

Network Working Group
Internet-Draft
Obsoletes: 4686 (if approved)
Intended status: Standards Track
Expires: 23 April 2026

B. Gondwana
Fastmail
R. Clayton
Yahoo
W. Chuang
Google
20 October 2025

DKIM2 Header Definitions
draft-gondwana-dkim2-header-04

Abstract

This document describes the email header fields defined for DKIM2, and how they work together to provide the required properties.

This is an early draft, a work in progress.

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 23 April 2026.

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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Imported ABNF	3
1.2. Defined ABNF	3
1.3. Definitions	3
2. The DKIM2 Header	4
2.1. Value of i=	5
2.2. Value of t=	5
2.3. Value of f=	6
2.4. Value of bh=	7
2.5. Value of b=	7
2.6. Value of n=	7
2.7. Value of rt=	7
2.8. Value of mf=	7
2.9. Value of d=	8
2.10. Value of pp=	8
2.11. Value of mv=	8
2.12. Value of h=	8
2.12.1. Implicitly signed headers	9
2.13. Values of s= and a=	9
3. Process for validating a DKIM2 message on receipt	9
4. Temporary Notes	10
4.1. Dealing with modifications	10
4.2. Dealing with replays	11
5. Feedback loops	11
6. Handling of messages that leave the DKIM2 ecosystem	12
6.1. DKIM2-foo headers	13
6.1.1. DKIM2-Authentication-Results	13
7. Security	13
7.1. Multiple addresses in the rt= field	13
8. IANA Considerations	13
9. References	13
9.1. Normative References	13
9.2. Informative References	14
Appendix A. Changes from Earlier Versions	14
A.1. draft-gondwana-dkim2-headers-04:	14
A.2. draft-gondwana-dkim2-headers-03:	15
A.3. draft-gondwana-dkim2-headers-02:	15

A.4. draft-gondwana-dkim2-headers-01:	15
A.5. draft-gondwana-dkim2-headers-00:	15
Authors' Addresses	15

1. Introduction

To achieve the goals laid out in [MOTIVATION], this document describes the content of a 'DKIM2-Signature' header field, which can be added to [IMF] messages. The 'DKIM2-Signature' header field contains signatures which can be verified using public keys from the DNS, in a similar way to how [DKIM] works.

The 'DKIM2-Signature' header field also borrows design elements from [ARC], however it places strict requirements on the alignment of the components of [DKIM2] header fields in the [IMF] message with the [SMTP] reverse-path and forward-path.

1.1. Imported ABNF

This document imports the following ABNF:

- * mailbox and domain from [SMTP] Section 4.1.2.
- * selector from [DKIM] Section 3.1.
- * tag-list from [DKIM] Section 3.2.
- * date-time from [DATETIME] Section 5.6.

1.2. Defined ABNF

We will copy (re-define) at least:

- * instance and position from [ARC] Section 3.9.

1.3. Definitions

- * "DKIM2 Header". Any header field with the name 'DKIM2-Signature'.
- * "Historical DKIM2 Header". Any DKIM2 Header where there exists another DKIM2 Header with a higher position on the message.
- * "Active DKIM2 Header". Any DKIM2 Header where there is no DKIM2 Header with a higher position on the message.
- * "Initial Timestamp". The timestamp on the DKIM2 Header in position 1.

2. The DKIM2 Header

The DKIM2 Header draws significant amounts of design from [ARC].

[DKIM2] Headers are a structured tag-list as defined in [DKIM] Section 3.2.

