Internet Engineering Task Force
Internet-Draft
Updates: 6376 (if approved)
Intended status: Informational
Expires: 14 April 2026

S. Nurpmeso, Ed.
11 October 2025

DKIM Access Control and Differential Changes
draft-nurpmeso-dkim-access-control-diff-changes-09

Abstract

This document specifies a DKIM (RFC 6376) extension that allows cryptographic verification of SMTP (RFC 5321) envelope data, and of DKIM signatures prior to IMF (RFC 5322) message content changes along the message path, addressing thus security glitches, and offering a new world of email solutions that move complexity away from lower network layers, where problems cannot be solved, complying with David Wheeler’s fundamental theorem of software engineering, that "We can solve any problem by introducing an extra level of indirection". It updates DKIM to obsolete certain aspects that reality has proven to be superfluous, incomplete, or obsoleted. It is the future of email for email of the future.

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 14 April 2026.

Copyright Notice

Table of Contents

1. Introduction

DKIM[RFC6376] was not designed to cover SMTP[RFC5321] envelope data, allowing replay of valid, verifiable messages to an infinite set of recipients by malicious third parties, undetectable by sender and recipients. (To aid SMTP delivery to recipients in various conditions the optional "x=" expiration tag timestamp must be chosen so far in the future that malicious players have plenty of time to misuse messages.)

Whereas DKIM[RFC6376] standardized rudimentary, incomplete approaches to undo modifications of IMF[RFC5322] message content that happen along the message path, the overall design was agreed in not to survive them (compare, for example, [RFC6377]). The resulting paradigm of DKIM is "as long as one signature can be verified cryptographically, DKIM verification will succeed". This is problematic as message content changes may be falsely attributed to (the) address(es) in the IMF originator field(s). (Later policy-enforcing standards effectively complicated the situation, in that false attribution may now technically be avoidable, but mitigations of practice like "user A via B" will still be attributed to "A" by a human for one, and, in short, anything is valid if one DKIM signature is.)

Potentially many DKIM signatures may exist in a message. DKIM[RFC6376] gives hints on how verification can be performed, but, in practice, mitigations are applied in order to reduce excessive and useless verifications on hops down the message path: elder signatures are removed, or renamed, as changes are performed on message content, for example, by mailing-lists. An approach to avoid excessive network traffic and CPU work during message verification mitigates careless configurations.

The presented ACDC extension addresses these and more issues, backward and forward compatible, easy adoptable, and easy integratable into the current, existing infrastucture.

1.1. Conventions and Terminology

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.

When in the below message "REJECT"ion is said, implementations may choose to instead move messages into a spam or quarantine state.

The term "FOSS" refers to Free and Open Source Software.

2. DKIM Vivid Adjustments

The adult, mature, and rock solid foundation of DKIM[RFC6376] is built upon. But this document obsoletes certain unused / incomplete aspects, and adjusts certain vivid parts of the DKIM core, as follows. The full context of the changes will become clear as the remains of this document unfold.

* The signature expiration tag "x=" is no longer optional. It MUST be used within DKIM-Signature: header fields to place a lifetime constraint when creating signatures.

  The maximum "t=" to "x=" delta MUST NOT be greater than 864000 seconds (ten days: to reach into the next working week). Example delta values for tag auto-generation may be the bounce defaults 432000 seconds (five days: used for example by the Mailman2 and mlmmj mailing-list managers and the postfix MTA), 345600 seconds (four days: OpenSMTPD MTA), 172800 seconds (two days: Exim MTA).

  | _Informative remark:_ The DKIM section 5.2 defined "reasonable
  | validation interval before [keys are] being removed from the
  | key server" is being pointed to in that respect.

  | _Informative remark:_ For internal "ingress" DKIM-Signature:
  | header fields (with the "I" flag, as below, set) this limit
  | does not apply: it MAY be as high as local policies desire, in
  | order to support delayed verification of validation results.

* The advices of DKIM section 5.4 and 5.4.1, Determine the Header Fields to Sign and Recommended signature content, respectively, still apply. To address MIME[RFC2045] attacks the term "highly advised" regarding MIME header fields is, however, emphasized further: MIME header fields SHOULD NOT remain unprotected.

  IANA is asked to introduce a registry of further header fields to be included in cryptographic signature protection. This document hereby includes the Author Header Field[RFC9057] in both lists.

* ACDC, as below, aware software is urged to "oversign" aka "seal" aka sign fields that are not present at the time of signing, how DKIM (section 3.5, "h=" tag description) calls it. Modifications and/or additions of "regular" intermediates will then be covered by their "regular" signature, and the differences will be discoverable via ACDC.

* ACDC, as below, aware software SHOULD use the AUID (DKIM, section 3.5, "i=" tag description). The value SHOULD enable the Signer to identify the originator for whom it creates the signature.

* ACDC aware software is suggested to offer configuration directives that ensure that the SMTP[RFC5321] envelope MAIL FROM matches the first address of the IMF[RFC5322] From: header field.
  // TODO: Or the formerly usual From:/Sender: paradigm, shall that be
  // revived from its DKIM- and DMARC-destructed state

* DKIM section 3.7 defines how "Computing the Message Hashes" has to be performed. Different to the RSA algorithm (RFC 8017) solely defined for DKIM at the time section 3.7 was written, modern algorithms include checksumming themselves. Section 3.7 is hereby modified in that the input to "sig-alg", the "data-hash", can adapt to standardized algorithms as appropriate. If an algorithm chooses adaption, "hash-alg" is only used to produce the "body-hash", whereas the input formerly used to create the "data-hash" is fed in full into "sig-alg", instead of to "hash-alg". More formally, the new pseudo-code for the signature algorithm is:

```
body-hash = hash-alg (canon-body, l-param)
data-hash = hash-alg (h-headers, D-SIG, body-hash)
signature = sig-alg (d-domain, selector,
                     data-hash / (h-headers, D-SIG, body-hash))
```

3. DKIM ACDC

The DKIM[RFC6376] extension Access Control and Differential Changes:

* Places DKIM signatures in an ordered, numbered, random-accessible sequence which’ state correlate. Identical DKIM signatures generated at the same hop, but which differ in only the used algorithm, share, however, a sequence number. With ACDC it can be, and usually is, sufficient to verify only the cheaply detectable highest numbered signature.

