destination. Although these messages were properly authenticated when sent, the alteration of their path and/or content by the intermediary hosts would cause authentication checks to fail at the final destination. In addition, DKIM was susceptible to "replay" of signatures, when attackers would construct abusive messages in such a way that they would pass validation checks for a DKIM signature that had been imported from another message entirely.

To address these shortcomings, DKIM2 provides methods not only for intermediary systems to provide details of the changes that they make to a message in transit, but also for receiving hosts to validate those changes in addition to the original message. DKIM2 also allows recipients to detect when messages have been unexpectedly "replayed" and can also ensure that delivery status notifications (DSNs) are only sent to entities that were involved in the transmission of a message.

This document describes the recommended usage of the DKIM2 protocol for sending hosts, intermediary hosts, and receiving hosts.

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/.

Copyright Notice

Table of Contents

1. Introduction

DomainKeys Identified Mail v2 (DKIM2) permits a person, role, or organization to document that they have handled an email message by associating a domain name [RFC1034] with the message [RFC5322]. A public key signature is used to record that they have been able to read the contents of the message and write to it.

Verification of claims is achieved by fetching a public key stored in the DNS under the relevant domain and then checking the signature.

Message transit from author to recipient is through Forwarders that typically make no substantive change to the message content and thus preserve the DKIM2 signature. Where they do make a change the changes they have made are documented so that these can be "undone" and the original signature validated.

When a message is forwarded from one system to another an additional DKIM2 signature is added on each occasion. This chain of custody assists validators in distinguishing between messages that were intended to be sent to a particular email address and those that are being "replayed" to that address.

The chain of custody can also be used to ensure that delivery status notifications are only sent to entities that were involved in the transmission of a message.

Organizations that process a message can add to their signature a request for feedback as to any opinion (for example, that the email was considered to be spam) that the eventual recipient of the message wishes to share.

This document discusses best practices for signing, handling, and validating messages that have a DKIM2 signature. These best practices are based in large part on the many years of experience the email community has with the authentication protocols that DKIM2 is intended to replace.

2. Terminology and Definitions

This section defines terms used in the rest of the document.

DKIM2 is designed to operate within the Internet Mail service, as defined in [RFC5598]. Basic email terminology is taken from that specification.

DKIM2 inherits many ideas from DKIM ([RFC6376]) which, for clarity we refer to in this specification as DKIM1. In addition, some features were influenced by experience from (see [CONCLUDEARC]) the experimental ARC protocol ([RFC8617]).

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 [RFC2119]. These words take their normative meanings only when they are presented in ALL UPPERCASE.

2.1. Signers

Elements in the mail system that sign messages on behalf of a domain are referred to as Signers. These may be MUAs (Mail User Agents), MSAs (Mail Submission Agents), MTAs (Mail Transfer Agents), or other agents such as mailing list "exploders". In general, any Signer will be involved in the injection of a message into the message system in some way. The key point is that a message must be signed before it leaves the administrative domain of the Signer.

2.2. Forwarder

[RFC5598] defines a Relay as transmitting or retransmitting a message but states that it will not modify the envelope information or the message content semantics. It also defines a Gateway as a hybrid of User and Relay that connects heterogeneous mail services. In this document we use the concept of a Forwarder which is an MTA that receives a message and then, as an alternative to delivering it into

a destination mailbox, can forward it on to another system in an automated, pre-determined, manner.

2.3. Reviser

As will be seen, a Forwarder may alter the message content or header fields, in such a way that existing signatures on the message will no longer validate. If so, then a record will be made of these changes. We call a Forwarder that makes such changes a Reviser.

2.4. Verifiers

Elements in the mail system that verify signatures are referred to as Verifiers. These may be Forwarders, Revisers, MTAs, Mail Delivery Agents (MDAs), or MUAs. It is an expectation of DKIM2 that a recipient of a message will wish to verify some or all signatures before determining whether or not to accept the message or pass it on to another entity.

2.5. Signing Domain

A domain name associated with a signature. This domain may be associated with the author of an email, their organization, a company hired to deliver the email, a mailing list operator, or some other entity that handles email. What they have in common is that at some point they had access to the entire contents of the email and were in a position to add their signature to the email.

2.6. Selectors

To support multiple concurrent public keys per signing domain, the key namespace is subdivided using "selectors".

The number of public keys and corresponding selectors for each domain is determined by the domain owner. Many domain owners will use just one selector, whereas administratively distributed organizations can choose to manage disparate selectors and key pairs in different regions or on different email servers.

