Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 7 May 2026

B. Gondwana
Fastmail
R. Clayton
Yahoo
W. Chuang
Google

3 November 2025

DKIM2 - signing the source and destination of every email
draft-ietf-dkim-dkim2-motivation-02

Abstract

This memo provides a rationale for building a new email accountability mechanism, based on the lessons learned from implementing the ARC experiment from RFC 8617 and other experiences from email system operators.

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 7 May 2026.

Copyright Notice

Table of Contents

1. Background and motivations

In 2007, [DKIM] (Domain Key Identified Mail / DKIM) was published, outlining a mechanism for a domain to sign email in a way that recipients could ensure that the email had come from an entity possessing the secret key matching a public key published in the DNS by the source domain.

[DKIM] has been updated and extended many times since then, and a large amount of operational experience has been gained using it.

There are a number of things beyond authenticating the original email that would be useful for mail system operators, particularly when it travels through multiple hops. There have been other attempts to solve some of these problems, e.g. [ARC] (Authenticated Received Chain / ARC), however they have not achieved the same level of widespread use as DKIM.

In particular, the following issues frustrate email system system operators:

1. You can legitimately receive a validly DKIM signed email, where there is no evidence inside the signed part that you were an intended recipient

2. An email can have a bounce (SMTP-FROM) email address for a domain which was never involved in the transit of that message

3. An email can be altered by forwarders or mailing lists, and there’s no way to know what parts of the message were changed

In the first two cases, a solution would be to have the sending system provide an unforgeable digital signature describing its identity and intent, such that if all parties involved in transiting versions of an email participate, there is no way for a third party to pretend to have been legitimately involved in processing the email, or to change or redirect the email in any way.

In the third case, a solution would be to provide a way to describe how to undo the changes, such that each domain’s signature can be attributed to the version of the message that was seen by systems which can sign on behalf of that domain.

2. Some properties for a system which would solve these issues

2.1. Explicit signing of all legitimate recipients for each message

By ensuring that the complete list of legitimate recipients for a message is encoded in the signed content of the message, it will become possible for receiving systems to confirm that they are an intended next hop for a message, and reject messages which the signer did not intend for them to receive.

Even if a message is BCC'd, a copy of that message sent to the BCC recipient can have that recipient address mentioned, without sending the same exact copy to the other recipients.

This mechanism does not survive naive forwarders, where the new destination address will not be explicitly mentioned, however a recipient system can track which addresses forward to it, and accept just those.

Over time as more software is updated to add signatures, the need to use heuristics becomes smaller, and eventually it will become possible to reject any messages where the [SMTP] RCPT TO forward-path addresses are not all present in the highest signature number header.

2.2. A chain of aligned signatures over multiple SMTP transactions

By having the initial signature be from the domain aligned to the From or Sender header, and each following hop adding its own signature with the domain of the recipient of the previous hop, it is possible to create a chain of custody where each recipient has confirmed that it should have received the message, and then signed the content with a key for its own domain.

If the recipient wishes to forward the message on to another address, it must apply its own DKIM2 header, signed by a key which is aligned to the domain of the recipient address in the previous DKIM2 header, and with a bounce address which is in the same domain.

The end result is, like ARC, a chain of domains which have handled the message; however unlike ARC, this chain MUST be fully linked in both directions, with every sending address aligning to the recipient address of the previous DKIM2 header.

2.3. A signed bounce format, sent in reverse along the same path

By having the mail-from address be signed and aligned to the signing domain, and having the bounce format include the signature headers of the message being bounced, it is required to have directly received the message to generate a bounce for it. This requirement eliminates the ability to cause backscatter entirely, as bounces can only go to a domain that sent the message, and only be sent from a domain which explicitly received that message.

The ability to avoid backscatter will allow receiving systems to delay their decisions about whether to accept a message, since they can make the decision without holding the connection open. This removes the need for mitigations like greylisting and even reduces the need for junk mail folders in jurisdictions where it is forbidden to discard messages once they are accepted.

Since the DSN messages always go back up the DKIM2 chain, any hop can strip off the higher number (i=) records; including the sender and recipient addresses for them, and create a bounce as if the forwarder itself was doing the rejection.