* Adds reversible data difference tracking, and as such supports cryptographical content verification of any (ACDC aware) intermediate message state, up to the initial variant as sent by the originator.

* Cryptographically protects the SMTP[RFC5321] envelope, that is, RCPT TO addresses as well as the MAIL FROM address. Replay of valid messages to initially not addressed recipients, as well as backscatter bounces to random addresses instead of the originator, become detectable.

* Allows cryptographically verifiable collection of statistics of organizational trust ([RFC5863], section 2.5) along the entire message path.

* Allows recognition of certain flagged conditions (along the message path) only by looking at the highest numbered signature.

The DKIM[RFC6376] extension Access Control and Differential Changes is announced by adding an "acdc=" tag to the DKIM-Signature: header field. (For efficiency reasons it SHOULD be placed early, before tags like "h=", "bh=" and "b=", for example.)

The tag starts with "sequence", a decimal number starting at 1, or incremented by 1 from the highest ACDC "sequence" encountered in the message; the maximum value is 999: if incrementing would result in overflow, the message MUST be rejected; detected sequence holes MUST also cause rejection (but see below); in both cases SMTP[RFC5321] reply code 550 is to be used; with enhanced SMTP status codes[RFC3463] 5.5.4 MUST be used.

Multiple DKIM-Signature: header fields with the same "sequence" MAY be generated by a domain, in which case each field MUST use a different "s=" selector and algorithm. To allow for key rotation flexibility while addressing a possible denial of service attack surface, maximally three selectors per algorithm MUST be used.

Flag description is normative. (Note the missing FWS separators around =.) ABNF[RFC5234]:

```
acdc = %x61 %x63 %x64 %x63 = sequence ":" 1*flag [ ":" id ]
sequence = 1*3DIGIT; DIGIT from RFC 5234
flag = "A" / "D" / "E" / "I" / "O" / "P" / "R" / "S" / "s" /
       "V" / "v" / "X" / "x" / "Y" / "y" / "Z" / "z"
id = *42(ALPHA / DIGIT / "+" / "-"); optional (bounce) identifier
```

A Alongside the "V" flag (see there) only: all existing signatures were verified.

D The message was modified at this hop, differential changes were generated, and are stored in a DKIM-DC: header field.

   The "Y" flag has to be set.

E The SMTP[RFC5321] envelope (MAIL FROM and/or RCPT TO) was modified. A new "Access Control" (see there) evaluation has been performed.

   The "O" flag has to be set if the MAIL FROM changed. The "y" flag has to be set.

I This DKIM-Signature: header field was generated at ingress. Special rules apply to these signatures, for example unlimited "x=" tag expiration.

> | _Informative remark:_ Such fields offer a cryptographically
> | verifiable message state authentication contract: for as long
> | as the ("local") "s=" selector announced key is available, the
> | message state at the time it entered the "local" email
> | processing system is assurable by for example user interfaces.

All signature instances with this flag set MUST be removed when messages enter and leave the email system.

O This hop claims the message origin.

This either means that the message originated at this hop, in which case the signature (usually, DKIM-typical) refers to the first address of the From: header field, and the "sequence" is 1.

It can also mean that the current hop was the, quoting [RFC3461], _"final delivery for the [original] message"_, that the message got a _"new envelope return address"_, that is, the MAIL FROM of the SMTP envelope was changed. In this case the "E" flag has to be set.

An "Access Control" (see there) evaluation has been performed.

P Postmaster mode. With this flag set the behavior of ACDC borders test mode in that rejections must not occur (due to ACDC). This is to allow for a communication possibility window in a situation where messages would always be rejected, due to misconfigurations et cetera, and as such reflects SMTP[RFC5321] section 4.5.1 Minimum Implementation.

If the "sequence" is 1, message recipients have to be inspected. If the IMF[RFC5322] header fields To: and Cc: only contain a single addressee with the local part postmaster[RFC1123], and if the same "postmaster" is addressed as a SMTP[RFC5321] RCPT TO recipient, and if no more than two RCPT TO recipients exist in total, then the "P" flag has to be set.

Once set, all future ACDC signatures must copy it. It MUST, however, be removed when in conjunction with the "D" and/or "E" flags the according SMTP envelope conditions are no longer satisfied.

R Reputation check to collect organizational trust ([RFC5863], section 2.5) along the signature chain was performed.

On top of the "V" (and possibly "A") flag(s) this means that all differential changes have been applied, and all signatures along the chain have been verified, and the entire chain validated correctly.

Only in signatures with a "sequence" greater than 1, and without the "Z" or "z" flag.

| _Informative remark:_ The presence of "R" reveals local state publically; however, in a chain of trust this seems desirable even. The use of organizational trust may for example mean to perform full reputation checks more and more sparingly, the higher the trust, falling back to only random checks.

S Only in conjunction with the "I" flag: upon ingress the SPF[RFC7208] state was successfully verified.

s Only in conjunction with the "I" flag: upon ingress the SPF[RFC7208] verification failed.

V ACDC signature verified successfully.

The signature with the highest "sequence" has been verified correctly, the sequence of ACDC signatures is complete, and their flags make sense (in the sequence). In conjunction with the flag "R" even deeper inspection was performed.

If multiple signatures with the same highest "sequence" exist, the verifier behavior is unspecified in that "V" signals success: at least one signature was checked, and all tested signatures verified successfully. If however all signatures were verified, the "A" flag MUST be set; in single-signature cases the "A" flag MAY be omitted.

