DMARC Working Group
Internet-Draft
Intended status: Experimental
Expires: October 25, 2018

K. Andersen
LinkedIn
B. Long, Ed.
Google
S. Jones, Ed.
TDP
S. Blank, Ed.
Valimail
M. Kucherawy, Ed.
TDP

April 23, 2018

Authenticated Received Chain (ARC) Protocol
draft-ietf-dmarc-arc-protocol-14

Abstract

The Authenticated Received Chain (ARC) protocol creates a mechanism whereby a series of handlers of an email message can conduct authentication of the email message as it passes among them on the way to its destination, and create an attached, authenticated record of the status at each step along the handling path, for use by the final recipient in making choices about the disposition of the message. Changes in the message that might break existing authentication mechanisms can be identified through the ARC Set of header fields.

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 October 25, 2018.

Table of Contents

1. Introduction

Modern email authentication techniques such as the Sender Policy Framework (SPF) [RFC7208] and DomainKeys Identified Mail (DKIM) [RFC6376] have become common. However, their end-to-end utility is limited by the effects of intermediaries along the transmission path, which either are not listed (for SPF) or which break digital signatures (for DKIM). These issues are described in substantial detail in those protocols' defining documents as well as in [RFC6377] and [RFC7960].

Technologies that build upon the use of SPF and DKIM can reduce the success of fraudulent email campaigns. To this end, Domain-based Mail Authentication, Reporting and Conformance (DMARC) [RFC7489], validates the domain of the RFC5322.From header field. However its use along email transmission paths that have independent intermediaries, such as some forwarders and essentially all mailing list services, produces false positive rejections that are problematic, both for the message authors, the intermediary service(s), and for those they are interacting with.

[RFC7960] documented the need for a mechanism which would survive legitimate alteration of a message, in spite of breaking the associated SPF and DKIM information so that the end receiver system(s) can avoid those false positive rejections on delivery. Authenticated Received Chain (ARC) builds upon DKIM mechanisms to provide a sequence of signatures that provide a view of the handling sequence for a message, especially the points where alterations of the content might have occurred. Equipped with this more complete information, the recipient system(s) can make a more informed handling choice, reducing or eliminating the rejections that would occur with the use of DKIM and/or SPF alone.

1.1. General Concepts

ARC provides a "chain of custody" for a message, allowing each entity that handles the message to see what entities handled it before, and to see what the authentication status of the message was at each step in the handling. The handling entity can then put its own entry into the chain of custody and then relay the message to the next handler.

When the message reaches final delivery, the decision to accept and deliver the message, or, alternatively, to reject, discard, or quarantine it, can take the chain of custody into account, applying local policy in addition to policies advertised by the (purported) sending domain. Consider, for example, this scenario:

1. A sender from mysender.example posts a message to a mailing list hosted at listmania.example;

2. The mailing list modifies the message by prepending the list name to the subject line, then sends it to the subscribers;

3. One of the subscribers is alice@mail.service.example, which receives the message from listmania.example.

Assuming the original message was DKIM-signed and mysender.example published an SPF record, the handling by the mailing list will break the DKIM signature because of the message modification, and the forwarding will cause the SPF check to fail in the next step. But listmania.example can add ARC headers to the message to add itself to the document's chain of custody. When mail.service.example sees the message, it can see that SPF and DKIM validation fail, but it can also see that both of these succeeded when they were checked by listmania.example, and can verify listmania's assertion.

As part of its evaluation of the message for delivery, mail.service.example can see that mysender.example publishes a DMARC policy asking that unauthenticated messages be rejected. But is can also see the assertion by listmania.example that the message was correctly authenticated when the message arrived there, and if it accepts that assertion, it can accept the message for further processing, rather than reject it, based on the additional information that ARC has provided.

1.2. Differences Between ARC and DKIM

In DKIM, every participating signing agent attaches a signature that is based on some of the content of the message, local policy, and the domain name of the signing agent's Administrative Management Domain (ADMD). Any verifier can process such a signature; a verified

signature means that the domain referenced in the signature's "d=" parameter has some responsibility for handling the message. An artifact of using digital signature technology for this means that verification also ensures that the portion of the message that was "covered" by the signature has not been altered since the signature was applied. The signatures themselves are generally independent of one another.

In contrast, a validated ARC Set conveys the following pieces of information:

1. An assertion that, at the time that the intermediary ADMD processed the message, the various assertions (such as SPF, DKIM-Signature(s) and/or ARC Chain) already attached to the message by other ADMDs were or were not valid;

2. As with DKIM, an assertion that, for a validated ARC signature, the domain name in the signature takes some responsibility for handling of the message and that the covered content of the message is unchanged since that signature was applied;

3. A further assertion that binds the ARC evaluation results into the ARC Chain sequence.

1.3. Definitions and Terminology

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

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 ([RFC2119][RFC8174]).

Because many of the core concepts and definitions are found in [RFC5598], readers should to be familiar with the contents of [RFC5598], and in particular, the potential roles of intermediaries in the delivery of email.

Syntax descriptions use Augmented BNF (ABNF) [RFC5234].

1.3.1. Terms defined and used in this document

- "ARC-Authentication-Results" (AAR) - an ARC header field described in Section 3.2.

- "ARC-Message-Signature" (AMS) - an ARC header field described in Section 3.3.

- "ARC-Seal" (AS) - an ARC header field described in Section 3.4.

- "ARC Set" - A single group of the header fields introduced in Section 2.1 is called an "ARC Set".

- "ARC Chain" - the complete sequence of ARC Sets for a message. The ARC Chain represents a "chain of custody" for the message, recording its authentication status at each ARC-participating ADMD that handled the message.

1.3.2. Referenced Definitions

The following terms are defined in other RFCs. Those definitions can be found as follows:

- ADMD - [RFC5598], Section 2.3

- MTA - [RFC5598], Section 4.3.2

- MSA - [RFC5598], Section 4.3.1

- MDA - [RFC5598], Section 4.3.3

The three header fields that are part of this specification borrow heavily from existing specifications. Rather than repeating all of the formal definitions that are being reused in ARC, this document only describes and specifies changes in syntax and semantics.

Language, syntax, and other details are imported from DKIM [RFC6376]. Specific references can be found below.

2. Protocol Elements and Features

As with other domain authentication technologies (such as SPF, DKIM, and DMARC), ARC makes no claims about the contents of the email message it has sealed. However, for a valid and passing ARC Chain, a Final Receiver is able to ascertain:

- all (participating) domains that claim responsibility for handling (and possibly modifying) the email message in transit;

- trace information, including:

  * the [RFC7601] Authentication-Results each participating ADMD saw; and

  * additional data needed to compile a DMARC report for the sending domain.

Given this information, each receiver is able to make a more informed local policy decision regarding message processing and, ultimately, delivery to the end user in spite of authentication failure(s) and to inform the message orgination system(s) through the DMARC report(s).

Every participant in an ARC Chain, except for the originating sender and Final Receiver, is both an ARC Validator (when receiving) and then an ARC Sealer (when sending a message onward).

INFORMATIONAL: It is important to understand that validating and then immediately sealing a message leaves no room for message modification, and many early implementations of ARC did not initially work because both operations were performed in a single pass over the message.

The following protocol features are functional parts and design decisions related the protocol that are not specific to either Validators or Sealers, but ensure that the ARC Chain conveys this information to a Final Receiver.

2.1. The "ARC Set" of Header Fields

Each "ARC Set" consists of the following three new header fields:

1. ARC-Authentication-Results (referred to below as "AAR"): virtually identical in syntax to an Authentication-Results field [RFC7601], this field records the results of all message authentication checks done by the recording ADMD at the time the message arrived. Additional information is placed in this field compared to a standard Authentication-Results field in order to support a more complete DMARC report;

2. ARC-Message-Signature (referred to below as "AMS"): virtually identical in syntax to DKIM-Signature, this field contains the signature about the message header and body as they existed at the time of handling by the ADMD adding it (including any modifications made by the sealing ADMD); and

3. ARC-Seal (referred to below as "AS"): highly similar in structure and format to a DKIM-Signature, this field applies a digital signature that protects the integrity of all three of these new fields when they are added by an ADMD, plus all instances of these fields added by prior ADMDs.

An ARC participant always adds all of these header fields before relaying a message to the next handling agent _en route_ to its destination. Moreover, they each have an "instance number" that increases with each ARC Set in the handling chain so that their

original order can be preserved and the three related header fields can be processed as a set.

2.1.1. Instance Tags

ARC includes an indicator in its header fields to show the order in which the header fields comprising an ARC Chain were added, and the specific members of each ARC Set. This is known as the "instance", and the indicator is an "i=" tag/value. That is, the members of the first ARC Set affixed to a message will all include "i=1". This is described in detail in section Section 3.1.

2.2. Chain Validation Status

ARC includes a mechanism which denotes the state of the ARC Chain at each step. The "chain validation status" ("cv" tag/value) is used to communicate the current chain status within the ARC Chain and also through Authentication-Results and ARC-Authentication-Results stamps as well as DMARC reporting.

The chain validation status has one of three possible values:

- none: There was no chain on the message when it arrived for validation; typically occurs when the message arrives at a Message Transfer Agent (MTA) or mediator from a Message Submission Agent (MSA) or when any upstream handlers may not be participating in ARC handling;

- fail: The message has a chain whose validation failed;

- pass: The message has a chain whose validation succeeded.