This would not be possible with SMTP-transaction-time rejection, as you can’t reliably hold open the connection from the previous hop while you talk to the next hop.

As asynchronous bounces will be common in DKIM2, this case becomes indistinguishable to the sender, allowing privacy-preserving forwarders to seamlessly operate.

Passing bounces back along the outgoing path also allows mailing lists to take responsibility for the event and not bother the person who sent a message to the list.

Provided that an email is correctly signed when received, it can be rejected at a later point in time. The DSN will be sent to the immediately preceding intermediary. Since the bounce travels back along the (fully authenticated) incoming path it cannot be sent to an uninvolved third party.

2.4. A way to describe changes

ARC describes a separate "Seal" header which which never gets modified, however this still allows an intermediate to make massive changes to a message and claim that it was still the original message. If a message goes through more than one set of modifications, it becomes impossible for the receiver to know what changes were made by each intermediary.

By defining an algebra sufficient to describe how to undo common changes, we can allow the receiver to compare the eventual message received with the original message sent, and decide which parties involved in changing the message are making the kind of changes that the recipient doesn't want.

Mailing lists (or alumni forwarders etc.) that alter the Subject header field (or other [IMF] headers) will record the previous header field contents. This is easy to undo for checking purposes.

Mailing lists that add text (either to a simple email body or one or more MIME parts within the body) will record details of the text they have added. This text can then be removed when checking earlier signatures.

2.4.1. Security gateways

There are some types of alteration, for example by security gateways, that may be impractical to describe in a cost-effective manner.

We would expect that outgoing gateways that may be adding disclaimers or rewriting internal identifiers would be provided with appropriate signing keys so that they could be the "first hop" as far as the rest of the email handling chain is concerned.

Incoming security gateways may be making substantial changes. Typically they will remove problematic types of attachment and rewrite URLs to use "interstitials".

Since this type of functionality is generally provided on a contracted basis, further intermediaries will be fully aware of the presence of the security gateway and can be configured to implicitly trust that it has checked earlier signatures and found them to be correct. Hence there is no need to be able to "undo" these changes, however there's still value in indicating which system made these changes.

3. Goals to be addressed

3.1. DKIM-replay

Because an email can currently be sent as "Bcc" such that there's no evidence in the message data of who the recipient is expected to be, it's possible to take a message that is correctly signed and replay it millions of times to different destination addresses as if they had been BCC'd. This message can be resent at any time.

DKIM2 headers will always have timestamps so that "old" signatures have no value.

A possibility to be investigated during testing is a "singleton" flag to allow senders to specify that this is a message for a single recipient (e.g. for authentication codes for billing transactions) and should not be expanded by mailing lists.

DKIM2 headers specify both "from" and "to" so that most opportunities to alter a message, re-sign it and replay it at scale will no longer be possible. Since the "to" address is always encoded in the email, any email to multiple recipients must be exploded by the sender, and each copy signed separately with different headers.

If the email is replayed (perhaps through a large system with many different customers) then if the email does not say that it has been duplicated then signatures can be assumed to be unique and hence simple caching (or Bloom filters) will identify replays. If the email has been duplicated then recipients can assign a reputation to the entity that did the duplication (along with the expected number of duplicates that will arrive from that entity) and assess duplicate signatures on that basis.

If the email is altered before duplication then it is again the case that this will be apparent to the recipient who can develop a reputation system for the entity that did the modification and replay.

3.2. Backscatter

The problem of backscatter, delivery status notifications sent to innocent third parties who had their address forged as the source of a message, has caused email recipients to implement a variety of countermeasures:

* in-band scanning: performing detailed analysis of the email content before replying to the DATA phase of the SMTP transaction, allowing immediate rejection but consuming resources on both ends of the connection, and limiting the time that can be used for the analysis to avoid timeouts.

* greylisting: replying with a temporary failure code to untrusted senders, allowing time to decide if the sender is trustworthy enough, but also delaying mail for an indeterminate period.

* delivery to "Spam" or "Junk" mailboxes - in some jurisdictions it’s not allowed to discard email that has been accepted, so providers must put the copy somewhere once they have accepted it, filling Junk mailboxes even if they’re very sure it’s bad.