Only in signatures with a "sequence" greater than 1.

v DKIM signature verified successfully.

In signatures with "sequence" 1, then missing the "O" flag, it means the message originated at a non ACDC aware hop, and normal DKIM processing was performed and succeeded. Unless DKIM processing succeeded for the DKIM signature which covered the messages' From: header field address, the "Z" flag must be set, otherwise the "z" flag.

In messages with a higher "sequence" it comes alongside the "X" flag: necessarily the ACDC chain was broken, and the message changed, by an intermediate non ACDC aware hop. The "z" flag must be set.

X ACDC verification failed ("V" flag conditions not met); however, the normal DKIM signature verification was performed, and succeeded.

The "z" flag must be set.

x DKIM verification failed.

In signatures with "sequence" 1, then missing the "O" flag, it means the message originated at a non ACDC aware hop, and normal DKIM processing was performed and failed. The "z" flag must be set.

In messages with a higher "sequence" it comes alongside the "X" flag: necessarily the ACDC chain was broken, and the message changed, by an intermediate non ACDC aware hop. The "z" flag must be set.

Y The message has seen IMF[RFC5322] modifications: somewhere along the chain the message data was modified. Once set, all future ACDC signatures must copy it.

y The message has seen SMTP[RFC5321] envelope modifications: somewhere along the chain the envelope was modified. Once set, all future ACDC signatures must copy it.

Z Announces the ACDC chain is incomplete. The message was processed by ACDC unaware hops. However, the message verifies correctly and seems to have never been modified non-reversibly. Once set, all future ACDC signatures must copy it, unless later downgraded to the "z" flag.

z The message has seen non-reversible modifications, and cannot be cryptographically verified back to its origin. Once set, all future ACDC signatures must copy it.

If this flag is set ACDC looses its decisive meaning and "degrades" to normal DKIM (except for access control): no more differential data is generated, and messages are distributed further / accepted if just any signature verifies. (Software configuration MAY allow otherwise.)

"id" The optional "bounce identifier" offers enough room to store

Universally Unique IDentifiers[RFC9562].

It may be generated to help sending domains to uniquely identify messages within the DKIM "t=" and "x=" time delta, as well as to ensure that successively sent identical messages are not detected as being the same.

Receiving domains SHOULD NOT use this identifier due to the denial of service attack surface, regardless of collected organizational trust (see "R" flag).

Unknown flags MUST be ignored. Invalid flag combinations and flag misuse MUST result in rejection with SMTP reply code 550; if enhanced status codes[RFC3463] are used, 5.5.4 MUST be used. (This includes the "P" flag upon incorrect use.)

4. The DKIM-Store header field

The DKIM-Store header field has no meaning in the email system. The sole purpose of mentioning it is to announce that it MUST be removed when messages enter and leave the email system.

It could for example be temporarily created and used by non-integrated mail filter (milter) software to pass informational data in between the "ingress" and the "egress" processing side. To aid in software bugs and possible configuration errors this specification enforces removal of all occurrences.

| _Informative remark:_ In order to achieve locality it is suggested
| to encrypt data passed around in this temporary header field with
| a key internal to the "local" email processing system.

5. Access Control

SMTP[RFC5321] delivers messages to individual domains. With ACDC, when a SMTP envelope was created or changed, all distinct domain-names found within the list of intended SMTP RCPT TO addressees are collected, as the message needs to be forged on this individual domain base: ACDC will create DKIM-AC: header fields covering SMTP envelopes, and include them as messages are sent to individual domains.

The domains _dkimacdc DNS entries, as below, are queried. Any domain that announces ACDC support can be served by a single message for all recipients (possible limits, as below, aside).

For other domains, to guarantee anonymity, it is necessary to differentiate in between public recipients in the To: and Cc: header fields, and recipients not covered by these header fields. However, another anonymous approach is presented below.

Remarks: quality-of-service: for simplicity messages may always be forged on a single recipient base, individually.

In any case the completely prepared message, including the readily prepared DKIM-Signature:(s), is forged, (a) DKIM-AC: header field(s) is/are generated which cover(s) the logical recipient subset, and the resulting message is then sent.

ACDC aware recipient domains are expected to manage a message DKIM-AC: identity cache to mitigate replay attacks. (Hint: the DKIM-AC: signature seems like a natural cache key source, see below.) The now mandatory and constrained "x=" tag allows for finite identity cache sizes.

To keep the identity cache a write-once data structure, ACDC senders MUST NOT generate DKIM-AC: header fields with more than half of the 100 recipients that SMTP[RFC5321] section 4.5.3.1.8 guarantees as a minimum, unless a DNSSEC[RFC4033][RFC4034][RFC4035] protected, or otherwise known trustworthy, _dkimacdc DNS entry announced a different limit.

If more recipients need to be addressed on a single domain, multiple message forges with recipient subsets must be generated: like this each message forge is "atomic", and the DKIM-AC: header field covers all the SMTP envelope.

| _Informative remark:_ Implementations MAY offer configuration
| options to specify other (higher, lower) recipient limits. Like
| this the much higher limits in actual use (for example, the Exim
| MTA default is 50000) can be utilized.

ACDC senders MUST be capable to deal with SMTP[RFC5321] section 4.5.3.1.10 Too Many Recipients Code errors signalled via SMTP reply code(s) 452 and 552 regardless of above limits. Whether implementations switch to single recipient delivery, use some kind of "nearing target approach" strategy, etc, is quality-of-service and not furtherly specified.

| _Informative remark:_ Notifying a local postmaster[RFC1123]. for
| a 452 reply code in conjunction with a _dkimacdc DNS entry for
| possible local configuration adjustments (or recipient domain
| postmaster pings) may make sense. It could also be used for
| collecting organizational trust ([RFC5863], section 2.5), however.

