DMARC Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 22, 2018

K. Andersen
LinkedIn
B. Long, Ed.
Google
S. Jones, Ed.
TDP
S. Blank, Ed.
ValiMail

December 19, 2017

Authenticated Received Chain (ARC) Protocol
draft-ietf-dmarc-arc-protocol-10

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 record the status of that authentication 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 DKIM or DMARC can be identified through the ARC set of header fields.

Status of This Memo

Copyright Notice

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 Compliance (DMARC) [RFC7489], validates the domain of the RFC5322.From author 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.

What is needed is a mechanism by which legitimate alteration of a message, which invalidates associated SPF and DKIM information, does not ultimately result in a rejection of an email message on delivery. Authenticated Receive Chain (ARC) builds upon DKIM mechanisms to provide a sequence of signatures that are more survivable than DKIM's and 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 false negatives inherent in use of DKIM and/or SPF themselves.

2. Overview

In DKIM, every participating signing agent attaches a signature that is based on the some of the content of the message, local policy, and the domain name of the participating Administrative Management Domain (ADMD). Any verifier can process such a signature; a verified signature means that the domain referenced in the DKIM-Signture'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 message content that was "covered" by the signature has not been altered since the signature was applied. The signatures themselves are generally independent of one another.

By contrast, an ARC signature conveys the following pieces of information:

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

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

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

This protocol accomplishes each of these by adding a new header field to the message for each of these pieces of information, as follows:

- 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 (see Section 5.2);

- 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; and

- 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.

A distinguishing feature of all of these is that an ARC participant always adds all of them before relaying a message to the next handling agent en route to its destination. Moreover, as described in Section 5.1, they each have an "instance" number that increases with each ADMD in the handling chain so that their original order can be preserved and the three related header fields can be processed as a group.

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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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].

- "ARC set" - A single group of the header fields introduced in Section 2 is called an "ARC set".

- "ARC chain" - The complete sequence of these groups (ARC sets) is called an "Authenticated Received Chain" or merely an "ARC chain". Although the "Received" header field is typically not included in the signed content, the name is based on the notion that this is in essence a cryptographically signed series of header fields that attest to the handling chain of a message much as Received fields always have.

3.1. 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.

4. 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, a Final Receiver is able to make a more-informed local policy decision regarding message delivery to the end user in spite of a DMARC failure.

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). The validated chain status as determined at message receipt must be passed to the sealer function in order for sealing to occur properly; how this is done is considered ADMD-specific and an implementation detail.

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.

4.1. Features of the ARC Protocol

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

4.1.1. 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 4.

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 failing ARC chain SHOULD be treated the same as a message with no ARC chain. [[NOTE TO WORKING GROUP: This paragraph probably is better placed in Verifier actions.]]

4.1.2. Optional Participation

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

4.1.3. Only one ARC Chain (One Chain to Rule Them All)

A message can have only one ARC chain on it at a time. Once broken, the chain cannot be restarted, as the chain of custody is no longer valid and responsibility for the message has been lost.

4.1.4. 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.

4.1.5. Benign nature of an ARC Set

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 value to a Final Receiver evaluating message delivery in the context of a DMARC failure. An ARC chain in general, and each ARC set in particular, provide additional information, and otherwise is benign. Specifically:

- properly adding an ARC set to a message does not damage or invalidate an existing chain, and

- validating a message exposes no new threat vectors (see Section 13).

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.

4.1.6. 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.

4.1.7. Trace Information

ARC includes trace information encoded in the AAR. While section Section 5.2 defines what information must be provided, sealing ADMDs may provide additional information, and validating receivers may use or ignore this trace information as they wish.

5. The ARC Header Fields

5.1. Instance ('i=') Tag

The header fields comprising a single ARC set are identified by the presence of a string in the value portion of the header field 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:

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

Valid ARC sets must have exactly one instance of each header field for a given position value. (Note that when multiple algorithms are supported, there is some nuance to this statement - see Section 10.)

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.

5.1.1. Valid Range for Instance Tags

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

ARC implementations MUST support at least ten (10) ARC sets.

An effective operational maximum will have to be developed through deployment experience in the field and will be documented within [ARC-USAGE] once determined.

ARC chains with more than the defined operational maximum count MUST be marked with "cv=fail".