Selectors can also be used to delegate a signing authority, which can be withdraw at any time. Selectors also make it possible to seamlessly replace keys on a routine basis by signing with a new selector, while keeping the key associated with the old selector available.

2.7. Senders

[RFC5598] defines a path for messages that starts with hops from an Author to an Originator to a Relay, before transiting additional hops on the way to their final destination. In this document, we will collectively refer to those first three hops (Author-->Originator-->Relay) as Sender.

3. Signing and Verification Cryptographic Algorithms

DKIM2 supports multiple hashing and digital signature algorithms. One hash function (SHA256) if specified here and two signing algorithms are defined by the DKIM2 protocol: RSA-SHA256 and Ed25519-SHA256. Signers and Verifiers *MUST* implement SHA256. Signers *SHOULD* implement both RSA-SHA256 and Ed25519-SHA256. Verifiers *MUST* implement both RSA-SHA256 and Ed25519-SHA256.

3.1. The SHA256 Hashing Algorithm

The SHA256 hashing algorithm is used to compute body and header hashes as defined in [DKIM2] and [DKIM2] The resultant values are stored within Message-Instance header fields.

3.2. The RSA-SHA256 Signing Algorithm

The RSA-SHA256 Signing Algorithm computes a hash over all the Message-Instance and DKIM2-Signature header fields as described in [DKIM2] using SHA-256 (FIPS-180-4-2015) as the hash-alg. That hash is then signed by the Signer using the RSA algorithm (defined in PKCS#1 version 1.5 [RFC8017]) as the crypt-alg and the Signer’s private key. The hash *MUST NOT* be truncated or converted into any form other than the native binary form before being signed. The signing algorithm *MUST* use a public exponent of 65537.

Signers *MUST* use RSA keys of at least 1024 bits. Verifiers *MUST* be able to validate signatures with keys ranging from 1024 bits to 2048 bits, and they *MAY* be able to validate signatures with larger keys.

3.3. The Ed25519-SHA256 Signing Algorithm

The Ed25519-SHA256 Signing Algorithm computes a hash over all the Message-Instance and DKIM2-Signature fields as described in [DKIM2] using SHA-256 (FIPS-180-4-2015) as the hash-alg. It signs the hash with the PureEdDSA variant Ed25519, as defined in Section 5.1 of [RFC8032].

3.4. Other Algorithms

Other algorithms *MAY* be defined in the future. Verifiers *MUST* ignore any hashes or signatures using algorithms that they do not implement.

3.5. Key Management

Some level of assurance is required that a public key is associated with the claimed Signer. DKIM2 does this by fetching the key from the DNS for the domain specified in the d= field.

DKIM2 keys are stored in a subdomain named "_domainkey". Given a DKIM2-Signature field with a "d=" tag of "example.com" and an "s1=" tag of "foo.bar", the DNS query will be for "foo.bar._domainkey.example.com".

NOTE: these keys are no different, and are stored in the same locations as those for DKIM1 ([RFC6376]).

Further details can be found in [DKIMKEYS].

4. Sender Actions

In order to participate in DKIM2, Senders _*MUST_ be Signers. The following sections describe best practices for such Senders.

4.1. Sign all outgoing messages with DKIM2 and DKIM1 signatures.

Because it is expected that the transition from DKIM1 to DKIM2 across the email ecosystem will be gradual, Senders *SHOULD* sign messages with both DKIM1 and DKIM2 keys until such time as the deployment of DKIM2 is effectively ubiquitous.

4.2. Sign with multiple keys

To ensure maximum interoperability, Senders *SHOULD* sign messages with multiple DKIM2 signatures, with each such signature using a different cryptographic algorithm.

4.3. Only Sign On Exit

Many Senders originate messages from infrastructure that requires the message transit multiple hops before reaching its egress point to travel to its destination. Senders *MUST* only apply DKIM2 signatures to messsages only at this last hop before it leaves their infrastructure.

4.4. Regularly Rotate DKIM Signing Keys

Senders *SHOULD* follow best practices for rotating DKIM keys, both for DKIM1 and DKIM2 - https://www.m3aawg.org/sites/default/files/m3aawg-dkim-key-rotation-bp-2019-03.pdf

5. Forwarder Actions

A Forwarder participating in DKIM2 *MUST* be a Signer. Further, as mentioned above, a Forwarder might also function as a Reviser and/or a Verifier before passing a message along to the next hop. This section describes the best practices for a Forwarder participating in DKIM2.