Field identifier	Type	Explanation
i=	position	Sequence Number (from 1 to N)
t=	date-time	Timestamp ([DATETIME])
f=	dkim2-flags	Indicates if feedback is requested, or that changes are not allowed
d=	domain	Signing domain
a=	TBD	Crypto algorithm(s) used (unless combined with b= to allow for multiple signatures on the same email, see discussion of crypto-agility above)
s=	selector	DKIM
b=	base64	Signature over hash value strings (DKIM uses b=)
bh=	base64	Body hash value (see discussion)
h=	header-list	Extra headers signed by this hop
mf=	mailbox	[SMTP] reverse-path MUST be exactly this value
rt=	mailbox-list	[SMTP] forward-path MUST only contain members of this list
pp=	domain	Per procuracionem (as proxy for) - if present, the domain being signed for
mv=	position	[DELTA] version number signed by this header

n=	TBD	a nonce value (could use for database lookup for DSN handling)
----	-----	--

Table 1

Note that we have not included a version number. Experience from [IMF] onwards shows that it is essentially impossible to change version numbers. If it becomes necessary to change DKIM2 in the sort of incompatible way that a new version number would be needed, we recommend using new header field names instead.

2.1. Value of i=

The maximum allowed position is 50.

If the Active DKIM2 Header is not in position 1, there MUST be exactly one Historical DKIM2 header for each lower integer number, starting at 1.

To allow [SMTP] transactions with more than one forward-path, there MAY be more than one Active DKIM2 header on a message.

If a message is recieved with multiple Active DKIM2 headers, the next signer MUST remove all but one of them, keeping the one with the forward-path for which it is creating an onward message. For example if one of the recipients of a multi-recipient message has a forwarding rule, then the DKIM2 header field for that recipient will be the one that is retained as the Historical DKIM2 Header for the previous position on that particular copy of the message.

2.2. Value of t=

The value is a [DATETIME] date-time.

For a message in transit, the timestamp SHOULD be less than one week ago. For bounces, they SHOULD be returned to their source within 2 weeks of the timestamp on the DKIM2 Header with position 1.

This requires that as the destination, or as any intermediate hop unable to deliver the message further, you SHOULD create a bounce within one week of the initial timestamp.

Also, as a recipient, you SHOULD reject any message with an initial timestamp more than a week in the past.

This allows signing hosts to rotate keys and only have to keep the old keys (and keep the private key private) for a maximum of 2 weeks.

2.3. Value of f=

The value is a comma-separated list of flags. The following flags are defined:

Flag	Position	Description
donotmodify	any	This signer requests that this message not be further modified
donotexplode	any	This signer requests that this message not be further exploded to multiple recipients
donotforward	any	This signer requests that this message not be sent to any different recipient
exploded	any	This signed wishes to state that it created multiple different copies of this same message
feedback	any	This signer requests that this message be included in any feedback loop reports

Table 2

The "donot" fields are advisory. They might be appropriate for some types of transactional email. Since it is only a request, intermediaries may, by local policy, not honor it, but they SHOULD NOT relay mail where the request has not been honored to third parties.

The "exploded" tag has security and privacy implications, which will be fleshed out in the security and privacy considerations if this stays in. If sending a BCC, or having a hidden alias target, setting exploded could leak that fact. If you don't set exploded and have taken a hidden copy, they could both wind up at a later system which might then reject the copies.

The "feedback" field is advisory, however its absence means that the sender does not want feedback on this message. This document does not describe a mechanism for determining how to send feedback, or what format that feedback should be in.

2.4. Value of bh=

The header hash is always calculated with the "relaxed" algorithm defined in [DKIM] section 3.4.2.

2.5. Value of b=

The body hash is always calculated with the "relaxed" algorithm defined in [DKIM] section 3.4.4.

2.6. Value of n=

The nonce value is available for any purpose, but may well be used as an index into a database to access meta-data about an email that has been handled in the past. DKIM2 signatures expire after a fixed period (a week would be appropriate) so that it is not necessary to hold information for indefinite periods or to handle DSNs for email that was delivered long ago.

2.7. Value of rt=

If a message is sent over [SMTP], then to be accepted as a valid DKIM2 message, every forward-path MUST exactly match the rf= value of an Active DKIM2 Header.

See Security Considerations in this document for a discussion of avoiding inadvertant information disclosure in cases where multiple Active DKIM2 headers are present.

2.8. Value of mf=

If there are multiple Active DKIM2 Headers, they must all have the same mf= value.

If a message is sent over [SMTP], then to be accepted as a valid DKIM2 message, the reverse-path MUST exactly match the mf= value of the Active DKIM2 Headers.