2.3. Trace Information

ARC includes trace information encoded in the AAR. While section Section 3.2 defines what information must be provided, sealing ADMDs may provide additional information, and validating receivers may use this trace information as they find it useful.

2.4. Key Management

The public keys for ARC header fields follow the same requirements, syntax and semantics as those for DKIM signatures, described in Section 3.6 of [RFC6376]. ARC places no requirements on the selectors and/or domains used for the ARC header field signatures.

2.5. All Failures are Permanent

Because ARC Chains are transmitted across multiple intermediaries, all errors, even temporary ones, become unrecoverable and are considered permanent.

Any error validating or sealing a chain, for whatever reason, MUST result in a "cv=fail" verdict as documented in Section 3.4.2.

2.6. Chain of Custody

At a high level, an ARC Chain represents a chain of custody of authentication and other trace information (AAR) related to a message, signed by each handler of the message. Each link in the chain (AMS) is designed to be brittle, insofar as it survives only until the next modification of the message. However, the sequence of intermediaries in the handling chain (AS) is designed to remain intact over the entirety of the chain.

The ARC Chain can be conceptualized through an analogy with the chain of custody for legal evidence. The material evidence itself is sealed within an tamper-proof bag (AMS) each time. When handed to a new party, that party both vouches for the state of the received evidence container (AAR) and signs for the evidence on a chain of custody report form (AS). As with all analogies, this one should not be taken to interpretive extremes, but primarily used as a conceptual framework.

An ARC Chain that is valid and passing has the attributes listed above in Section 2.

2.7. Optional Participation

Validating an existing chain and then adding your own ARC Set to a message allows you to claim responsibility for handling the message and modifications, if any, done by your ADMD to benefit message delivery downstream. However, no ADMD is obligated to perform these actions.

2.8. Broad Responsibility to Seal

Any mediator ([RFC5598], section 5) that modifies a message may seal its own changes. ARC is not solely intended for perimeter MTAs.

2.9. One Chain to Rule Them All

A message can have only one ARC Chain on it at a time (see Section 3.1). Once broken, the chain cannot be continued, as the chain of custody is no longer valid and responsibility for the message has been lost. For further discussion of this topic and the designed restriction which prevents chain continuation or re-establishment, see [ARC-USAGE].

2.10. Sealing is Always Safe

Even when an ARC Chain is valid and passes, its value is limited to very specific cases. An ARC Chain is specifically designed to provide additional information to a receiver evaluating message delivery in the context of an authentication failure and otherwise be benign. Specifically:

- properly adding an ARC Set to a message does not damage or invalidate an existing chain,
- sealing a chain when you did not modify a message does not negatively affect the chain, and
- validating a message exposes no new threat vectors (see Section 9).

INFORMATIONAL: If an ADMD is unsure whether it will be re-emitting and/or modifying a message, it may elect to seal all inbound mail. For complex or nested ADMD relationships such as found in some hosted mail solutions, this "inbound seal" can be used to facilitate traversal of internal boundaries as well as properly conveying incoming state to any egress MTAs that may need to assert a seal upon exit from the ADMD. Since these internal relationships are highly implementation dependent, this protocol definition can not usefully explore such usage except to note that it is intentionally allowed within the scope of this specification.

3. The ARC Header Fields

3.1. Instance ('i=') Tag

The header fields comprising a single ARC Set are identified by a common "instance" tag value. The instance tag is a string in each header field value that complies with the "tag-spec" ABNF found in Section 3.2 of [RFC6376]. The tag-name is "i" and the value is the text representation of a positive integer, indicating the position in the ARC sequence this set occupies, where the first ARC Set is numbered 1. In ABNF terms:

```
position = 1*2DIGIT ; 1 - 50
instance = [FWS] %x69 [FWS] "=" [FWS] position [FWS] ";"
```

Valid ARC Sets MUST have exactly one instance of each header field (of three) for a given instance value and signing algorithm.

(_INFORMATIONAL:_ Initial development of ARC is only being done with a single allowed signing algorithm, but parallel work in the DCRUP working group [1] is expanding that. For handling multiple signing algorithms, see [ARC-MULTI].)

The 'i' tag value can range from 1-50 (inclusive).

ARC Chains longer than the defined maximum count MUST be marked as failed.

INFORMATIONAL: Because the AMS and AS header field values are made up of tag-spec constructs, the i= tag may be found anywhere within the header field value, but is represented throughout this spec in the initial position for convenience. Implementers are encouraged to place the i= tag at the beginning of the field value to facilitate human inspection of the headers.

3.2. ARC-Authentication-Results (AAR)

The ARC-Authentication-Results header field is syntactically and semantically identical, except for the header field name itself and its instance tag, to an Authentication-Results header field (defined in Section 2.2 of [I-D-7601bis]).

Formally, the header field is specified as follows using ABNF [RFC5234]:

```
arc-info = instance [CFWS] ";" authres-payload
arc-authres-header = "ARC-Authentication-Results:" [CFWS] arc-info
```

The AAR MUST contain all Authentication-Results from within the participating ADMD, regardless of how many Authentication-Results headers are on the message.

3.3. ARC-Message-Signature (AMS)

The ARC-Message-Signature header field is simplified version of a DKIM-Signature header field [RFC6376], with the following modifications:

- There is an "i" tag, as described in Section 3.1.

- There is no "v" tag defined for the AMS header. As required for undefined tags (in [RFC6376]), if seen, it MUST be ignored.

ARC-related header fields (ARC-Seal, ARC-Message-Signture, ARC-Authentication-Results) MUST NOT be included in the content covered by the signature in the signature in this header field.

The AMS SHOULD include any DKIM-Signature header fields already present on the message in the header fields covered by this signature.

Authentication-Results header fields MUST NOT be included since they are likely to be deleted by downstream ADMDs (per Section 5 of [RFC7601]), thereby breaking the AMS signature.

3.4. ARC-Seal (AS)

The ARC-Seal header field is syntactically and semantically similar to a DKIM-Signature field, with the following exceptions:

- There is an "i" tag, as described in Section 3.1.

- The ARC-Seal covers none of the body content of the message. It only covers specific header fields as defined below: Section 3.4.1. No body canonicalization is done.

- Only "relaxed" header canonicalization (Section 3.4.2 of [RFC6376]) is used.

- The only supported tags are "i" (from Section 3.1 of this document), and "a", "b", "d", "s", "t" from Section 3.5 of [RFC6376].

- An additional tag, "cv" is defined in Section 3.4.2

3.4.1. Covered Header Fields

The ARC-Seal signs a specific canonicalized form of the ARC Set header values. The ARC set header values are compiled in increasing instance order, starting at 1, and include the set being added at the time of sealing the message.

Within a set, the header fields are listed in the following order:

1. ARC-Authentication-Results

2. ARC-Message-Signature

3. ARC-Seal

Where the ARC-Seal is the one being generated, it is input to the hash function in its final form except with an empty "b=" value, in the same manner by which a DKIM-Signature signs itself ([RFC6376], section 3.7).

Note that the signing scope for the ARC-Seal is modified in the situation where a chain has failed validation (see Section 5.1).

3.4.2. The 'cv' Tag

A new tag "cv" (chain validation) indicates the outcome of evaluating the existing ARC Chain upon arrival at the ADMD that is adding this header field. The values are defined per Section Section 2.2.

In ABNF terms:

```
chain-status = ("none" / "fail" / "pass")
seal-cv-tag = %x63.76 [FWS] "=" [FWS] chain-status
```

4. Verifier Actions

A verifier takes the following steps to validate the ARC Chain. Canonicalization, hash functions, and signature validation methods are imported from Section 5 of [RFC6376].

1. Collect all ARC Sets currently on the message. If there were none, the ARC state is "none" and the algorithm stops here.

2. Check the morphology of the ARC Chain. If any of these conditions are not met, the chain state is "fail" and the algorithm stops here:

    1. Each ARC Set must be complete (e.g., contains exactly one of each of the three ARC-specific header fields);

    2. The instance values must form a continuous sequence from 1..N with no gaps or repeats;

    3. The cv value for all ARC-Seal(s) must be non-failing:

        1. For i > 1, the value must be "pass";

        2. For i = 1, the value must be "none".

3. For each ARC-Message-Signature from the "N"th instance to the first, validate the AMS:

1. If the "N"th instance (most recent) signature fails, then the chain state is "fail" and the algorithm stops here.

2. If one of the prior AMS signatures fails to validate (for instance "M"), then set the oldest-pass value to the lowest AMS instance number which passed (M+1) and go onto the next step (there is no need to check any other (older) AMS signatures). This does not affect the validity of the chain.

3. If all AMS signatures verify, set the oldest-pass value to zero (0).

4. For each ARC-Seal from the "N"th instance to the first, validate the seal.

    1. If any seal is not valid, the chain state is "fail" and the algorithm stops here.

    2. If all seals pass validation, then the chain state is "pass", and the algorithm is complete.

The end result of the verifier's checks via this algorithm MUST be added into the Authentication-Results header(s) for the ADMD.

INFORMATIONAL: Recipients of an ARC Chain that is invalid or does not pass SHOULD NOT draw negative conclusions without a good understanding of the wider handling context. Until ARC usage is widespread, intermediaries will continue to modify messages without ARC seals.

As with a failing DKIM signature ([RFC6376] Section-6.3), a message with a failing ARC Chain MUST be treated the same as a message with no ARC Chain.