An ACDC aware recipient domain that receives an "acdc=" tagged message without a DKIM-AC: header field MUST reject the message with SMTP reply code 550; if enhanced status codes[RFC3463] are used, 5.5.4 MUST be used.

It MUST likewise fail if the DKIM-AC: header field does not cover the SMTP envelope data. It MUST test for a superset of recipients, and only fail if an envelope recipient is not included in the DKIM-AC: header field.

It MUST reject messages which fail the signature check of a DKIM-AC: or DKIM-Signature: header field, or the condition and flag check verification, with SMTP reply code 550; the enhanced status code MUST be 5.7.7.

(Senders MAY use Delivery Status Notifications[RFC3461] to fine-tune the resulting behavior.)

5.1. The DKIM-AC header field

The syntax of this header field is the usual semicolon separated list of DKIM-style tags of unspecified order; unknown tags MUST be ignored. It is used to cryptographically link the SMTP envelope to the sent IMF[RFC5322] mail message.

The "sn=" tag is the linked DKIM-Signature: "sequence", best placed early. Multiple signatures with the same "sequence", but different algorithms may exist, and so may DKIM-AC: header fields.

The selector of the linked signature is given by the "s=" tag, the used algorithm can be deduced from there.

The "f=" tag is the SMTP[RFC5321] MAIL FROM of the covered message, the complete addr-spec.

The "d=" tag value is the recipient domain, with one to multiple "t=" tag(s) for the local-parts of RCPT TOs.

| _Warning:_ SMTP[RFC5321] address local-parts permit quoted-
| strings!

In case the recipient domain for a particular message forge has not announced support for ACDC, and to strengthen SMTP envelope anonymity in permanent IMF[RFC5322] message data, and very especially if the connection to the recipient domain does not involve intermediate hops, if detectable, the tag "d=" MAY (without intermediates), and the tags "f=", and any "t=" SHOULD be omitted, and instead an "ec=" tag be placed. The content of this tag is BASE64[RFC4648] encoded,

and furtherly unspecified. It is suggested to place the content of for example "f=" in an encrypted form, decryptable only with a key internal to the "local" email processing system.

Mirroring DKIM-Signature: the tag list is concluded with the "b=" tag that is the cryptographic signature data of the DKIM-AC: header field. However, the reassembled (see DKIM[RFC6376], section 3.5) "b=" value of the linked DKIM-Signature: is "virtually assigned", and included when creating the cryptographic signature; thereafter the "b=" tag is assigned its own value.

All instances of DKIM-AC: header fields MUST be removed by ACDC aware software as soon as possible: they MUST NOT be delivered by local delivery agents as part of the message. They MUST, however, exist in rejected messages.

However, if a domain is only an intermediate, which was neither directly addressed nor which originated the mail, and which does not modify the SMTP envelope either, then it MUST NOT remove the "current" DKIM-AC: header field, and it MUST NOT generate a new one.

5.2. The _dkimacdc.DOMAIN DNS TXT RR

The syntax of this DNS resource record is the usual semicolon separated list of DKIM-style tags of unspecified order; unknown tags MUST be ignored. However, FWS separation of tag, equal sign, and value is not allowed.

DNS CNAME chains MUST be followed when looking up this DNS RR.

The optional tag "rl=" contains an unsigned integer that asserts the guaranteed minimum number of recipients that may be used as RCPT TOs in a single transaction; it may be as small as 1. 0 or invalid values are treated as 1.

The tags "v=" and "a=" mirror their DKIM[RFC6376] (sections 7.5 and 7.6) tags, however, "v=" is optional, and none to multiple "a=" tags MAY exist: they indicate, in descending order, the most desirable algorithms for this domain. Senders SHOULD try to honour the first fit, and exclusively so if the algorithm is a well established one. (The exact definition is beyond this specification: for example, at the time of this writing, only RSA-SHA256 meets this requirement, ED25519-SHA256 does not.)

6. Differential Changes

Whenever an ACDC enabled domain detects during DKIM-Signature: creation that the relaxed representation of a message was modified along its flight from ingress to egress, for example, when it was processed by a mailing-list which tagged the subject and added a message footer, a DKIM-DC: header field has to be created.

| _Informative remark:_ In an unbroken chain of ACDC signatures the DKIM-DC: covered changes can be applied in reverse order of creation in order to cryptographically verify all intermediate message states and signatures, back to the original version as sent by the sender.

6.1. The DKIM-DC header field

The syntax of this header field is the usual semicolon separated list of DKIM-style tags of unspecified order; unknown tags MUST be ignored.

The "sn=" tag is the linked DKIM-Signature: ACDC "sequence", best placed early.

The "c=" tag identifies the compression method used for the data in "h=" and/or "b="; the value "z" means ZLIB[RFC1950], whereas "lzma2" means [LZMA2]. ZLIB MUST be supported by signers and verifiers, LZMA2 MUST only be supported by verifiers. (FOSS implementations of all compression types are available.)

The "h=" tag is used to store differential data for header fields, "b=" that for body content. Both tags are optional, but at least one MUST exists in a valid DKIM-DC: header field.

The differential data is the result of the BSDiff algorithm, as below, compressed with the method given in "c=", then BASE64[RFC4648] encoded.

| _Informative remark:_ The higher cost of using [LZMA2] compression can be amortized by the lesser necessary I/O. (It often can be deduced from prepared patch data and/or the number of recipients which compression is advisable.)

All header fields in the registry of headers to be included in cryptographic signature protection, as above, MUST be included. All header fields covered by the DKIM-Signature: MUST be included. All ACDC enabled DKIM-Signature: and DKIM-DC: (aka ditto DKIX-, as below) header fields MUST be included.

6.2. The BSDiff differential algorithm

The differential changes are created with the DKIM "relaxed" normalized header field and body data, respectively, as seen on egress, alongside the equally normalized data present before modifications took place, that is, on ingress.