The domain part of the mf= value MUST exactly match the d= value on all DKIM2 Headers.

2.9. Value of d=

For DKIM2 Headers with position greater than 1, the value of d= MUST be aligned with the domain of the rt= value of the immediately previous DKIM2 Header, for example the d= value for the DKIM2 header with i=3 must be the same as the domain part of the rt= value for the DKIM2 header with i=2.

2.10. Value of pp=

If the d= domain is not one of the domains of the previous hop, then pp= contains the domain which is being signed on behalf of.

The domain in pp= MUST be the exact domain of one of the addresses in rt= of the previous instance.

The relationship between the pp= and d= values can be one of:

1. an exact match.
2. d= is a parent domain (not crossing the PSL prefix list) of pp=
3. d= is a parent domain of an MX record for pp= (again, not crossing the PSL)
4. d= is explicitly listed in the DNS or by some other out of band method, in a record to be defined in a later revision of this document, or some future specification.

2.11. Value of mv=

This value is an integer for the matching [DELTA] version that this signature was created for. The inclusion of a version implicitly includes all MailVersion header fields with a lower or equal version, and excludes all MailVersion header fields with a greater version, when calculating the h= value.

2.12. Value of h=

See the definition in [DKIM2]

2.12.1. Implicitly signed headers

All DKIM2 Headers with a position less than or equal to the position of the DKIM2 Header itself are implicitly included in the signed headers for that DKIM2 Header. So in a message the Active DKIM2 Header at position 3, all the DKIM2- prefixed header are included in the signature. The Historical signature at position 2 includes the prefixed headers for positions 1 and 2 only, excluding those with position 3 - and of course the Historical signature with position 1 only includes those prefixed headers that are also at position 1 and excludes the others.

The MailVersion headers for lower or equal versions to the mv= field are also implicitly signed, and those with a higher version are not signed.

2.13. Values of s= and a=

TBD; we want to support multiple signatures with different algorithms in the same DKIM2 Header, so we need to figure out how to represent that to allow crypto agility.

3. Process for validating a DKIM2 message on receipt

[BOUNCE] describes that bounce messages are only allowed for validated messages.

To be able to safely create bounces, a DKIM2 aware MTA will run the following checks before responding to the DATA step of an [SMTP] transaction.

- 1) find the Active DKIM2 Header in the message header. If none are present, accept the message if local policy allows.
- 2) validate that the timestamp on the Active DKIM2 Header is less than one week old.
- 3) validate that the mf= on the Active DKIM2 Header matches the SMTP reverse-path, and that the d= matches the domain.
- 4) validate that every mailbox in the SMTP forward-path matches an item in the rt= on the Active DKIM2 Header.
- 5) validate that the local system can sign for each address in the SMTP forward-path.
- 6) fetch the public key for each given selector and algorithm that the receiver supports (or reply with a 5xx if no algorithms are supported). Reply with a 4xx error if the public key is unable to be fetched due to a temporary error.
- 7) validate that the signature is valid on the Active DKIM2 Header

This is sufficient information to be able to validate the bounce address and that the message was intended for the named recipients, so it can now be accepted subject to other local policy. At this stage if you generate a bounce, it will go back to the signer and the signer will accept it from you.

A receiver MAY choose not to perform the above tests during the SMTP transaction, or MAY choose to accept a message despite it failing those tests, however it MUST perform the tests before creating a [BOUNCE] DSN, and MUST NOT send a [BOUNCE] DSN if it is unable to create a valid signature.

4. Temporary Notes

The below is text from an earlier version of this document which I think is valuable to preserve at this point, however it probably belongs in another document and has not been updated to match the above.

4.1. Dealing with modifications.

Find the highest numbered DKIM2 header that reports a modification. Undo the modification and repeat. When all modifications have been done then there should be a match with the original signature (at hop1). If not then the email has been altered (in an undocumented manner) on its way to you and it SHOULD be rejected.

Note that it is not necessary to check the signature on a DKIM2 header that reports a modification. Undoing the modification and discovering that the message can now be authenticated is sufficient.