4.1. Authentication-Results Information

Certain information pertinent to ascertaining message disposition can be lost in transit when messages are handled by intermediaries. For example, failing DKIM signatures are sometimes removed by MTAs, and most DKIM signatures on messages modified by intermediaries will fail. Recording the following information in the Authentication-Results stamped as part of the ARC evaluation provides a mechanism for this information to survive transit through a particular ADMD.

Stamped ARC evaluation results is limited to the Chain Validation status (cv) from Section 2.2.

The ptypes and properties defined in this section SHOULD be recorded in the Authentication-Results:

- smtp.client-ip - The connecting client IP address from which the message is received;

- header.oldest-pass - The instance number of the oldest AMS that still validates, or 0 if all pass.

4.2. Handling DNS Problems While Validating ARC

DNS-based failures to verify a chain are treated no differently than any other ARC violation. They result in a "cv=fail" verdict.

4.3. Responding to ARC Validity Violations During the SMTP Transaction

If a receiver determines that the ARC Chain has failed, the receiver MAY signal the breakage through the extended SMTP response code 5.7.7 [RFC3463] "message integrity failure" [ENHANCED-STATUS] and corresponding SMTP response code.

5. Sealer Actions

An ARC sealer MUST take the following actions when presented with a message:

1. Before creating an ARC signature, perform any other, normal authentication and/or signing, so that the ARC signature can cover those results.

2. Build and attach the new ARC Set:

    1. If an ARC Chain exists on the message, then set "N" equal to the highest instance number found on the chain (i=); otherwise set "N" equal to zero for the following steps.

    2. Generate and attach to the message an ARC-Authentication-Results header field as defined in Section Section 3.2, using instance number N+1 and the same content from the previous step.

    3. Generate and attach to the message an ARC-Message-Signature header field as defined in Section 3.3 above, using instance number N+1.

    4. Generate and attach to the message an ARC-Seal header field using the general algorithm described in Section 3.4 above,

the chain validation status as determined in Section 4, and instance number N+1.

5.1. Marking and Sealing "cv=fail" (Invalid) Chains

The header fields signed by the AS header field b= value in the case of a chain failure MUST be only the matching instance headers created by the MTA which detected the malformed chain, as if this newest ARC Set was the only set present.

INFORMATIONAL: In the case of a malformed or otherwise invalid chain there is no way to generate a deterministic set of AS header fields ({#implicit_as_h}) so this approach is mandated.

6. Recording and Reporting the Results of ARC Evaluation

The evaluation of an ARC Chain provides information which will be useful to both the receiver (or intermediary) and to the initial sender of the message. This information should be preserved and reported as follows.

6.1. Information from an ARC Evaluation

The evaluation of an ARC Chain produces a list of domain names for participating intermediaries which handled the message, to wit:

- A list of the "d=" domains found in the validated ARC-Seal header fields

- The "d=" domain found in the most recent (highest instance number) AMS header field (since that is the only one necessarily validated)

In the case of a failed chain, only the terminal ARC Set is covered by the ARC-Seal so the reporting is limited to the findings in that terminal ARC Set.

6.2. Recording (local) ARC Evaluation Results

Receivers who process an attached ARC Chain SHOULD add an "arc=[pass|fail|policy]" method annotation into a locally-affixed Authentication-Results [RFC7601] header field along with any salient comment(s).

Details of the ARC Chain which was evaluated should be included in the Authentication-Results and AAR headers per Section Section 4.1.

6.3. DMARC Reporting of ARC Findings - Interim

Receivers SHOULD indicate situations in which ARC evaluation influenced the results of their local policy determination. DMARC reporting of ARC-informed decisions can be accomplished by adding a local_policy comment explanation containing the list of data discovered in the ARC evaluation, which at a minimum SHOULD include: * the Chain Validation status, * the domain and selector for each AS, * the IP addresses of the mail originating ADMD:

```
<policy_evaluated>
  <disposition>none</disposition>
  <dkim>fail</dkim>
  <spf>fail</spf>
  <reason>
   <type>local_policy</type>
   <comment>arc=pass ams[2].d=d2.example ams[2].s=s1 as[2].d=d2.example
     as[2].s=s2 as[1].d=d1.example as[1].s=s3 client-ip[1]=10.10.10.13</
comment>
  </reason>
</policy_evaluated>
```

In the sample above, d2.example is the sealing domain for ARC[2] and d1.example is the sealing domain for ARC[1].

Intermediary message handlers SHOULD generate DMARC reports on messages which transit their system just like any other message which they receive. This will result in multiple reports for each mediated message as they transit the series of handlers. DMARC report consumers should be aware of this behaviour and make the necessary accommodations.

7. Privacy Considerations

The ARC Chain provides a verifiable record of the handlers for a message. Anonymous remailers will probably not find this compatible with their operating goals.

8. IANA Considerations

[[Note to the RFC Editors: Some of these fields are defined both here and in [I-D-7601bis]. Please delete the overlap from whichever document goes through the publication process after the other.]]

This specification adds three new header fields as defined below.

8.1. Authentication-Results Method Registry Update

This draft adds one item to the IANA "Email Authentication Methods" registry:

- Method : arc
  Defined: [I-D.ARC]
  ptype: header
  Property: chain evaluation result
  Value: chain evaluation result status (see Section 3.4)
  Status: active

8.2. Email Authentication Result Names Registry Update

This draft updates the Email Authentication Results registry, most recently defined in [I-D-7601bis], with one new authentication method and several status codes, all defined by this document:

- Auth Method : arc
  Code: "none", "pass", "fail"
  Specification: [I-D.ARC] Section 3.4.2 Status: active

- Method : spf
  Defined: [I-D.ARC]
  ptype: smtp
  Property: client-ip
  Value: the connecting client IP address from which the message is received
  Status: active

- Method : arc
  Defined: [I-D.ARC]
  ptype: header
  Property: oldest-pass
  Value: the oldest instance with a still validating AMS signature
  Status: active

8.3. Definitions of the ARC header fields

This specification adds three new header fields to the "Permanent Message Header Field Registry", as follows:

- Header field name: ARC-Seal
  Applicable protocol: mail
  Status: draft
  Author/Change controller: IETF
  Specification document(s): [I-D.ARC]
  Related information: [RFC6376]

- Header field name: ARC-Message-Signature
  Applicable protocol: mail
  Status: draft
  Author/Change controller: IETF
  Specification document(s): [I-D.ARC]
  Related information: [RFC6376]

- Header field name: ARC-Authentication-Results
  Applicable protocol: mail
  Status: standard
  Author/Change controller: IETF
  Specification document(s): [I-D.ARC]
  Related information: [RFC7601]

9. Security Considerations

The Security Considerations of [RFC6376] and [RFC7601] apply directly to this specification.

9.1. Header Size

Inclusion of ARC Sets in the header of emails may cause problems for some older or more constrained MTAs if they are unable to accept the greater size of the header.

9.2. DNS Operations

Operators who receive a message bearing N ARC Sets have to complete up to N+1 DNS queries to evaluate the chain (barring DNS redirection mechanisms which can increase the lookups for a given target value). This has at least two effects:

1. An attacker can send a message to an ARC participant with a concocted sequence of ARC Sets bearing the domains of intended victims, and all of them will be queried by the participant until a failure is discovered. The difficulty of forging the signature values should limit the extent of this load to domains under control of the attacker.

2. DKIM only does one DNS check per signature, while this one can do many (per chain). Absent caching, slow DNS responses can cause SMTP timeouts; and backlogged delivery queues on mediating systems. This could be exploited as a DoS attack.

9.3. Message Content Suspicion

Recipients are cautioned to treat messages bearing ARC Sets with the same suspicion that they apply to all other email messages. This includes appropriate content scanning and other checks for potentially malicious content. The handlers which are identified within the ARC Chain may be used to provide input to local policy engines in cases where DMARC validation fails (due to mediation impacting SPF attribution, DKIM validity or alignment).

Note that a passing ARC Chain may not adequately mean that the message is safe because:

1. You have to trust all signatories; and

2. Even trusted systems may have become compromised or may not properly authenticate messages, so even with a chain of trusted participants, the message might still never have authenticated in the first place (which is why you have the AAR to inspect) or could have been subject to unintended modifications.

10. Evaluating the Efficacy of the ARC Protocol (Experimental Considerations)

The ARC protocol is designed to mitigate some of the most common failure conditions for email which transits intermediary handlers en route to the final recipient. Some of these problems have happened due to the adoption of the DMARC protocol [RFC7489] and are listed in [RFC6377] and [RFC7960].

As the ARC protocol becomes standardized and implemented amongst intermediary handlers, the following aspects should be evaluated in order to determine the success of the protocol in accomplishing the intended benefits.

NOTE: Terminology within this section does NOT follow [RFC2119] interpretation. This section represents the current thoughts of the working group regarding unanswered questions related to the protocol. Wider deployment will inform these topics and probably expand them.

10.1. Success Consideration

Currently, many receivers have heuristically determined overrides in order to rescue mail from intermediary-caused failures. Many of those overrides rely on inferrence rather than direct evidence.

ARC will be a success if, for ARC sealed messages, receivers are able to implment ARC-based algorithmic decisions based on the direct

evidence found within the ARC Chain. This is especially relevant for DMARC processing when the DKIM d= value is aligned with the rfc5322.From author domain.

10.2. Failure Considerations

The intent of ARC is to be at most value-add and at worst benign. If ARC opens up significant new vectors for abuse (see Section 9) then this protocol will be a failure. Note that weaknesses inherent in the mail protocols ARC is built upon (such as DKIM replay attacks and other known issues) are not new vectors which can be attributed to this specification.