5.2. ARC-Authentication-Results (AAR)

The ARC-Authentication-Results header field is syntactically and semantically identical to an Authentication-Results header field (defined in Section 2.2 of [RFC7601] (A-R)), except that several optional data fields SHOULD be added. ([[NOTE: these optional data fields are being proposed as amendments to [RFC7601] through a "bis" process. Depending on the sequencing for this specification and said "7601bis" work, it may be possible to drop the noted sections from this specification.]]) The first element ("Authentication-Results:") in authres-header is replaced with arc-authres-prefix as follows:

```
arc-authres-header-prefix = "ARC-Authentication-Results:" [CFWS] arc-info
arc-info = instance *([CFWS] propspec) [CFWS] ";"
```

The purpose of this header field is to transmit the results of any authentication done on the message upstream to participating ADMDs validating and continuing the chain.

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

5.2.1. ptypes and properties for arc-info

[[NOTE: This section is being proposed as an amendment to [RFC7601] through a "bis" process. Depending on the sequencing for this specification and said "7601bis" work, it may be possible to this section from this specification.]]

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.

The AAR, through these ptype-properties stamped in arc-info, provide a mechanism for this information to survive transit.

The ptypes and properties defined in this section SHOULD be stamped in the AAR:

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

o header.ds - The domain/selector pair for each DKIM signature on the message (header.ds=example.com,selector). Note that this is a concatenation of the header.d and header.s field values from [[TODO: reference DKIM A-R method spec]] separated by the comma character (0x2c). These values are joined into a single arc-info field value in order to avoid indexing and correlation ambiguity between the possible multiple DKIM signatures which may be found on any given message;

o arc.closest-fail - The hop number of the most recent AS that fails to validate, or 0 if all hops pass.

5.3. ARC-Message-Signature (AMS)

The ARC-Message-Signature header field is syntactically and semantically identical to a DKIM-Signature header field [RFC6376], with the following exceptions:

o There is an "i" tag, as described in Section 5.1.

o There is no "v" tag.

ARC-Seal header fields MUST NOT be included in the content covered by 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.

The AMS header field MAY include (sign) the AAR header field(s).

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

As with a DKIM-Signature, the purpose of this header field is to allow the ADMD generating it to take some responsibility for handling this message as it progresses toward delivery.

5.4. 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 5.1.

- The ARC-Seal covers none of the body content of the message. It only covers specific header fields. (See below: Section 5.4.2.) As a result, no body canonicalization is done. Further, only "relaxed" header canonicalization (Section 3.4.2 of [RFC6376]) is used.

- The only supported tags are "i" (Section 5.1 supercedes the [RFC6376] definition), and "a", "b", "d, "s", "t". The latter 5 tag definitions are copied from Section 3.5 of [RFC6376].

- An additional tag, "cv" is defined. (See below: Section 5.4.1)

5.4.1. The 'cv' Tag

A new tag "cv" (chain validation) indicates the the outcome of evaluating the existing ARC chain upon arrival at the ADMD that is adding this header field. It accepts 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) from a Message Submission Agent (MSA) or when any upstream MTAs 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.

In ABNF terms:

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

5.4.2. Implicit Header Fields

The ARC-Seal signs a canonicalized form of the ARC set header values. The ARC set header values are compiled in increasing instance order, starting at 1, and inclue 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.

Note that the signing scope for the ARC-Seal is modified in the situation where a chain has failed validation (see Section 8.3).

6. Verifier Actions

The verifier takes the following steps to determine the current state of the ARC chain on the message. Canonicalization, hash functions, and signature validation methods are imported from Section 5 of [RFC6376].

[[Note: need markdown flag to have subordinate numbering distinction]]

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

2. If the form of any ARC set is invalid (e.g., does not contain exactly one of each of the three ARC-specific header fields), then the chain state is "fail" and the algorithm stops here.

    1. To avoid the overhead of unnecessary computation and delay from crypto and DNS operations, the cv value for all ARC-Seal(s) MAY be checked at this point. If any of the values

are "fail", then the overall state of the chain is "fail" and the algorithm stops here.

3. Conduct verification of the ARC-Message-Signature header field bearing the highest instance number. If this verification fails, then the chain state is "fail" and the algorithm stops here.

4. For each ARC-Seal from the "N"th instance to the first, apply the following logic:

    1. If the value of the "cv" tag on that seal is "fail", the chain state is "fail" and the algorithm stops here. (This step SHOULD be skipped if the earlier step (2.1) was performed)

    2. In Boolean nomenclature: if ((i == 1 && cv != "none") or (cv == "none" && i != 1)) then the chain state is "fail" and the algorithm stops here (note that the ordering of the logic is structured for short-circuit evaluation).