Over time a reputation can be developed for a intermediary which is making modifications and given sufficient trust then the "undo" step could be skipped. Note that the signature of the DKIM2 header that reports the modification would need to be checked to ensure reputation accrued to the correct entity.

If the modification is substantial (eg URLs rewritten, MIME parts removed) and it cannot be undone then the receiver (who may not be the immediate next hop) MUST trust the system doing the modification. If it does not then the mail SHOULD be rejected.

It will be noted that some modifications can totally change the meaning of an email. This specification does not try to limit modifications. We believe that being able to attribute modifications to a particular entity will allow reliable blocking of malicious intermediaries once they are identified.

4.2. Dealing with replays

Checking source and destination as recorded by the previous hop makes many "DKIM replay" scenarios impossible.

It is possible to exclude all replays by determining if any DKIM2 header reports an expansion event (one incoming email resulting in multiple further emails). If not then you would expect that the (original) hash of the email is unique and duplicates can be rejected.

If a expansion event is recorded then receiving multiple copies would not be a surprise. It will be necessary to use local policy to assess whether the number of copies received is acceptable or not.

Over time you may wish to develop a reputation for a DKIM2 identity which is doing expansions and conclude that a specific number of copies is to be expected. This can be used to refine local policy.

5. Feedback loops

Some mailbox providers are prepared to report their, or their customers', opinions about incoming email -- for example: that a customer marked a particular email as "spam". These systems are generally called "feedback loops".

There are usually bureaucratic systems to ensure reports are only sent to entities that wish to receive them and the mailbox provider may decide that some entities should not be sent any feedback.

The senders of email, the originator and/or a commercial company (an ESP) hired to send the email generally favor feedback loops because it allows them to make their emails more acceptable, and the commercial companies can rapidly become aware of customers whose email is widely disliked.

In DKIM2 any intermediary can request feedback, but it is still the decision of the mailbox provider as to whether any feedback will be sent. They may still require pre-registration on a per domain basis to receive feedback if only to ensure that any nominated email address is appropriate and is not an unsuspecting third party.

Note that feedback can be sent to any requesting entity. There is nothing special about a requester being at hop1 or hop2 on a chain. In particular some forwarding systems late in the chain may wish to become aware if they are forwarding emails that are then reported to be spam.

6. Handling of messages that leave the DKIM2 ecosystem

Note that DKIM2 signed email can also be DKIM1 signed, and so systems that are not DKIM2 aware can and will operate as they do at present.

DKIM2 capable servers will announce the capability in their initial banner in the usual manner for SMTP extensions.

When a DKIM2 signed email is delivered to a server that does not understand DKIM2 and leaves the DKIM2 ecosystem the DKIM2 specific events can no longer be expected to occur. In particular any failures to be deliver will be reported to the address in the relevant return path and not back along the DKIM2 chain.

A DKIM2 signed email may be delivered to a server that understands DKIM2 but if that server needs to forward the email elsewhere it may find that there is no signing key available for the relevant domain (recalling that the incoming email recorded the destination domain and it is necessary for the next "hop" to match with that. In such a case, once more the email will leave the DKIM2 ecosystem.

Refusing to allow an email to leave the DKIM2 ecosystem may be an appropriate choice in some circumstances. If so then an appropriate DSN should be created and passed back along the chain in the normal manner.

It is more likely that local policy will be to pass the email to the next intermediary even though this means that it leaves the DKIM2 ecosystem. In such a case it would be possible to add a final DKIM2 header to record this event, but doing this adds considerable complexity, and would provide limited information which was not otherwise available, hence no such header will be added.

If, after having left the DKIM2 ecosystem, the email reaches a DKIM2 aware system then the email may have been altered in such a way that the DKIM2 signatures now fail. The receiving system will apply its local policy to determine whether or not to accept the email.

If the DKIM2 signatures on the mail are valid, except that the last header does not specify the receiving system as the next hop, then once again it will a matter for local policy as to whether to accept the email. It might be thought that it was obvious that the email was acceptable, but the non-DKIM2-aware intermediaries that have handled it may have duplicated the email and there will be no DKIM2 headers to record this.