10.3. Open Questions

The following open questions are academic and have no clear answer at the time of the development of the protocol. However, wide-spread deployment should be able to gather the necessary data to answer some or all of them.

10.3.1. Value of the ARC-Seal (AS) Header

Data should be collected to show if the ARC-Seal (AS) provides value beyond the ARC Message Signature (AMS) for either making delivery decisions or catching malicious actors trying to craft or replay malicious chains.

10.3.2. DNS Overhead

Longer ARC Chains will require more queries to retrieve the keys for validating the chain. While this is not believed to be a security issue (see Section 9.2), it is unclear how much overhead will truly be added. This is similar to some of the initial processing and query load concerns which were debated at the time of the DKIM specification development.

Data should be collected to better understand usable length and distribution of lengths found in valid ARC Chains along with the the DNS impact of processing ARC Chains.

An effective operational maximum will have to be developed through deployment experience in the field.

10.3.3. Distinguishing Valuable from Worthless Trace Information

There are several edge cases where the information in the AAR can make the difference between message delivery or rejection. For example, if there is a well known mailing list that ARC seals but

doesn't do its own initial DMARC enforcement, a Final Receiver with this knowledge could make a delivery decision based upon the authentication information it sees in the corresponding AAR header.

Certain trace information in the AAR is useful/necessary in the construction of DMARC reports. It would be beneficial to identify the value-add of having intermediary-handled mail flow information added into the DMARC reports going back to senders.

Certain receivers believe the entire set of trace information would be valuable to feed into machine learning systems to identify fraud and/or provide other signals related to message delivery.

It is unclear what trace information will be valuable for all receivers, regardless of size.

Data should be collected on what trace information receivers are using that provides useful signals that affect deliverability, and what portions of the trace data are left untouched or provide no useful information.

Since many such systems are intentionally proprietary or confidential to prevent gaming by abusers, it may not be viable to reliably answer this particular question. The evolving nature of attacks can also shift the landscape of "useful" information over time.

11. Implementation Status

[[Note to the RFC Editor: Please remove this section before publication along with the reference to [RFC6982].]]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC6982]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

This information is known to be correct as of the seventh interoperability test event which was held on 2017-07-15 & 16 at IETF99.

For a few of the implementations, later status information was available as of December 2017.

11.1. GMail test reflector and incoming validation

Organization: Google
Description: Internal production implementation with both debug analysis and validating + sealing pass-through function
Status of Operation: Production - Incoming Validation
Coverage: Full spec implemented as of [ARC-DRAFT-06]
Licensing: Proprietary - Internal only
Implementation Notes:

- Full functionality was demonstrated during the interop testing on 2017-07-15.

Contact Info: arc-discuss@dmarc.org [2]

11.2. AOL test reflector and internal tagging

Organization: AOL
Description: Internal prototype implementation with both debug analysis and validating + sealing pass-through function
Status of Operation: Beta
Coverage: ARC Chain validity status checking is operational, but only applied to email addresses enrolled in the test program.
This system conforms to [ARC-DRAFT-06]
Licensing: Proprietary - Internal only
Implementation Notes:

- 2017-07-15: Full functionality verified during the interop testing.

Contact Info: arc-discuss@dmarc.org [3]

11.3. dkimpy

Organization: dkimpy developers/Scott Kitterman
Description: Python DKIM package
Status of Operation: Production
Coverage:

- 2017-07-15: The internal test suite is incomplete, but the command line developmental version of validator was demonstrated to interoperate with the Google and AOL implementations during the interop on 2017-07-15 and the released version passes the tests in [ARC-TEST] arc_test_suite [4] with both python and python3.

Licensing: Open/Other (same as dkimpy package = BCD version 2)
Contact Info: https://launchpad.net/dkimpy

11.4. OpenARC

Organization: TDP/Murray Kucherawy
Description: Implemention of milter functionality related to the OpenDKIM and OpenDMARC packages
Status of Operation: Beta
Coverage: Built to support [ARC-DRAFT-10]
Licensing: Open/Other (same as OpenDKIM and OpenDMARC packages)
Implementation Notes:

- The build is FreeBSD oriented but some packages have been built for easier deployment on RedHat-based Linux platforms.

- Some issues still exist when deploying in a chained milter arrangement (such as OpenSPF -> OpenDKIM -> OpenDMARC -> OpenARC) with coordination between the stages. When deployed in a "sandwich" configuration around an MLM, there is no effective mechanism to convey trust from the ingress (validator) to egress (signer) instances. (_NOTE_: this is expected to resolved with a new release of OpenDMARC expected in January 2018.)

Contact Info: arc-discuss@dmarc.org [5]

11.5. Mailman 3.2 patch

Organization: Mailman development team
Description: Integrated ARC capabilities within the Mailman 3.2 package
Status of Operation: Patch submitted
Coverage: Based on OpenARC
Licensing: Same as mailman package - GPL
Implementation Notes:

- Appears to work properly in at least one beta deployment, but waiting on acceptance of the pull request into the mainline of mailman development

Contact Info: https://www.gnu.org/software/mailman/contact.html

11.6. Copernica/MailerQ web-based validation

Organization: Copernica
Description: Web-based validation of ARC-signed messages
Status of Operation: Beta
Coverage: Built to support [ARC-DRAFT-05]

Licensing: On-line usage only
Implementation Notes:

- Released 2016-10-24

- Requires full message content to be pasted into a web form found at http://arc.mailerq.com/ (warning - https is not supported).

- An additional instance of an ARC signature can be added if one is willing to paste a private key into an unsecured web form.

- 2017-07-15: Testing shows that results match the other implementations listed in this section.

Contact Info: https://www.copernica.com/

11.7. Rspamd

Organization: Rspamd community
Description: ARC signing and verification module
Status of Operation: Production, though deployment usage is unknown
Coverage: Built to support [ARC-DRAFT-06]
Licensing: Open source
Implementation Notes:

- 2017-06-12: Released with version 1.6.0

- 2017-07-15: Testing during the interop showed that the validation functionality interoperated with the Google, AOL, dkimpy and MailerQ implementations

Contact Info: https://rspamd.com/doc/modules/arc.html and https://github.com/vstakhov/rspamd

11.8. PERL MAIL::DKIM module

Organization: FastMail
Description: Email domain authentication (sign and/or verify) module, previously included SPF / DKIM / DMARC, now has ARC added
Status of Operation: Production, deployment usage unknown
Coverage: Built to support [ARC-DRAFT-10]
Licensing: Open Source
Implementation Notes:

- 2017-12-15: v0.50 released with full test set passing for ARC

Contact Info: http://search.cpan.org/~mbradshaw/Mail-DKIM-0.50/

11.9. PERL Mail::Milter::Authentication module

Organization: FastMail
Description: Email domain authentication milter, uses MAIL::DKIM (see above)
Status of Operation: Intial validation completed during IETF99 hackathon with some follow-on work during the week
Coverage: Built to support [I-D.ARC]
Licensing: Open Source
Implementation Notes:

- 2017-07-15: Validation functionality which interoperates with Gmail, AOL, dkimpy was demonstrated; later in the week of IETF99, the signing functionality was reported to be working

- 2017-07-20: ARC functionality has not yet been pushed back to the github repo but should be showing up soon

Contact Info: https://github.com/fastmail/authentication_milter

11.10. Sympa List Manager

Organization: Sympa Dev Community
Description: Work in progress
Status of Operation: Work in progress
Coverage: unknown
Licensing: open source
Implementation Notes:

- 2018-01-05: Tracked as https://github.com/sympa-community/sympa/issues/153

Contact Info: https://github.com/sympa-community

11.11. Oracle Messaging Server

Organization: Oracle
Description:
Status of Operation: Intial development work during IETF99 hackathon. Status since then unknown.
Coverage: Work in progress
Licensing: Unknown
Implementation Notes:

- 2018-03: Protocol handling components are completed, but crypto is not yet functional.

Contact Info: Chris Newman

11.12. MessageSystems Momentum and PowerMTA platforms

Organization: MessageSystems/SparkPost
Description: OpenARC integration into the LUA-enabled Momentum processing space
Status of Operation: Beta
Coverage: Built to support [ARC-DRAFT-10]
Licensing: Unknown
Implementation Notes:

- Initial deployments for validation expected in mid-2018.

Contact Info:

12. References

12.1. Normative References

[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>.

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

[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>.

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

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

[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>.

[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>.

[RFC7601] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", RFC 7601, DOI 10.17487/RFC7601, August 2015, <https://www.rfc-editor.org/info/rfc7601>.

[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>.

12.2. Informative References

[ARC-DRAFT-05]
Andersen, K., "Authenticated Received Chain (ARC) Protocol (I-D-05)", n.d., <https://tools.ietf.org/html/draft-ietf-dmarc-arc-protocol-05>.

[ARC-DRAFT-06]
Andersen, K., "Authenticated Received Chain (ARC) Protocol (I-D-06)", n.d., <https://tools.ietf.org/html/draft-ietf-dmarc-arc-protocol-06>.

[ARC-DRAFT-10]
Andersen, K., "Authenticated Received Chain (ARC) Protocol (I-D-10)", n.d., <https://tools.ietf.org/html/draft-ietf-dmarc-arc-protocol-10>.

[ARC-MULTI]
Andersen, K., "Using Multiple Signing Algorithms with ARC", January 2018, <https://tools.ietf.org/html/draft-ietf-dmarc-arc-multi-01>.