5.1. Attempt Verification of Existing DKIM Signatures

A Forwarder *MUST* attempt to Verify the DKIM signatures found in a message when it first arrives to the Forwarder.

5.2. Sign all outgoing messages with DKIM2 and DKIM1 signatures.

Forwarders participating in DKIM2 *MUST* be Signers, In addition, for the reason mentioned above in the Sender Actions section of this document, Forwarders *SHOULD* sign messages with both DKIM1 and DKIM2 keys until such time as the deployment of DKIM2 is effectively ubiquitous.

5.3. Sign Even If Just Forwarding and Not Revising

Forwarders participating in DKIM2 *MUST* DKIM2 sign any message they handle, regardless of whether or not they Revise the message. Such signatures maintain a proper DKIM2 "chain of custody" and allow for cleaner verification and unwinding of changes at future hops.

5.4. Sign with multiple keys

To ensure maximum interoperability, Forwarders *SHOULD* sign messages with multiple DKIM2 signatures, with each such signature using a different cryptographic algorithm.

5.5. Only Sign On Exit

As with Senders, if the Fowarder’s infrastructure requires the message to transit multiple hops before reaching its egress point to travel to its destination, then the Forwarder *MUST* only apply DKIM2 signatures to messsages only at this last hop before it leaves their infrastructure.

5.6. Regularly Rotate DKIM Signing Keys

Forwarders *SHOULD* follow best practices for rotating DKIM keys, both for DKIM1 and DKIM2 - https://www.m3aawg.org/sites/default/files/m3aawg-dkim-key-rotation-bp-2019-03.pdf

5.7. Continue doing From munging

When a message reaches a Forwarder, if the Forwarder determines that the Domain Owner of the RFC5322.From header domain has published a DMARC record for that domain, the Fowarder *SHOULD* alter the RFC5322.From header in such a way as to ensure that the message will not fail DMARC validation when it reaches its destination. Some strategies for doing this are discussed in Section 7.4 of [DMARCbis]

5.8. Bron’s idea - Privacy preseving forwarder - remove mention of forward target from bounces

Need some text for this...

6. Receiving Host Actions

Receiving Hosts in the DKIM2 ecosystem are those that host mailboxes for the domain to which the message is ultimately routed. Receiving Hosts are Verifiers, and depending on local policy, the disposition of the message may be influenced by whether or not the message passes DKIM2 verification checks.

Verification of messages will rely in part on whether or not a full DKIM2 "chain of custody" exists for the message, meaning whether or not the Verifier can reliably determine if each hop that handled the message prior to its reaching the Receiving Host applied a proper DKIM2 signature to the message. We will define three conditions, as follows:

* A message that was DKIM2 signed by each host handling the message prior to its reaching the Receiving Host is one that "Never Left The DKIM2 Ecosystem"

* A message that was DKIM2 signed by some, but not all, hosts handling the message prior to its reaching the Receiving Host is one that was "In and Out of The DKIM2 Ecosystem"

* A message that contains no DKIM2 signatures when reaching the Receiving Host is one that "Never Entered The DKIM2 Ecosystem"

Each condition will come with its own set of recommendations for the Receiving Host.

6.1. Messages That Never Left The DKIM2 Ecosystem

If such a message passes DKIM2 Verification performed by the Receiving Host, the Receiving Host should apply local policy for determining message handling and disposition.

If such a message fails DKIM2 Verification performed by the Receiving Host, the Receiving Host *MAY* safely reject the message under assumption that the DSN(s) will only be sent to entities responsible for transmission of the message. The Receiving Host's local policy, however, will dictate final handling and disposition.

6.2. Messages That Were In and Out of the DKIM2 Ecosystem

Examples of messages that match this condition would be:

* The message was DKIM2 signed by the Sender, and passed through one or more Forwarders on its way to the Receiver, and at least one Forwarder did not DKIM2 sign the message.

* The message was not DKIM2 signed by the Sender, and passed through one or more Forwarders on its way to the Receiver, and at least one Forwarder did DKIM sign the message.

Need text on how to detect that message left DKIM2 Ecosystem?

For such messages, local policy will dictate handling. Receivers will have to decide whether or not a successful DKIM2 Verification is a condition of message acceptance or whether or not a failed Verification might not preclude acceptance but might influence message disposition. As DKIM2 deployment widens and protocol usage matures, Receivers might alter their local policies to be more reliant on DKIM2 Verification.