    3. Initialize a hash function corresponding to the "a" tag of the ARC-Seal.

    4. Compute the canonicalized form of the ARC header fields, in the order described in Section 5.4.2, using the "relaxed" header canonicalization defined in Section 3.4.2 of [RFC6376]. Pass the canonicalized result to the hash function.

    5. Retrieve the final digest from the hash function.

    6. Retrieve the public key identified by the "s" and "d" tags in the ARC-Seal, as described in Section 4.1.6.

    7. Determine whether the signature portion ("b" tag) of the ARC-Seal and the digest computed above are valid according to the public key. (See also Section Section 8.4 for failure case handling)

    8. If the signature is not valid, the chain state is "fail" and the algorithm stops here.

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

[[Note from Dave: possibly delete the following paragraph as it is more usage/procedural than specification guidance.

KA: It was added to clarify the separation of the verification and signing steps as some of the initial implementations failed to realize that they were not necessarily done in one fell swoop.

KA (v-10): With the addition of the {protocol-elements} section, does the WG think that this can be reasonably removed from this location?]]

The verifier should save the cv state for subsequent use by any sealing which may be done later (potentially after message modification) within the same trust boundary. The cv state may be recorded by sealing at the time of verification in an initial ARC set (for the ADMD) or may be recorded out of band depending on the architecture of the ADMD.

7. Signer Actions

[[Note from Dave: This seems more like implementation guidance than specification detail. KA: see explanation just above referring to the previous note.]]

This section includes a specification of the actions an ARC signer takes when presented with a message.

The signer MUST undertake the following steps:

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 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 5.3 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 5.4 above, the chain validation status as determined in Section 6, and instance number N+1.

8. Usage of ARC and Chain Validity

8.1. Relationship between DKIM-Signature and AMS signing scopes

DKIM-Signatures SHOULD never sign any ARC header fields.

[[KA: Response to Dave's concern: If DKIM covers ARC and ARC covers DKIM, which comes first? The chicken or the egg? I'm open to alternate ways to phrase this without opening the "modifying the DKIM spec" can of worms.]]

8.2. Assessing Chain Validity Violations

Email transit can produce broken signatures for a wide variety of benign reasons. This includes possibly breaking one or more ARC signatures. Therefore, receivers need to be wary of ascribing motive to such breakage although patterns of common behaviour may provide some basis for adjusting local policy decisions.

ARC does not attempt to protect an entire message. There are various ways that a message can still be problematic, in spite of having a valid ARC chain. Consequently, all normal, content-based analysis SHOULD still be performed on any message having a valid chain of ARC header sets.

8.3. 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 'i=' instance headers created by the MTA which detected the malformed chain, as if this newest ARC set was the only set present.

8.4. 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.

8.5. Responding to ARC Validity Violations

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.

9. 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.

9.1. 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.

9.2. Recording (local) ARC Evaluation Results

Receivers MAY 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 5.2.1.

9.3. DMARC Reporting of ARC Findings - Interim

[[Note: Discussion on the IETF DMARC-WG list has indicated some interest in more substantial reporting for analytic purposes. To support that effort, the following guidance is provided only as an interim, minimal data set. A more complete reporting construct will be specified in a related spec - TBD. (see the additional fields specified in Section 5.2.1)]]

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 (Section 9.1 and Section 5.2.1):

```
<policy_evaluated>
  <disposition>delivered</disposition>
  <dkim>fail</dkim>
  <spf>fail <comment>source.ip=10.0.0.1</comment></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</comment>
  </reason>
</policy_evaluated>
```

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

Mediators 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.

10. Supporting Alternate Signing Algorithms

[[Note: Some additional development of this section is needed.]]

In the following branch diagrams, each algorithm is represented by an 'A' or 'B' at each hop to depict the ARC chain that develops over a five hop scenario. 'x' represents a hop that does not support that algorithm.

Note that during a transitional period where multiple algorithms are allowed, all of the statements in this spec which refer to "exactly one set of ARC headers per instance" need to be understood as "at least one set per instance and no more than one instance-set per algorithm".

10.1. Introductory Period

Intermediaries MUST be able to validate ARC chains built with either algorithm but MAY create ARC sets with either (or both) algorithm.