By requiring bounce addresses to aligned with the most recent signature domain, we can avoid backscatter, allowing recipients to always take the message, and later return a bounce. This fulfils any legal obligation to inform the sender if the message isn’t delivered, while also avoiding the timeout and greylisting re-connection issue that currently exists, so messages are spooled for less time on intermediates, and recipients can take their time to analyse messages; even delivering the message to a mailbox and then upon receiving further intelligence, undoing the delivery and generating a bounce.

Privacy-preserving forwarding services will also see every bounce from any dkim2-supporting destination mailbox, allowing them to strip off the details of the further hop(s) and generate a bounce as if they had been the terminal node of the delivery and were just making a delayed decision.

4. Other areas of interest

4.1. Algorithmic dexterity

The final specification will require both RSA and elliptic curve be implemented for algorithmic agility. However this document acknowledges the long standing lack of adoption of elliptic curve ,and elliptic curve support may not be needed for development. The specifications will provide support for multiple algorithms. If there is IETF consensus around a "post-quantum" scheme then that will also be included.

Dexterity will become essential if advances in cryptanalysis cause a particular type of algorithm to become deprecated. To allow a phased switch away from such an algorithm we will make provision for more than one signature to be present in a single DKIM2 header. Systems capable of checking both signatures will require both to be correct. If only one signature is correct then email will be rejected with a clear message -- allowing interworking issues to be easily debugged.

To allow for future dexterity, it makes sense to allow multiple signatures with the same or different algorithms, from the same domain, on the same message.

4.2. Sender indications of intent

Having a way to indicate "the sender wants you not to make any modifications to this message" will allow senders to indicate the same intent they current achieve with a DMARC p=reject policy to stop messages which don’t have a verifying DKIM signature.

Having a way to indicate "this message is for a single recipient" has been requested by some services like document signature services.

Having a way to indicate "this message will be useless after time X" will be useful for things like confirmation codes which have limited validity, allowing intermediate systems to return the message if they haven’t been able to complete delivery by the expiry time.

4.3. Signer requests for feedback

Each signer on the chain may wish to receive feedback about messages, in the way that they currently use multiple DKIM signatures along with DMARC policies.

We can add a flag to allow intermediate signers (email sending providers, mailing lists, forwarders, etc) to say whether they wish to receive feedback about each message that they sign.

4.4. Simplification of signed header list

Currently DKIM signatures list a particular number of copies of each header field which are included in the signature, and the signer can choose exactly which headers to sign.

It is both valuable to mandate a set of headers, and the existence of a change algebra will allow us to insist that all copies of a named header field are always signed, reducing the risk of header stuffing attacks.

5. Security

TBA

6. IANA Considerations

TBA

7. Normative 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>.

[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>.

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

[SMTP] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/rfc/rfc5321>.

Appendix A. Changes from Earlier Versions

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

A.1. draft-ietf-dkim-dkim2-motivation-02:

* Updated the background and motivations to be more about the problems

* Moved "simplification of signed header list" into the "other areas", it’s not core to solving the underlying problems.

A.2. draft-ietf-dkim-dkim2-motivation-01:

* saying DKIM1 is silly, just calling it DKIM

* updated DKIM reference to RFC6376

* use named references

A.3. draft-ietf-dkim-dkim2-motivation-00:

* no changes other than the name

A.4. draft-gondwana-dkim2-motivation-03:

* typo fixes

* updated title

* allowed for multiple recipients to be signed (but still all legtimiate recipients MUST be explicitly signed)

* rewrote to be more "motivation/goals" and less "implementation design"

* removed the ’obsoletes’

A.5. draft-gondwana-dkim2-motivation-02:

* changed section title because DKIM1/2 do not really interwork as such

* removed implementation details, this is the motivation doc

* significant rewrite based on feedback from mailing list

A.6. draft-gondwana-dkim2-motivation-01:

* remove the z= parameter on the grounds that it adds too much complexity

* document that messages MUST NOT re-enter the DKIM2 world once the chain has been broken.

A.7. draft-gondwana-dkim2-motivation-00:

* initial version

Authors’ Addresses

Bron Gondwana
Fastmail
Email: brong@fastmailteam.com

Richard Clayton
Yahoo
Email: rclayton@yahooinc.com

Wei Chuang
Google
Email: weihaw@google.com