6.3. Messages Never Entered The DKIM2 Ecosystem

Receivers participating in DKIM2 will have to establish local policy to dictate what to do with messages that arrive bearing no DKIM2 signatures. This policy may change over time, as Receivers observe what percentage of mail arrives each day bearing DKIM2 signatures vice the percentage that arrives without, and how their mailbox holders engage with each kind.

6.4. DKIM1 and DKIM2 Interoperability

(From Wei)

When DKIM2 is first deployed, there will be a mix of DKIM-only (RFC6376) and DKIM2-capable participants and there will be significant interaction between these two groups. To maximize compatibility with the DKIM receivers, many DKIM2-capable participants will sign and verify DKIM in addition to DKIM2. At origination, DKIM2-capable sender should sign with DKIM and DKIM2 as some receivers may only support DKIM, and others both DKIM and DKIM2. DKIM2-capable receivers should verify both DKIM and DKIM2 signatures when found, but that said, should not treat results equivalently. When forwarding, DKIM2-capable sender should perform DKIM2 signing only when the DKIM2 security properties to protect against replay and backscatter are met. This property is met when any prior DKIM2 signatures are be valid per draft-clayton-dkim2-spec.

DKIM does not provide the same anti-replay and backscatter protections that DKIM2 can. Consequently DKIM2-capable participant should not DKIM2 sign a forwarded message with the same consideration if there is only DKIM signatures available. Under certain circumstances a DKIM2-capable forwarder may choose to DKIM2 sign a message with a prior DKIM signature if it can be certain that the message was not replayed or introduce backscatter using local-policy. Some properties to consider are looking for the same anti-replay and backscatter properties described in draft-clayton-dkim2-spec except signed with the DKIM signature. Typically this only is only apparent for DKIM signed messages at origination, meaning when the payload From header is DMARC aligned with the DKIM signed domain and the signature is valid. Forwarders may wish to consider other validations under local-policy as well.

6.5. Notes

Pull stuff from RFC 6376

Keys are same, but some features in keys in DKIM1 will be deprecated; unknown fields will be ignored, as with DKIM1

DKIM2 and DMARC

Discuss implications of requesting feedback

7. References

7.1. Normative References

[DKIM2] Clayton, R., Chuang, W., and B. Gondwana, "DomainKeys Identified Mail Signatures v2 (DKIM2)", Work in Progress, Internet-Draft, draft-clayton-dkim2-spec-06, 20 January 2026, <https://datatracker.ietf.org/doc/html/draft-clayton-dkim2-spec-06>.

[DKIMKEYS] Chuang, W., "Domain Name Specification for DKIM2", Work in Progress, Internet-Draft, draft-chuang-dkim2-dns-03, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-chuang-dkim2-dns-03>.

[DMARCbis] Herr, T. and J. R. Levine, "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", Work in Progress, Internet-Draft, draft-ietf-dmarc-dmarcbis-41, 4 April 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-dmarc-dmarcbis-41>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/rfc/rfc1034>.

[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/rfc/rfc2119>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <https://www.rfc-editor.org/rfc/rfc5322>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <https://www.rfc-editor.org/rfc/rfc6376>.

[RFC7208] Kitterman, S., "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1", RFC 7208, DOI 10.17487/RFC7208, April 2014, <https://www.rfc-editor.org/rfc/rfc7208>.

[RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015, <https://www.rfc-editor.org/rfc/rfc7489>.

7.2. Informative References

[CONCLUDEARC]
Adams, J. T. and J. R. Levine, "Concluding the ARC Experiment", Work in Progress, Internet-Draft, draft-adams-arc-experiment-conclusion-01, 4 December 2025, <https://datatracker.ietf.org/doc/html/draft-adams-arc-experiment-conclusion-01>.

[RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, DOI 10.17487/RFC5598, July 2009, <https://www.rfc-editor.org/rfc/rfc5598>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/rfc/rfc8017>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/rfc/rfc8032>.

[RFC8617] Andersen, K., Long, B., Ed., Blank, S., Ed., and M. Kucherawy, Ed., "The Authenticated Received Chain (ARC) Protocol", RFC 8617, DOI 10.17487/RFC8617, July 2019, <https://www.rfc-editor.org/rfc/rfc8617>.

Appendix A. Changes from Earlier Versions

[[This section to be removed by RFC Editor]]

Author’s Address

Todd Herr
GreenArrow Email
Email: todd@someguyinva.com