The header fields MUST be sorted byte-wise (by numeric US-ASCII byte value) by-name, the formed subgroups MUST remain in the header stack order defined by DKIM[RFC6376] section 5.4.2, Signatures Involving Multiple Instances of a Field.

The BSDiff algorithm of Colin Percival, which has excellent characteristics (and sees broadest use for decades), is then used to create a binary delta of the header or body lines.

| There is a FOSS [BSDIPA] plug-and-play ISO C99 and perl
| implementation available that iterated the FreeBSD operating
| system implementation of BSDiff, and includes further references
| on the algorithm.

6.2.1. BSDiff adaption

* First of all: the string suffix sorting and difference creation approach of Colin Percival has been left unchanged.

* The original had been fixated on 64-bit file sizes and content representation, the adaption adds a 32-bit variant that almost halves memory usage, and produces smaller patch control data. It is deemed sufficient for email purposes. (32-bit and 64-bit patches are not interchangeable.)

* In order to reduce memory usage during patch generation, the adaption uses a shared memory region for differential and extra data: the former is therefore stored in reversed order, top down. (Reduces memory usage by the size of the target data set.)

* The adoption stores data in big endian (network; MSF; most significant byte first) instead of little endian (LSF; least significant byte first) byte order.

* The original uses three separate bzip2 streams to serialize control, differential and extra data. The adaption separated patch generation from the I/O layer, which sees the entire readily prepared patch data, as below.

* The original header did not contain the size of the extra data, which was stored last, with its size implicitly extending to the end of the patch. The adaption includes the extra data size in the header, allowing more verification tests to be applied with only the header being readily parsed. This also enables the I/O layer to allocate perfectly sized memory with only the header data being available.

6.2.2. Patch content

Overall, the patch consists of the header, followed by the control data. Thereafter the two byte (8-bit octet) streams of differential data (in reverse order) and extra data conclude the patch.

The header and the control data consist of 32-bit signed integers, stored in big endian byte order (as above).

The header consists of four values denoting the length of the control block in bytes, the length of the difference data block, the length of the extra data block, concluded by the length of the original data source; The sum of the first three values must be one less than the maximum positive 32-bit signed integer. Control data copy instructions also do not exceed this value.

The control data is a stream of tuples of three values each, the first denoting the length of differential data to copy in bytes, the second that of extra data to copy; the read positions within the differential and extra data move by the same amount of bytes. The last value denotes the number of bytes to seek relatively in the data source after the copying (if any) has taken place: of all the values, only this one may be negative.

6.3. Rationale

Differences are included to allow DKIM verifiers to restore previous message content for the purpose of cryptographically verifying elder DKIM-Signature: header fields.

This for example allows for collecting trustworthy statistics of organizational trust ([RFC5863], section 2.5). Or user interfaces may visually restore an initial From: header field when messages come from a known mailing-list.

For example, user interfaces could use traffic light semantics that unfold on click to traffic light semantics of all message versions, which would (with precautions) visualize differences: this can empower users to make decisions on the trustworthiness of intermediates, and to, for example, enable the above mentioned From: header field restoration.

However, the data exists in the DKIM "relaxed" normalized variant, former states are not meant to be usable messages by themselves. For example some embedded OpenPGP signature and text couple would likely fail to verify, dependent upon the original MIME transfer encoding.

| _Informative remark:_ This was deemed acceptable because of the
| purpose of including differential changes, and because a
| visualization of the DKIM covered message should still be
| sufficient to allow users making responsible decisions.

Finally, the given example will likely verify as part of the complete received message, unless altered along the SMTP path: ACDC can ideally say where (and exactly what, in an unbroken ACDC chain).

7. Mitigations for Future

As of the time of this writing the email infrastructure is deeply divided due to standards like DMARC and SPF, which require mitigations to be applied in order to keep existing infrastructures in a usable state.

For example, SPF will not survive a single hop, which means that alias expansion, a widely used core feature of the email infrastructure, does no longer work. The IETF has no solution for this problem, but the FOSS scene has created a "Sender Rewriting Scheme" so that aliases can be used regardless.

As another example, DMARC causes a lot of mailing-lists to apply mitigations of various form and style: old DKIM signatures are removed, or renamed, often the From: header field is rewritten in a "User A via List B" style.

7.1. ACDC mitigations

This memo suggests to apply mitigations actively as part of DKIM processing, at minimum temporarily, until, at some future time, the email infrastructure has adapted to a new reality. Future engineers can then decide how to proceed further.

In any case it seems wise to move decisions on actual content changes away from the SMTP layer, to reduce failures to cryptographic signature failures, and let users and/or algorithms in a higher layer decide whether a certain content change or applied mitigation is "acceptable", or not.

* Rename DKIM-Signature: header fields to DKIX-Signature:. This mitigation MUST be applied. (With a possible exception of the first, as below.)

  Because DKIM-Signature: header fields are removed or renamed, also by unanchored regular expressions, which would match for example "EDKIM-Signature", ACDC aware software should rename all elder ACDC enabled DKIM-Signature: header fields to DKIX-Signature: upon egress.

  Since only one DKIM-Signature: will have to be verified successfully by non ACDC aware DKIM software, and ACDC aware software can toggle the single byte back before verifying elder signatures, this should be easy in practice: just treat DKIM-Signature: and DKIX-Signature: alike, but toggle before cryptographic verification.

  For backward compatibility the most portable algorithm should remain in the single-instance DKIM-Signature:. If multiple DKIM-Signature:s are created, mitigation SHOULD therefore be applied even when newly generating signatures.

* Mitigate non-local MAIL FROM envelopes.