The introductory period should be at least six (6) months.

10.2. Co-Existence Period

Intermediaries MUST be able to validate ARC chains build with either algorithm and MUST create ARC sets with both algorithms. Chains ending with either algorithm may be used for the result.

10.3. Deprecation Period

ARC sets built with algorithms that are being deprecated MAY be considered valid within an ARC chain, however, intermediaries MUST NOT create additional sets with the deprecated algorithm.

The deprecation period should be at least two (2) years.

10.4. Obsolescence Period

ARC sets built with algorithms that are obsolete MUST NOT be considered valid within an ARC chain. Intermediaries MUST NOT create any sets with any obsoleted algorithm.

11. 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.

12. IANA Considerations

This specification adds three new header fields as defined below.

12.1. 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 5.4)

  Status: active

  Version: 1

12.2. 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]

13. Security Considerations

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

13.1. 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.

13.2. 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 partipant 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.

13.3. 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).

14. Implementation Status

[[Note: For minimizing section number references when the RFC editor removes this section, it has been moved to be the last section of the document before the Appendicies.]]

[[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.

14.1. 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 [1]

14.2. 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 [2]

14.3. 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] (https://github.com/ValiMail/arc_test_suite) with both python and python3.

Licensing: Open/Other (same as dkimpy package = BCD version 2)

Contact Info: https://launchpad.net/dkimpy

14.4. 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-06]

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.

- 2017-07-15: Testing showed problems with the hash calculation for the AMS header b= field. Several other bugs were discovered and were either fixed during the following week of IETF meetings or are under active repair.

- 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.

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

14.5. Mailman 3.1+ patch

Organization: Mailman development team

Description: Integrated ARC capabilities within the Mailman 3.1+ package

Status of Operation: Patch submitted

Coverage: Unknown

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

14.6. 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/

14.7. 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

14.8. PERL Mail::Milter::Authentication module

Organization: FastMail

Description: Email domain authentication milter, previously included SPF / DKIM / DMARC, now has ARC added

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

15. References

15.1. Normative References

[RFC1345] Simonsen, K., "Character Mnemonics and Character Sets", RFC 1345, DOI 10.17487/RFC1345, June 1992, <https://www.rfc-editor.org/info/rfc1345>.

[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>.

[RFC2142] Crocker, D., "Mailbox Names for Common Services, Roles and Functions", RFC 2142, DOI 10.17487/RFC2142, May 1997, <https://www.rfc-editor.org/info/rfc2142>.

[RFC2606] Eastlake 3rd, D. and A. Panitz, "Reserved Top Level DNS Names", BCP 32, RFC 2606, DOI 10.17487/RFC2606, June 1999, <https://www.rfc-editor.org/info/rfc2606>.

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

[RFC4686] Fenton, J., "Analysis of Threats Motivating DomainKeys Identified Mail (DKIM)", RFC 4686, DOI 10.17487/RFC4686, September 2006, <https://www.rfc-editor.org/info/rfc4686>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[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>.

[RFC5585] Hansen, T., Crocker, D., and P. Hallam-Baker, "DomainKeys Identified Mail (DKIM) Service Overview", RFC 5585, DOI 10.17487/RFC5585, July 2009, <https://www.rfc-editor.org/info/rfc5585>.

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

[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>.

[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>.

[RFC6651] Kucherawy, M., "Extensions to DomainKeys Identified Mail (DKIM) for Failure Reporting", RFC 6651, DOI 10.17487/RFC6651, June 2012, <https://www.rfc-editor.org/info/rfc6651>.

[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>.

15.2. Informative References

[ARC-DRAFT-05]
Andersen, K., Long, B., and S. Jones, "Authenticated Received Chain (ARC) Protocol (I-D-06)", n.d., <https://tools.ietf.org/html/draft-ietf-dmarc-arc-protocol-06>.

[ARC-DRAFT-06]
Andersen, K., Long, B., and S. Jones, "Authenticated Received Chain (ARC) Protocol (I-D-05)", n.d., <https://tools.ietf.org/html/draft-ietf-dmarc-arc-protocol-05>.

[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>.

[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>.

15.3. URIs

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

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

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

[4] mailto:dmarc@ietf.org

[5] 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.]]

(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 is 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 [4]. Earlier discussions can be found at arc-discuss@dmarc.org [5].

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