In any event, systems that accept email which has been outside the DKIM2 ecosystem MUST NOT add any further DKIM2 headers.

6.1. DKIM2-foo headers

DKIM2 allows for extension headers which are always added to the signature, but ONLY where they have an `i=` with a value equal to or lower than the matching DKIM2 header. This allows for extensions to add something at each DKIM2 hop; with it automatically added to the signed header set.

6.1.1. DKIM2-Authentication-Results

If present, is identical to how ARC-Authentication-Results from ARC are used, a place for any hop to add their calculated Authentication-Results header in a way that is signed; allowing other hops to add a similar header without needing to use modification algebra to remove it when reversing the calculation.

7. Security

Mostly TBD

7.1. Multiple addresses in the `rt=` field

If a message has multiple values in the `rt=` field, it is imperative that the creating system ensure that it doesn't leak BCC information to other recipients. This MUST be done by the sending system, by creating a separately signed message for each BCC recipient.

8. IANA Considerations

TBD

We will register the header name DKIM2 and the header prefix DKIM2- with reference to this document for syntax and constraints on the syntax of future headers with that prefix.

9. References

9.1. Normative References

- [BOUNCE] Robinson, A., "DKIM2 Procedures for bounce processing", Work in Progress, Internet-Draft, draft-robinson-dkim2-bounce-processing-01, 7 July 2025, <<https://datatracker.ietf.org/doc/html/draft-robinson-dkim2-bounce-processing-01>>.
- [DATETIME] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/rfc/rfc3339>>.

- [DELTA] Gondwana, B., "A method for describing changes made to an email", Work in Progress, Internet-Draft, draft-gondwana-dkim2-modification-alegebra-04, 17 October 2025, <<https://datatracker.ietf.org/doc/html/draft-gondwana-dkim2-modification-alegebra-04>>.
- [DKIM] 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>>.
- [DKIM2] Clayton, R., Chuang, W., and B. Gondwana, "DomainKeys Identified Mail Signatures v2 (DKIM2)", Work in Progress, Internet-Draft, draft-clayton-dkim2-spec-01, 14 October 2025, <<https://datatracker.ietf.org/doc/html/draft-clayton-dkim2-spec-01>>.
- [IMF] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/rfc/rfc5322>>.
- [MOTIVATION] Gondwana, B., Clayton, R., and W. Chuang, "DKIM2 - signing the source and destination of every email", Work in Progress, Internet-Draft, draft-gondwana-dkim2-motivation-03, 25 June 2025, <<https://datatracker.ietf.org/doc/html/draft-gondwana-dkim2-motivation-03>>.
- [SMTP] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <<https://www.rfc-editor.org/rfc/rfc5321>>.

9.2. Informative References

- [ARC] 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]]

A.1. draft-gondwana-dkim2-headers-04:

- * Replace nd= tag with pp= tag and update docs

A.2. draft-gondwana-dkim2-headers-03:

- * Add nd= tag and update the instructions to use it
- * Add mv= tag and align with modification-algebra-04
- * Remove the modifiedbody and modifiedheader flags, they are duplicative of the mv= data; which is more useful.
- * Update the rt= tag to be a list of addresses and require that all forward-path mailboxes be members of that list.

A.3. draft-gondwana-dkim2-headers-02:

- * cross-reference Richard's spec draft and rename to DKIM2-Signature
- * NOTE: we need to figure out if the two drafts make sense as separate documents or whether we should just merge them

A.4. draft-gondwana-dkim2-headers-01:

- * major rewrite
- * included support for multiple Active DKIM2 headers, as I have had side discussions that indicate that large corporate systems would have a lot of difficulty without this, and it removes constraints at the SMTP layer that were previously present.
- * cross referenced other drafts

A.5. draft-gondwana-dkim2-headers-00:

- * initial version
- * content extracted from draft-gondwana-dkim-motifivation

Authors' Addresses

Bron Gondwana
Fastmail
Email: brong@fastmailteam.com

Richard Clayton
Yahoo
Email: rclayton@yahooinc.com

Wei Chuang
Google
Email: weihaw@google.com