  Because a possible SPF check will fail on the next hop (in situations with strict and, thus, seriously, the only conceivably useful SPF policies) if a message that does not originate locally leaves the email system on egress, with a SMTP envelope MAIL FROM of a foreign domain, mitigate such addresses. so that the current hop becomes the, quoting [RFC3461], _"final delivery for the [original] message"_. DKIM software SHOULD offer options to exclude certain domains from these mitigations.

  To mitigate, synthesize for example an address of the local domain with a local-part starting with DKIM=, followed by at least 16 bytes of the BASE64[RFC4648] encoded HMAC[RFC2104] of a dedicated cryptographic private key and the original MAIL FROM. (The SMTP size limit of local-part is 64 octets, however the overall "reverse-path" limit of RFC 821 and RFC 2821 was 256 octets.)

As another example, using a dedicated subdomain is an approach that avoids any possible local-part ambiguities. Then for example the IETF mailing-list From: header field DMARC mitigation approach could be used, as in decomposing the original MAIL FROM into "local-part=40domain", followed by the "subdomain.domain" of the mitigating host. SMTP size limits have to be respected again.

The synthesized address MUST be linkable to the original MAIL FROM for at least 864000 seconds (ten days: to reach into the next working week). It SHOULD be linkable only by message bounces which have a verifiable DKIM signature of the local domain, and it MAY only link for the signature of the exact message for which the address was synthesized. The optional bounce identifier "id" may be usable for this purpose.

* Mitigate From: header fields, if necessary.

  When a message was changed in between ingress and egress, so that the DKIM signature related to the From: header field will no longer verify. Then, if the From: header field was not already locally mitigated (by for example mailing-list software), actively mitigate the From: header field, so that the current hop becomes the, quoting [RFC3461], _"final delivery for the [original] message"_ in respect to the IMF[RFC5322] message that is visible to recipients. DKIM software SHOULD offer options to exclude certain domains from these mitigations.

  To mitigate, place the original name-addr in the Reply-To: header field, unless that already exists, and replace From: with synthesized content. The examples of non-local MAIL FROM envelope mitigation apply also here in respect to addr-spec; yet, the dedicated subdomain approach results in visually more appealing header field content. For the display-name a "From: X via Y" notation MAY be used, where "X" denotes the original display-name.

  For example, if the original content was "Forename Surname <for.sur@example1.net>" then the mitigation could be "Forename Surname via @dkim.example2.net" <for.sur=40example1.net@dkim.example2.net>, or even the very direct "Forename Surname" <for.sur=40example1.net@dkim.example2.net> that the IETF mailing-lists use to mitigate DMARC.

8. Example

An example that shows the flow of a single message with multiple different recipients, including mailing-lists and aliases. It assumes all recipients support ACDC, but does not assume direct, TLS secured connection to recipient domains, therefore it keeps DMARC compatibility and plain DKIMv1 and SPF compatibility, as noted, which results in quite some DKIM-/DKIX- duplication. In a all-ACDC world only DKIX- would remain, or only ever one DKIM-Signature:, the last, as no DMARC rules apply no more. SPF would only need to announce -all, therefore going for the strictest possible mode.

Originator (yet forged for recipient domain f.g):

```
MAIL FROM: <a@b.c>
RCPT TO: <d@f.g>
RCPT TO: <e@f.g>
...
DKIM-AC: sn=1; s=K1; f=a@b.c; d=f.g; t=d; t=e; b=...
DKIM-Signature: acdc=1:O; s=K1; ...
DKIX-Signature: acdc=1:O; s=K1; ...
From: a@b.c
To: d@f.g, e@f.g, x@y.z, u@v.w, r@s.t, o@p.q
...
```

f.g, local delivery (to d@ and e@):

```
...
DKIM-Signature: acdc=2:AIV; ...
DKIX-Signature: acdc=1:O; s=K1; ...
From: a@b.c
...
```