[ARC-TEST]
Blank, S., "ARC Test Suite", January 2017, <https://github.com/Valimail/arc_test_suite>.

[ARC-USAGE]
Jones, S., Adams, T., Rae-Grant, J., and K. Andersen, "Recommended Usage of the ARC Headers", December 2017, <https://tools.ietf.org/html/draft-ietf-dmarc-arc-usage-01>.

[ENHANCED-STATUS]
"IANA SMTP Enhanced Status Codes", n.d.,
<http://www.iana.org/assignments/smtp-enhanced-status-codes/smtp-enhanced-status-codes.xhtml>.

[I-D-7601bis]
Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", February 2018,
<https://datatracker.ietf.org/doc/draft-ietf-dmarc-rfc7601bis/>.

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, DOI 10.17487/RFC6982, July 2013,
<https://www.rfc-editor.org/info/rfc6982>.

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

[RFC7960] Martin, F., Ed., Lear, E., Ed., Draegen. Ed., T., Zwicky, E., Ed., and K. Andersen, Ed., "Interoperability Issues between Domain-based Message Authentication, Reporting, and Conformance (DMARC) and Indirect Email Flows", RFC 7960, DOI 10.17487/RFC7960, September 2016,
<https://www.rfc-editor.org/info/rfc7960>.

12.3. URIs

[1] https://datatracker.ietf.org/wg/dcrup/about/

[2] mailto:arc-discuss@dmarc.org

[3] mailto:arc-discuss@dmarc.org

[4] https://github.com/Valimail/arc_test_suite

[5] mailto:arc-discuss@dmarc.org

[6] https://trac.ietf.org/trac/dmarc/ticket/17

[7] mailto:dmarc@ietf.org

[8] mailto:arc-discuss@dmarc.org

Appendix A. Appendix A - Design Requirements

(This section is re-inserted for background information from [ARC-DRAFT-06] and earlier versions.)

The specification of the ARC framework is driven by the following high-level goals, security considerations, and practical operational requirements.

A.1. Primary Design Criteria

- Provide a verifiable "chain of custody" for email messages;
- Not require changes for originators of email;
- Support the verification of the ARC header field set by each hop in the handling chain;
- Work at Internet scale; and
- Provide a trustable mechanism for the communication of Authentication-Results across trust boundaries.

A.2. Out of Scope

ARC is not a trust framework. Users of the ARC header fields are cautioned against making unsubstantiated conclusions when encountering a "broken" ARC sequence.

Appendix B. Appendix B - Example Usage

[[Note: The following examples were mocked up early in the definition process for the spec. They no longer reflect the current definition and need various updates which will be included in a future draft. Issue 17 [6]]]

(Obsolete but retained for illustrative purposes)

B.1. Example 1: Simple mailing list

B.1.1. Here's the message as it exits the Origin:

```
Return-Path: <jqd@d1.example>
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
    (authenticated bits=0)
    by segv.d1.example with ESMTP id t0FN4a8O084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
    (envelope-from jqd@d1.example)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=d1.example;
    s=20130426; t=1421363082;
    bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
    h=Message-ID:Date:From:MIME-Version:To:CC:Subject:Content-Type:
     Content-Transfer-Encoding;
    b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijrvQw
     bv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD4Gd3TRJl
     gotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
Message-ID: <54B84785.1060301@d1.example>
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe <jqd@d1.example>
To: arc@dmarc.org
Subject: Example 1

Hey gang,
This is a test message.
--J.
```

B.1.2. Message is then received at example.org

B.1.2.1. Example 1, Step A: Message forwarded to list members

Processing at example.org:

- example.org performs authentication checks
- No previous Authentication-Results or ARC-Seal headers are present
- example.org adds ARC-Authentication-Results header
- example.org adds Received: header
- example.org adds a ARC-Seal header

Here's the message as it exits example.org:

```
Return-Path: <jqd@d1.example>
ARC-Seal: i=1; a=rsa-sha256; t=1421363107;
    s=seal2015; d=example.org; cv=none;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61
     TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69
     EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed;
    d=example.org; s=clochette; t=1421363105;
    bh=FjQYm3HhXStuzauzV4Uc02o55EzATNfL4uBvEoy7k3s=;
    h=List-Id:List-Unsubscribe:List-Archive:List-Post:
     List-Help:List-Subscribe:Reply-To:DKIM-Signature;
    b=Wb4EiVANwAX8obWwrRWpmlhxmdIvj0dv0psIkiaGOOug32iTAcc74/iWvlPXpF1F5
     vYVF0mw5cmKOa824tKkUOOE3yinTAekqnly7GJuFCDeSA1fQHhStVV7BzAr3A+m4bw
     a6RIDgr3rOPJil678dZTHfztFWyjwIUxB5Ajxj/M=
Received: from segv.d1.example (segv.d1.example [72.52.75.15])
    by lists.example.org (8.14.5/8.14.5) with ESMTP id t0EKaNU9010123
    for <arc@example.org>; Thu, 14 Jan 2015 15:01:30 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Authentication-Results: i=1; lists.example.org;
    spf=pass smtp.mfrom=jqd@d1.example;
    dkim=pass (1024-bit key) header.i=@d1.example;
    dmarc=pass
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
    (authenticated bits=0)
    by segv.d1.example with ESMTP id t0FN4a8O084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
    (envelope-from jqd@d1.example)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=d1.example;
    s=20130426; t=1421363082;
    bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
    h=Message-ID:Date:From:MIME-Version:To:CC:Subject:Content-Type:
     Content-Transfer-Encoding;
    b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijr
     vQwbv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD4G
     d3TRJlgotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
Message-ID: <54B84785.1060301@d1.example>
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe <jqd@d1.example>
To: arc@example.org
Subject: [Lists] Example 1

Hey gang,
This is a test message.
--J.
```

B.1.3. Example 1: Message received by Recipient

Let's say that the Recipient is example.com

Processing at example.com:

- example.com performs usual authentication checks
- example.com adds Authentication-Results: header, Received header
- Determines that message fails DMARC
- Checks for ARC-Seal: header; finds one
- Validates the signature in the ARC-Seal: header, which covers the ARC-Authentication-Results: header
- example.com can use the ARC-Authentication-Results values or verify the DKIM-Signature from lists.example.org

Here's what the message looks like at this point:

```
Return-Path: <jqd@d1.example>
Received: from example.org (example.org [208.69.40.157])
    by clothilde.example.com with ESMTP id
    d200mr22663000ykb.93.1421363207
    for <fmartin@example.com>; Thu, 14 Jan 2015 15:02:40 -0800 (PST)
Authentication-Results: clothilde.example.com; spf=fail
    smtp.from=jqd@d1.example; dkim=pass (1024-bit key)
    header.i=@example.org; dmarc=fail; arc=pass
ARC-Seal: i=1; a=rsa-sha256; t=1421363107;
    s=seal2015; d=example.org; cv=none;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61
     TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69
     EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed;
    d=example.org; s=clochette; t=1421363105;
    bh=FjQYm3HhXStuzauzV4Uc02o55EzATNfL4uBvEoy7k3s=;
    h=List-Id:List-Unsubscribe:List-Archive:List-Post:
     List-Help:List-Subscribe:Reply-To:DKIM-Signature;
    b=Wb4EiVANwAX8obWwrRWpmlhxmdIvj0dv0psIkiaGOOug32iTAcc74/iWvlPXpF
     1F5vYVF0mw5cmKOa824tKkUOOE3yinTAekqnly7GJuFCDeSA1fQHhStVV7BzAr3
     A+m4bwa6RIDgr3rOPJil678dZTHfztFWyjwIUxB5Ajxj/M=
Received: from segv.d1.example (segv.d1.example [72.52.75.15])
    by lists.example.org (8.14.5/8.14.5) with ESMTP id t0EKaNU9010123
    for <arc@example.org>; Thu, 14 Jan 2015 15:01:30 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Authentication-Results: i=1; lists.example.org;
```

```
    spf=pass smtp.mfrom=jqd@d1.example;
    dkim=pass (1024-bit key) header.i=@d1.example;
    dmarc=pass
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
    (authenticated bits=0)
    by segv.d1.example with ESMTP id t0FN4a8O084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
    (envelope-from jqd@d1.example)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=d1.example;
    s=20130426; t=1421363082;
    bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
    h=Message-ID:Date:From:MIME-Version:To:CC:Subject:Content-Type:
     Content-Transfer-Encoding;
    b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijrvQw
     bv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD4Gd3TRJl
     gotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
Message-ID: <54B84785.1060301@d1.example>
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe <jqd@d1.example>
To: arc@example.org
Subject: [Lists] Example 1

Hey gang,
This is a test message.
--J.
```

B.2. Example 2: Mailing list to forwarded mailbox

B.2.1. Here's the message as it exits the Origin:

```
Return-Path: <jqd@d1.example>
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
    (authenticated bits=0)
    by segv.d1.example with ESMTP id t0FN4a8O084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
    (envelope-from jqd@d1.example)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=d1.example;
    s=20130426; t=1421363082;
    bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
    h=Message-ID:Date:From:MIME-Version:To:CC:Subject:Content-Type:
     Content-Transfer-Encoding;
    b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijrvQw
     bv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD4Gd3TRJl
     gotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
Message-ID: <54B84785.1060301@d1.example>
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe <jqd@d1.example>
To: arc@example.org
Subject: Example 1

Hey gang,
This is a test message.
--J.
```

B.2.2. Message is then received at example.org

B.2.2.1. Example 2, Step A: Message forwarded to list members

Processing at example.org:

- example.org performs authentication checks
- example.org applies standard DKIM signature
- No previous Authentication-Results or ARC-Seal headers are present
- example.org adds ARC-Authentication-Results header
- example.org adds usual Received: header
- example.org adds a ARC-Seal header

Here's the message as it exits Step A:

```
Return-Path: <jqd@d1.example>
ARC-Seal: i=1; a=rsa-sha256; t=1421363107;
    s=seal2015; d=example.org; cv=none;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz6
     1TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L
     69EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed;
    d=example.org; s=clochette; t=1421363105;
    bh=FjQYm3HhXStuzauzV4Uc02o55EzATNfL4uBvEoy7k3s=;
    h=List-Id:List-Unsubscribe:List-Archive:List-Post:
     List-Help:List-Subscribe:Reply-To:DKIM-Signature;
    b=Wb4EiVANwAX8obWwrRWpmlhxmdIvj0dv0psIkiaGOOug32iTAcc74/iWvlPXpF
     1F5vYVF0mw5cmKOa824tKkUOOE3yinTAekqnly7GJuFCDeSA1fQHhStVV7BzAr3
     A+m4bwa6RIDgr3rOPJil678dZTHfztFWyjwIUxB5Ajxj/M=
Received: from segv.d1.example (segv.d1.example [72.52.75.15])
    by lists.example.org (8.14.5/8.14.5) with ESMTP id t0EKaNU9010123
    for <arc@example.org>; Thu, 14 Jan 2015 15:01:30 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Authentication-Results: i=1; lists.example.org;
    spf=pass smtp.mfrom=jqd@d1.example;
    dkim=pass (1024-bit key) header.i=@d1.example;
    dmarc=pass
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
    (authenticated bits=0)
    by segv.d1.example with ESMTP id t0FN4a8O084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
    (envelope-from jqd@d1.example)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=d1.example;
    s=20130426; t=1421363082;
    bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
    h=Message-ID:Date:From:MIME-Version:To:CC:Subject:Content-Type:
     Content-Transfer-Encoding;
    b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijr
     vQwbv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD4G
     d3TRJlgotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
Message-ID: <54B84785.1060301@d1.example>
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe <jqd@d1.example>
To: arc@example.org
Subject: [Lists] Example 1

Hey gang,
This is a test message.
--J.
```

B.2.2.2. Example 2, Step B: Message from list forwarded

The message is delivered to a mailbox at gmail.com
Processing at gmail.com:

- gmail.com performs usual authentication checks
- gmail.com adds Authentication-Results: and Received: header
- Determines that message fails DMARC
- Checks for ARC-Seal: header; finds one
- Validates the signature in the ARC-Seal: header, which covers the ARC-Authentication-Results: header
- Uses the ARC-Authentication-Results: values, but:
- Instead of delivering message, prepares to forward message per user settings
- Applies usual DKIM signature
- gmail.com adds it's own ARC-Seal: header, contents of which are
  - version
  - sequence number ("i=2")
  - hash algorithm (SHA256 as example)
  - timestamp ("t=")
  - selector for key ("s=notary01")
  - domain for key ("d=gmail.com")
  - headers included in hash ("h=ARC-Authentication-Results:ARC-Seal")
  - Note: algorithm requires only ARC-Seals with lower sequence # be included, in ascending order
  - signature of the header hash

Here's what the message looks like at this point:

Return-Path: <jqd@d1.example>

```
ARC-Seal: i=2; a=rsa-sha256; t=1421363253;
    s=notary01; d=gmail.com; cv=pass;
    b=sjHDMriRZ0Mui5eVEOGscRHWbQHcy97lvrduHQ8h+f2CfIrxUiKOE44x3LQwDWR
     YbDjf5fcM9MdcIahC+cP59BQ9Y9DHwMDzwRTnM7NVb4kY+tSaVnLoIOaP9lF/sut
     txO+RRNr0fCFw==
ARC-Message-Signature: i=2; a=rsa-sha256; c=relaxed/relaxed;
    d=gmail.com; s=20120806;
    h=mime-version:content-type:x-original-sender:
     x-original-authentication-results:precedence:mailing-list:
     list-id:list-post:list-help:list-archive:sender:reply-to:
     list-unsubscribe:DKIM-Signature;
    bh=2+gZwZhUK2V7JbpoO2MTrU19WvhcA4JnjiohFm9ZZ/g=;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61
     TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69
     EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0Ab8Oi1ebYV/hIBmfhS
     LF1E80hMPcMijONfTQB6g5Hoh/kE6N2fgp6aSngL/WA3+g3Id8ElhXHvIGcJRFeM
     KdJqiW5cxdqPTRW+BnR5ee6Tzg06kr265NTDIAU8p8fQNuLfZj49MMA+QwDBJtXw
     bQoZyRtb6X6q0mYaszUB8kw==
Received: by mail-yk0-f179.google.com with SMTP id 19so2728865ykq.10
    for <mailbox@gmail.com>; Thu, 14 Jan 2015 15:02:45 -0800 (PST)
Authentication-Results: i=2; gmail.com; spf=fail
    smtp.from=jqd@d1.example; dkim=pass (1024-bit key)
    header.i=@example.org; dmarc=fail; arc=pass
ARC-Seal: i=1; a=rsa-sha256; t=1421363107;
    s=seal2015; d=example.org; cv=none:
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61
     TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69
     EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed;
    d=example.org; s=clochette; t=1421363105;
    bh=FjQYm3HhXStuzauzV4Uc02o55EzATNfL4uBvEoy7k3s=;
    h=List-Id:List-Unsubscribe:List-Archive:List-Post:
     List-Help:List-Subscribe:Reply-To:DKIM-Signature;
    b=Wb4EiVANwAX8obWwrRWpmlhxmdIvj0dv0psIkiaGOOug32iTAcc74/iWvlPXpF
     1F5vYVF0mw5cmKOa824tKkUOOE3yinTAekqnly7GJuFCDeSA1fQHhStVV7BzAr3
     A+m4bwa6RIDgr3rOPJil678dZTHfztFWyjwIUxB5Ajxj/M=
Received: from segv.d1.example (segv.d1.example [72.52.75.15])
    by lists.example.org (8.14.5/8.14.5) with ESMTP id t0EKaNU9010123
    for <arc@example.org>; Thu, 14 Jan 2015 15:01:30 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Authentication-Results: i=1; lists.example.org;
    spf=pass smtp.mfrom=jqd@d1.example;
    dkim=pass (1024-bit key) header.i=@d1.example;
    dmarc=pass
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
    (authenticated bits=0)
    by segv.d1.example with ESMTP id t0FN4a8O084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
```

```
    (envelope-from jqd@d1.example)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=d1.example;
    s=20130426; t=1421363082;
    bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
    h=Message-ID:Date:From:MIME-Version:To:CC:Subject:Content-Type:
     Content-Transfer-Encoding;
    b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijr
     vQwbv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD4G
     d3TRJlgotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
Message-ID: <54B84785.1060301@d1.example>
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe <jqd@d1.example>
To: arc@example.org
Subject: [Lists] Example 1

Hey gang,
This is a test message.
--J.
```

B.2.3. Example 2: Message received by Recipient

Let's say that the Recipient is example.com
Processing at example.com:

- example.com performs usual authentication checks
- example.com adds Authentication-Results: header, Received header
- Determines that message fails DMARC
- Checks for ARC-Seal: header; finds two
- Validates the signature in the highest numbered ("i=2") ARC-Seal: header, which covers all previous ARC-Seal: and ARC-Authentication-Results: headers
- Validates the other ARC-Seal header ("i=1"), which covers the ARC-Authentication-Results: header
- example.com uses the ARC-Authentication-Results: values