x@y.z -- a mailing-list!
It redistributes after RFC 2369 and RFC 2919 additions, in-message unsubscribe footer, and From: mitigated (in best RFC 3461 manner):

```
MAIL FROM: <x@y.z>
RCPT TO: <l@m.n>
...
DKIM-AC: sn=2; s=K2; f=x@y.z; d=m.n; t=l; b=...
DKIM-DC: sn=2; c=lzma2; h=BASE64; b=BASE64
DKIM-Signature: acdc=2:ADEOVYy; s=K2 ...
DKIX-Signature: acdc=2:ADEOVYy; s=K2 ...
DKIX-Signature: acdc=1:O; s=K1; ...
From: a(AT)b(DOT)c via x@y.z <x@y.z>
Reply-To: a@b.c
```

```
    ...
    List-Unsubscribe: bla

  u@v.w -- an expanded alias!
  The hop honours RFC 3461, and changes MAIL FROM;
  it keeps DKIM-Signature: acdc=1 for DMARC compatibility:

    MAIL FROM: <u@v.w>
    RCPT TO: <realu@realv.realw>
    ...
    DKIM-AC: sn=2; s=K2; f=u@v.w; d=realv.realw; t=realu; b=...
    DKIM-Signature: acdc=2:EOVy; s=K2 ...
    DKIX-Signature: acdc=2:EOVy; s=K2 ...
    DKIX-Signature: acdc=1:O; s=K1; ...
    DKIM-Signature: acdc=1:O; s=K1; ...
    From: a@b.c
    ...

  r@s.t -- an expanded alias!
  Note: not mitigated, will later fail SPF;
  it keeps DKIM-Signature: acdc=1 for DMARC compatibility:

    MAIL FROM: <a@b.c>
    RCPT TO: <realr@reals.realt>
    ...
    DKIM-AC: sn=2; s=K2; f=a@b.c; d=reals.realt; t=realr; b=...
    DKIM-Signature: acdc=2:EVy; s=K2 ...
    DKIX-Signature: acdc=2:EVy; s=K2 ...
    DKIX-Signature: acdc=1:O; s=K1; ...
    DKIM-Signature: acdc=1:O; s=K1; ...
    From: a@b.c
    ...

  ..therefore with mitigation,
  also to comply with RFC 3461;
  it keeps DKIM-Signature: acdc=1 for DMARC compatibility:

    MAIL FROM: <DKIM=a=40b.c@s.t>
    RCPT TO: <realr@reals.realt>
    ...
    DKIM-AC: sn=2; s=K2; f=DKIM=a=40b.c@s.t; ...
    DKIM-Signature: acdc=2:EOVy; s=K2 ...
    DKIX-Signature: acdc=2:EOVy; s=K2 ...
    DKIX-Signature: acdc=1:O; s=K1; ...
    DKIM-Signature: acdc=1:O; s=K1; ...
    From: a@b.c
    ...
```

```
   o@p.q -- a mailing-list!
   It redistributes after RFC 2369 and RFC 2919 additions,
   in-message unsubscribe footer, without From: mitigation.
   Note: not mitigated, will later fail DMARC!

     MAIL FROM: <o@p.q>
     RCPT TO: <X@X.X>
     ...
     DKIM-AC: sn=2; s=K2; f=o@p.q; d=X.X; t=X; b=...
     DKIM-DC: sn=2; c=lzma2; h=BASE64; b=BASE64
     DKIM-Signature: acdc=2:DEOVYy; s=K2 ...
     DKIX-Signature: acdc=2:DEOVYy; s=K2 ...
     DKIX-Signature: acdc=1:O; s=K1; ...
     From: a@b.c
     ...
     List-Unsubscribe: bla

   ..therefore with mitigation:

     MAIL FROM: <o@p.q>
     RCPT TO: <X@X.X>
     ...
     DKIM-AC: sn=2; s=K2; f=o@p.q; d=X.X; t=X; b=...
     DKIM-DC: sn=2; c=lzma2; h=BASE64; b=BASE64
     DKIM-Signature: acdc=2:DEOVYy; s=K2; d=dkim.p.q ...
     DKIX-Signature: acdc=2:DEOVYy; s=K2; d=dkim.p.q ...
     DKIX-Signature: acdc=1:O; s=K1; ...
     From: "a@b.c" <a=40b.c@dkim.p.q>
     Reply-To: a@b.c
     ...
     List-Unsubscribe: bla

   l@m.n (recipient of x.y.z mailing-list), local delivery:

     ...
     DKIM-Signature: acdc=3:IVYy;
     DKIM-DC: sn=2; c=lzma2; h=BASE64; b=BASE64
     DKIX-Signature: acdc=2:ADEOVYy; s=K2 ...
     DKIX-Signature: acdc=1:O; s=K1; ...
     From: a(AT)b(DOT)c via x@y.z <x@y.z>
     ...

   realr@reals.realt -- expanded alias target, local delivery:

     ...
     DKIM-Signature: acdc=3:IVy;
     DKIX-Signature: acdc=2:EOVy; s=K2 ...
     DKIX-Signature: acdc=1:O; s=K1; ...
```

```
From: a@b.c
...
```

9. IANA Considerations

IANA is asked to create a registry of header fields that shall be cryptographically covered by DKIM/ACDC. This memo extends the list mentioned by DKIM[RFC6376] with the Author Header Field[RFC9057].

10. Security Considerations

Public-key cryptography is the safest approach to identification of counterparts and verification of data. This specification enables DKIM to cryptographically verify SMTP envelopes, and to cryptographically verify all message transitions back to the original message sender.

11. References

11.1. Normative References

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[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/info/rfc6376>.

11.2. Informative References

[RFC1123] Braden, R., Ed., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, DOI 10.17487/RFC1123, October 1989, <https://www.rfc-editor.org/info/rfc1123>.

[RFC1950] Deutsch, P. and J. Gailly, "ZLIB Compressed Data Format Specification version 3.3", RFC 1950, DOI 10.17487/RFC1950, May 1996, <https://www.rfc-editor.org/info/rfc1950>.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <https://www.rfc-editor.org/info/rfc2045>.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[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/info/rfc2119>.

[RFC3461] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, DOI 10.17487/RFC3461, January 2003, <https://www.rfc-editor.org/info/rfc3461>.

[RFC3463] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 3463, DOI 10.17487/RFC3463, January 2003, <https://www.rfc-editor.org/info/rfc3463>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

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

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

[RFC5863] Hansen, T., Siegel, E., Hallam-Baker, P., and D. Crocker, "DomainKeys Identified Mail (DKIM) Development, Deployment, and Operations", RFC 5863, DOI 10.17487/RFC5863, May 2010, <https://www.rfc-editor.org/info/rfc5863>.

[RFC6377] Kucherawy, M., "DomainKeys Identified Mail (DKIM) and Mailing Lists", BCP 167, RFC 6377, DOI 10.17487/RFC6377, September 2011, <https://www.rfc-editor.org/info/rfc6377>.

[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/info/rfc7208>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9057] Crocker, D., "Email Author Header Field", RFC 9057, DOI 10.17487/RFC9057, June 2021, <https://www.rfc-editor.org/info/rfc9057>.

[RFC9562] Davis, K., Peabody, B., and P. Leach, "Universally Unique IDentifiers (UUIDs)", RFC 9562, DOI 10.17487/RFC9562, May 2024, <https://www.rfc-editor.org/info/rfc9562>.

[BSDIPA] "BSDIPA, a mutation of BSDiff", <https://github.com/sdaoden/s-bsdipa>.

[LZMA2] "LZMA2: The .xz File Format", <https://tukaani.org/xz/xz-file-format.txt>.

Appendix A. Further DKIM Updates

* This specification obsoletes the simple canonicalization type; It MUST NOT be used by software announcing ACDC.

  Rationale: in order to minimize processing cost in time and space for and of differential processing, being able to work on and with only one data representation is beneficial. The "extremely crude ASCII Art attacks" mentioned in DKIM[RFC6376] section 8.1 are considered to be a rather artificial attack vector.