Here's what the message looks like at this point:

```
Return-Path: <jqd@d1.example>
Received: from mail-ob0-f188.google.com (mail-ob0-f188.google.com
    [208.69.40.157]) by clothilde.example.com with ESMTP id
    d200mr22663000ykb.93.1421363268
    for <fmartin@example.com>; Thu, 14 Jan 2015 15:03:15 -0800 (PST)
```

```
Authentication-Results: clothilde.example.com; spf=fail
    smtp.from=jqd@d1.example; dkim=pass (1024-bit key)
    header.i=@gmail.com; dmarc=fail; arc=pass
ARC-Seal: i=2; a=rsa-sha256; t=1421363253;
    s=notary01; d=gmail.com; cv=pass;
    b=sjHDMriRZ0Mui5eVEOGscRHWbQHcy97lvrduHQ8h+f2CfIrxUiKOE44x3LQwDWR
     YbDjf5fcM9MdcIahC+cP59BQ9Y9DHwMDzwRTnM7NVb4kY+tSaVnLoIOaP9lF/sut
     txO+RRNr0fCFw==
ARC-Message-Signature: i=2; a=rsa-sha256; c=relaxed/relaxed;
    d=gmail.com; s=20120806;
    h=mime-version:content-type:x-original-sender:
     x-original-authentication-results:precedence:mailing-list:
     list-id:list-post:list-help:list-archive:sender:reply-to:
     :list-unsubscribe:DKIM-Signature;
    bh=2+gZwZhUK2V7JbpoO2MTrU19WvhcA4JnjiohFm9ZZ/g=;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61
     TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69
     EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0Ab8Oi1ebYV/hIBmfhS
     LF1E80hMPcMijONfTQB6g5Hoh/kE6N2fgp6aSngL/WA3+g3Id8ElhXHvIGcJRFeM
     KdJqiW5cxdqPTRW+BnR5ee6Tzg06kr265NTDIAU8p8fQNuLfZj49MMA+QwDBJtXw
     bQoZyRtb6X6q0mYaszUB8kw==
Received: by mail-yk0-f179.google.com with SMTP id 19so2728865ykq.10
    for <mailbox@gmail.com>; Thu, 14 Jan 2015 15:02:45 -0800 (PST)
Authentication-Results: i=2; gmail.com; spf=fail
    smtp.from=jqd@d1.example; dkim=pass (1024-bit key)
    header.i=@example.org; dmarc=fail; arc=pass
ARC-Seal: i=1; a=rsa-sha256; t=1421363107;
    s=seal2015; d=example.org; cv=none;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61
     TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69
     EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed;
    d=example.org; s=clochette; t=1421363105;
    bh=FjQYm3HhXStuzauzV4Uc02o55EzATNfL4uBvEoy7k3s=;
    h=List-Id:List-Unsubscribe:List-Archive:List-Post:
     List-Help:List-Subscribe:Reply-To:DKIM-Signature;
    b=Wb4EiVANwAX8obWwrRWpmlhxmdIvj0dv0psIkiaGOOug32iTAcc74/iWvlPXpF
     1F5vYVF0mw5cmKOa824tKkUOOE3yinTAekqnly7GJuFCDeSA1fQHhStVV7BzAr3
     A+m4bwa6RIDgr3rOPJil678dZTHfztFWyjwIUxB5Ajxj/M=
Received: from segv.d1.example (segv.d1.example [72.52.75.15])
    by lists.example.org (8.14.5/8.14.5) with ESMTP id t0EKaNU9010123
    for <arc@example.org>; Thu, 14 Jan 2015 15:01:30 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Authentication-Results: i=1; lists.example.org;
    spf=pass smtp.mfrom=jqd@d1.example;
    dkim=pass (1024-bit key) header.i=@d1.example;
    dmarc=pass
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
```

```
      (authenticated bits=0)
      by segv.d1.example with ESMTP id t0FN4a8O084569;
      Thu, 14 Jan 2015 15:00:01 -0800 (PST)
      (envelope-from jqd@d1.example)
   DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=d1.example;
      s=20130426; t=1421363082;
      bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
      h=Message-ID:Date:From:MIME-Version:To:CC:Subject:Content-Type:
       Content-Transfer-Encoding;
      b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijr
       vQwbv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD4G
       d3TRJlgotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
   Message-ID: <54B84785.1060301@d1.example>
   Date: Thu, 14 Jan 2015 15:00:01 -0800
   From: John Q Doe <jqd@d1.example>
   To: arc@example.org
   Subject: [Lists] Example 1

   Hey gang,
   This is a test message.
   --J.
```

B.3. Example 3: Mailing list to forwarded mailbox with source

B.3.1. Here's the message as it exits the Origin:

```
Return-Path: <jqd@d1.example>
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
    (authenticated bits=0)
    by segv.d1.example with ESMTP id t0FN4a8O084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Seal: i=1; a=rsa-sha256; t=1421363107;
    s=origin2015; d=d1.example; cv=none;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61T
     X6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69EU
     8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed;
    d=d1.example; s=20130426; t=1421363082;
    bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
    h=MIME-Version:CC:Content-Type:Content-Transfer-Encoding;
    b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijrv
     Qwbv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD4Gd3
     TRJlgotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
Message-ID: <54B84785.1060301@d1.example>
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe <jqd@d1.example>
To: arc@example.org
Subject: Example 1

Hey gang,
This is a test message.
--J.
```

B.3.2. Message is then received at example.org

B.3.2.1. Example 3, Step A: Message forwarded to list members with source

Processing at example.org:

- example.org performs authentication checks
- example.org applies standard DKIM signature
- Checks for ARC-Seal: header; finds one (i=1)
- Validates the signature in the ARC-Seal (i=1): header, which covers the d1.example ARC-Message-Signature: header
- example.org adds ARC-Authentication-Results header
- example.org adds usual Received: header

- example.org adds a DKIM-Signature
- example.org adds a ARC-Seal header, contents of which are
  - sequence number ("i=2")
  - hash algorithm (SHA256 as example)
  - timestamp ("t=")
  - chain validity ("cv=")
  - selector for key ("s=seal2015")
  - domain for key ("d=example.org")
  - signature ("b=")

Here's the message as it exits Step A:

```
Return-Path: <jqd@d1.example>
ARC-Seal: i=2; a=rsa-sha256; t=1421363107;
    s=seal2015; d=example.org; cv=pass;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz6
     1TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L
     69EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=2; a=rsa-sha256; c=relaxed/relaxed;
    d=example.org; s=clochette; t=1421363105;
    bh=FjQYm3HhXStuzauzV4Uc02o55EzATNfL4uBvEoy7k3s=;
    h=List-Id:List-Unsubscribe:List-Archive:List-Post:
     List-Help:List-Subscribe:From:Reply-To:DKIM-Signature;
    b=Wb4EiVANwAX8obWwrRWpmlhxmdIvj0dv0psIkiaGOOug32iTAcc74/iWvlPXpF
     1F5vYVF0mw5cmKOa824tKkUOOE3yinTAekqnly7GJuFCDeSA1fQHhStVV7BzAr3
     A+m4bwa6RIDgr3rOPJil678dZTHfztFWyjwIUxB5Ajxj/M=
Received: from segv.d1.example (segv.d1.example [72.52.75.15])
    by lists.example.org (8.14.5/8.14.5) with ESMTP id t0EKaNU9010123
    for <arc@example.org>; Thu, 14 Jan 2015 15:01:30 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Authentication-Results: i=2; lists.example.org;
    spf=pass smtp.mfrom=jqd@d1.example;
    dkim=pass (1024-bit key) header.i=@d1.example;
    dmarc=pass
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
    (authenticated bits=0)
    by segv.d1.example with ESMTP id t0FN4a8O084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Seal: i=1; a=rsa-sha256; t=1421363107;
    s=origin2015; d=d1.example; cv=none;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61
     TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69
     EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed;
    d=d1.example; s=20130426; t=1421363082;
    bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
    h=MIME-Version:CC:Content-Type:Content-Transfer-Encoding;
    b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijr
     vQwbv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD4G
     d3TRJlgotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
Message-ID: <54B84785.1060301@d1.example>
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe <jqd@d1.example>
To: arc@example.org
Subject: [Lists] Example 1

Hey gang,
This is a test message.
--J.
```

B.3.2.2. Example 3, Step B: Message from list forwarded with source

The message is delivered to a mailbox at gmail.com
Processing at gmail.com:

- gmail.com performs usual authentication checks
- gmail.com adds Authentication-Results: and Received: header
- Determines that message fails DMARC
- Checks for ARC-Seal: header; finds two
- Validates the signature in the ARC-Seal (i=2): header, which covers the ARC-Authentication-Results: header
- Validates the signature in the ARC-Seal (i=1): header, which covers the d1.example ARC-Message-Signature: header
- Uses the ARC-Authentication-Results: values, but:
- Instead of delivering message, prepares to forward message per user settings
- Applies usual DKIM signature
- gmail.com adds it's own ARC-Seal: header, contents of which are
  - version
  - sequence number ("i=2")
  - hash algorithm (SHA256 as example)
  - timestamp ("t=")
  - selector for key ("s=notary01")
  - domain for key ("d=gmail.com")
  - Note: algorithm requires only ARC-Seals with lower sequence # be included, in ascending order
  - signature of the chain

Here's what the message looks like at this point:

Return-Path: <jqd@d1.example>

```
ARC-Seal: i=3; a=rsa-sha256; t=1421363253;
    s=notary01; d=gmail.com; cv=pass;
    b=sjHDMriRZ0Mui5eVEOGscRHWbQHcy97lvrduHQ8h+f2CfIrxUiKOE44x3LQwD
     WRYbDjf5fcM9MdcIahC+cP59BQ9Y9DHwMDzwRTnM7NVb4kY+tSaVnLoIOaP9lF
     /suttxO+RRNr0fCFw==
ARC-Message-Signature: i=3; a=rsa-sha256; c=relaxed/relaxed;
    d=gmail.com; s=20120806;
    h=mime-version:content-type:x-original-sender
     :x-original-authentication-results:precedence:mailing-list
     :list-id:list-post:list-help:list-archive:sender
     :list-unsubscribe:reply-to;
    bh=2+gZwZhUK2V7JbpoO2MTrU19WvhcA4JnjiohFm9ZZ/g=;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz6
     1TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L
     69EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0Ab8Oi1ebYV/hIBm
     fhSLF1E80hMPcMijONfTQB6g5Hoh/kE6N2fgp6aSngL/WA3+g3Id8ElhXHvIGcJ
     RFeMKdJqiW5cxdqPTRW+BnR5ee6Tzg06kr265NTDIAU8p8fQNuLfZj49MMA+QwD
     BJtXwbQoZyRtb6X6q0mYaszUB8kw==
Received: by mail-yk0-f179.google.com with SMTP id 19so2728865ykq.10
    for <mailbox@gmail.com>; Thu, 14 Jan 2015 15:02:45 -0800 (PST)
Authentication-Results: i=3; gmail.com; spf=fail
    smtp.from=jqd@d1.example; dkim=pass (1024-bit key)
    header.i=@example.org; dmarc=fail; arc=pass
ARC-Seal: i=2; a=rsa-sha256; t=1421363107;
    s=seal2015; d=example.org; cv=pass;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61
     TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69
     EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=2; a=rsa-sha256; c=relaxed/relaxed;
    d=example.org; s=clochette; t=1421363105;
    bh=FjQYm3HhXStuzauzV4Uc02o55EzATNfL4uBvEoy7k3s=;
    h=List-Id:List-Unsubscribe:List-Archive:List-Post:
     List-Help:List-Subscribe:Reply-To:DKIM-Signature;
    b=Wb4EiVANwAX8obWwrRWpmlhxmdIvj0dv0psIkiaGOOug32iTAcc74/iWvlPXpF1
     F5vYVF0mw5cmKOa824tKkUOOE3yinTAekqnly7GJuFCDeSA1fQHhStVV7BzAr3A+
     m4bwa6RIDgr3rOPJil678dZTHfztFWyjwIUxB5Ajxj/M=
Received: from segv.d1.example (segv.d1.example [72.52.75.15])
    by lists.example.org (8.14.5/8.14.5) with ESMTP id t0EKaNU9010123
    for <arc@example.org>; Thu, 14 Jan 2015 15:01:30 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Authentication-Results: i=2; lists.example.org;
    spf=pass smtp.mfrom=jqd@d1.example;
    dkim=pass (1024-bit key) header.i=@d1.example;
    dmarc=pass
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
    (authenticated bits=0)
    by segv.d1.example with ESMTP id t0FN4a8O084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
```

```
      (envelope-from jqd@d1.example)
   ARC-Seal: i=1; a=rsa-sha256; t=1421363107;
       s=origin2015; d=d1.example; cv=none;
       b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61
        TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69
        EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
   ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed;
       d=d1.example; s=20130426; t=1421363082;
       bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
       h=MIME-Version:CC:Content-Type:Content-Transfer-Encoding;
       b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYij
        rvQwbv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD
        4Gd3TRJlgotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
   Message-ID: <54B84785.1060301@d1.example>
   Date: Thu, 14 Jan 2015 15:00:01 -0800
   From: John Q Doe <jqd@d1.example>
   To: arc@example.org
   Subject: [Lists] Example 1

   Hey gang,
   This is a test message.
   --J.
```

B.3.3. Example 3: Message received by Recipient

Let's say that the Recipient is example.com
Processing at example.com:

- example.com performs usual authentication checks
- example.com adds Authentication-Results: header, Received header
- Determines that message fails DMARC
- Checks for ARC-Seal: header; finds three
- Validates the signature in the highest numbered ("i=2") ARC-Seal: header, which covers all previous ARC-Seal: and ARC-Authentication-Results: headers
- Validates the other ARC-Seal header ("i=2"), which covers the ARC-Authentication-Results: header
- Validates the other ARC-Seal header ("i=1"), which covers the d1.example ARC-Message-Signature: header
- example.com uses the ARC-Authentication-Results: values

Here's what the message looks like at this point:

```
Return-Path: <jqd@d1.example>
Received: from mail-ob0-f188.google.com (mail-ob0-f188.google.com
    [208.69.40.157]) by clothilde.example.com with ESMTP id
    d200mr22663000ykb.93.1421363268
    for <fmartin@example.com>; Thu, 14 Jan 2015 15:03:15 -0800 (PST)
Authentication-Results: clothilde.example.com; spf=fail
    smtp.from=jqd@d1.example; dkim=pass (1024-bit key)
    header.i=@gmail.com; dmarc=fail; arc=pass
ARC-Seal: i=3; a=rsa-sha256; t=1421363253;
    s=notary01; d=gmail.com; cv=pass;
    b=sjHDMriRZ0Mui5eVEOGscRHWbQHcy97lvrduHQ8h+f2CfIrxUiKOE44x3LQwDW
     RYbDjf5fcM9MdcIahC+cP59BQ9Y9DHwMDzwRTnM7NVb4kY+tSaVnLoIOaP9lF/s
     uttxO+RRNr0fCFw==
ARC-Message-Signature: i=3; a=rsa-sha256; c=relaxed/relaxed;
    d=gmail.com; s=20120806;
    h=mime-version:content-type:x-original-sender
     :x-original-authentication-results:precedence
     :mailing-list:list-id:list-post:list-help:list-archive:sender
     :list-unsubscribe:reply-to;
    bh=2+gZwZhUK2V7JbpoO2MTrU19WvhcA4JnjiohFm9ZZ/g=;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz6
     1TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L
     69EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0Ab8Oi1ebYV/hIBm
     fhSLF1E80hMPcMijONfTQB6g5Hoh/kE6N2fgp6aSngL/WA3+g3Id8ElhXHvIGcJ
     RFeMKdJqiW5cxdqPTRW+BnR5ee6Tzg06kr265NTDIAU8p8fQNuLfZj49MMA+QwD
     BJtXwbQoZyRtb6X6q0mYaszUB8kw==
Received: by mail-yk0-f179.google.com with SMTP id 19so2728865ykq.10
    for <mailbox@gmail.com>; Thu, 14 Jan 2015 15:02:45 -0800 (PST)
Authentication-Results: i=3; gmail.com; spf=fail
    smtp.from=jqd@d1.example; dkim=pass (1024-bit key)
    header.i=@example.org; dmarc=fail; arc=pass
ARC-Seal: i=2; a=rsa-sha256; t=1421363107;
    s=seal2015; d=example.org; cv=pass;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz6
     1TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L
     69EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=2; a=rsa-sha256; c=relaxed/relaxed;
    d=example.org; s=clochette; t=1421363105;
    bh=FjQYm3HhXStuzauzV4Uc02o55EzATNfL4uBvEoy7k3s=;
    h=List-Id:List-Unsubscribe:List-Archive:List-Post:
     List-Help:List-Subscribe:Reply-To:DKIM-Signature;
    b=Wb4EiVANwAX8obWwrRWpmlhxmdIvj0dv0psIkiaGOOug32iTAcc74/iWvlPXpF1
     F5vYVF0mw5cmKOa824tKkUOOE3yinTAekqnly7GJuFCDeSA1fQHhStVV7BzAr3A+
     m4bwa6RIDgr3rOPJil678dZTHfztFWyjwIUxB5Ajxj/M=
Received: from segv.d1.example (segv.d1.example [72.52.75.15])
    by lists.example.org (8.14.5/8.14.5) with ESMTP id t0EKaNU9010123
```

```
    for <arc@example.org>; Thu, 14 Jan 2015 15:01:30 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Authentication-Results: i=2; lists.example.org;
    spf=pass smtp.mfrom=jqd@d1.example;
    dkim=pass (1024-bit key) header.i=@d1.example;
    dmarc=pass
Received: from [10.10.10.131] (w-x-y-z.dsl.static.isp.com [w.x.y.z])
    (authenticated bits=0)
    by segv.d1.example with ESMTP id t0FN4a8O084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Seal: i=1; a=rsa-sha256; t=1421363107;
    s=origin2015; d=d1.example; cv=none;
    b=pCw3Qxgfs9E1qnyNZ+cTTF3KHgAjWwZz++Rju0BceSiuwIg0Pkk+3RZH/kaiz61
     TX6RVT6E4gs49Sstp41K7muj1OR5R6Q6llahLlQJZ/YfDZ3NImCU52gFWLUD7L69
     EU8TzypfkUhscqXjOJgDwjIceBNNOfh3Jy+V8hQZrVFCw0A=
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed;
    d=d1.example; s=20130426; t=1421363082;
    bh=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKf1pdkxtfGyWaU=;
    h=MIME-Version:To:CC:Subject:Content-Type:Content-Transfer-Encoding;
    b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijr
     vQwbv9uUA1t94kMN0Q+haFo6hiQPnkuDxku5+oxyZWOqtNH7CTMgcBWWTp4QD4G
     d3TRJlgotsX4RkbNcUhlfnoQ0p+CywWjieI8aR6eof6WDQ=
Message-ID: <54B84785.1060301@d1.example>
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe <jqd@d1.example>
To: arc@example.org
Subject: [Lists] Example 1

Hey gang,
This is a test message.
--J.
```

Appendix C. Acknowledgements

This draft originated with the work of OAR-Dev Group.

The authors thank all of the OAR-Dev group for the ongoing help and though-provoking discussions from all the participants, especially: Alex Brotman, Brandon Long, Dave Crocker, Elizabeth Zwicky, Franck Martin, Greg Colburn, J. Trent Adams, John Rae-Grant, Mike Hammer, Mike Jones, Steve Jones, Terry Zink, Tim Draegen.

Grateful appreciation is extended to the people who provided feedback through the discuss mailing list.

Appendix D. Comments and Feedback

Please address all comments, discussions, and questions to dmarc@ietf.org [7]. Earlier discussions can be found at arc-discuss@dmarc.org [8].

Authors' Addresses

Kurt Andersen
LinkedIn
1000 West Maude Ave
Sunnyvale, California 94085
USA

Email: kurta@linkedin.com

Brandon Long (editor)
Google

Email: blong@google.com

Steven Jones (editor)
TDP

Email: smj@crash.com

Seth Blank (editor)
Valimail

Email: seth@valimail.com

Murray Kucherawy (editor)
TDP

Email: superuser@gmail.com