* This specification obsoletes the DKIM "l=" tag that restricts the number of DKIM covered bytes of the normalized message body. This tag MUST NOT be used by software announcing ACDC support, and all the message body MUST always be used to create the body hash.

  Rationale: "l=" has always been insufficient to deal with message changes caused by mailing-lists etc, but effectively includes the security risk that message parts which are not covered by the signature appear as "valid content" to users looking at a DKIM verified message. The ACDC differential changes offer a better approach to deal with message changes, while completely covered message bodies ensure content validity.

* For the "i=" tag this specification obsoletes the possible use of DKIM-Quoted-Printable for the optional Local-part.

  Rationale: because the syntax is "a standard email address where the local-part MAY be omitted", quoted-printable encoding is not necessary for representation.

* This specification obsoletes the DKIM "z=" tag that was defined "for diagnostic use" to copy a freely defined set of header fields and their values present during signature creation. This tag MUST NOT be used by software announcing ACDC.

  Rationale: the ACDC differential changes provide access to the same information.

* For the "q=" tag this specification obsoletes the possible use of DKIM-Quoted-Printable for the optional x-sig-q-tag-args of possibly introduced future query types.

  Rationale: shall ever a new type become standardized beside the dns/txt that is with DKIM from the very start, that standard can very well give meaning to a hyphenated-word proxy identifier without making use of byte values which would require encoding.

* This specification obsoletes the DKIM key representation tag "n=" that was meant to include "notes that might be of interest to a human", "intended for use by administrators, not end users", and which "should be used sparingly".

  Rationale: no use case has been encountered in the DNS, let alone serious such; if future space unconstrained key providers other than DNS should ever exist and be used to distribute DKIM keys, it is likely that they support inclusion of strings via some method that need not be included in the DKIM key representation itself.

* Because above changes remove all use cases for the dkim-quoted-printable encoding defined in RFC 6376 2.11, this specification obsoletes the DKIM-Quoted-Printable encoding.

* This specification obsoletes the use of FWS in ag-spec. Second its use was never encountered by the author. But first of all MIME[RFC2045] introduced parameters in ABNF as parameter := attribute "=" value without FWS, and its presence complicates parsers and hinders parser code reuse. The "acdc=" tag is defined without FWS support.

Appendix B. What if this becomes DKIMv3?

Due to an IETF tooling bug iteration -7 of this draft could not become submitted on July 21st of 2025, before the IETF meeting in Madrid. A week later [rt5.ietf.org #44510] is still open, so this gives me time to add this preface with the presumed outcome of the session, introduction of the completely new DKIM2 that tracks in public SMTP envelope data beyond RFC 3461 _"final delivery for the [original] message"_ that is, in mind. This proposal here would thus become "DKIM3" in the event that it is revived if the internet community (in large) objects using DKIM2.

So, shall this draft be revived at some later time, and if compatibility to the original DKIMv1 is not an issue, i would vote to simply go for DKIX- prefix all the time, and use the shorter DKIX-Sig:; i would furtherly vote for using header abbreviations for certain standard headers, as in, for example, D:F:T:L=IU[:.:xy] instead of Date:From:To:List-ID:List-Unsubscribe, where custom names come after standardized abbreviations after a separating field, "." in the above (but care for syntax), otherwise simply an empty field.

If compatibility to DKIMv1 is an issue, then maybe the above is of some value "as-is". _However_, it could be DKIMv1 _completely_, Dave Crocker asked for that (on the list), and had his own "DKOR" draft. This is easy: just move the "acdc=" tag out of DKIM-Signature:, and into a dedicated header, like, say, DKIX-Ctl:, that MUST be covered by the signature, like the -DC: ones. We could, in fact, not use any "sequence", at all: that was my original idea, but on the ML voices said "it is believed reordering occurs", and i, unfortunately, heard these voices; having heard what else there was said on the ML, i would no longer do that, and instead rely on the header stack as such, it will be DKIX-AC: .. and later (groups of) DKIM-Signature:, DKIX-Ctl:, and optionally DKIX-Diff:. In order.

Btw, if compatibility with DKIMv1 is not an issue, knowledge about MIME boundaries could be thought about, and each part be taken and checksummed by itself. This would allow for hops which simply remove

certain parts due to spam or other policies, like stripping HTML parts from multipart/alternative containers. However, this would tend to become patchy and overly complicated, and is possibly a hopeless task all in all.

Greetings, and Ciao! from Germany.

Appendix C. Apologies

By adopting this draft the IETF recognizes that certain developments of the past two decades mutilated the existing email infrastructure to the point where it stopped functioning. Without malicious intent this was the effective outcome of trials to create a stronger security policy. It left operators of for example mailing-lists and alias farms without options to continue operating their infrastructure, except through non-standardized, sometimes partial and imperfect solutions created by third parties. By applying mitigations through a standardized approach it is hereby tried to create a solid foundation for a better future.

Appendix D. Acknowledgements

Thanks to, in the order of appearance, Jesse Thompson, Richard Clayton for arguments against reliance on header field stacks, and pro the numbering scheme, and especially for noticing the partial transaction replay attack problem, Douglas Foster, Michael Thomas for explicit man-in-the-middle replay addressing; Alessandro Vesely inspired the explicitness of the E flag, Bron Gondwana for the inspiration to split up binary differences of headers and body, and Lasse Collin for LZMA2 (and clarification vs XZ). A big fat acknowledgment is due to Murray S. Kucherawy. Special thanks to Klaus Schulze, Manuel Goettsching, both also as Ash Ra Tempel, Laeuten der Seele, Laurent Garnier, as well as the Sleeping Environmental Bot broadcast.

Author’s Address

Steffen Nurpmeso (editor)
Email: steffen@sdaoden.eu