Authenticated Received Chain (ARC)
draft-andersen-arc-00

Abstract

Authenticated Received Chain (ARC) permits an organization which is creating or handling email to indicate their involvement with the handling process by adding a cryptographicly signed header (or headers) in a manner analagous to that of DomainKeys Identified Mail (DKIM). Assertion of responsibility is validated through a cryptographic signature and by querying the Signer’s domain directly to retrieve the appropriate public key. Changes in the message which may break DKIM, may be tracked through the ARC set of headers.

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 http://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 April 18, 2016.
Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

Table of Contents

1. Introduction ........................................ 3
2. Requirements ....................................... 4
   2.1. Primary Design Criteria .......................... 4
   2.2. Out of Scope ..................................... 4
   2.3. Utility .......................................... 4
3. Terminology ........................................ 4
4. Overview ............................................ 5
5. Definition .......................................... 5
   5.1. Description of the new headers ................... 6
   5.1.1. ARC-Seal ....................................... 6
   5.1.2. ARC-Message-Signature ......................... 7
   5.1.3. ARC-Authentication-Results .................... 8
   5.2. Constructing the ARC-Seal Header Set ............ 8
   5.2.1. Handling Violations in the ARC Header Set .... 8
   5.3. Constructing the ARC-Message-Signature header ... 9
   5.4. Key Management and Binding ....................... 9
   5.4.1. Namespace ...................................... 9
6. Usage ................................................. 9
   6.1. Participation .................................... 9
   6.2. Relationship between DKIM Signatures and ARC Headers .. 10
   6.3. Relationship of ARC-Message-Signatures and ARC-Seals .. 10
   6.4. Validating the ARC set of headers ............... 10
   6.5. Assessing violations of ARC set validity .......... 10
   6.6. Reporting violations of ARC set validity .......... 11
   6.7. Recording results of ARC evaluation .......... ... 11
7. Privacy Considerations ............................ 11
8. IANA Considerations ............................... 11
9. Security Considerations ........................... 11
1. Introduction

The development of strong domain authentication through SPF and DKIM has led to the implementation of the DMARC framework [RFC7489]. Implicit within the DMARC framework is a requirement that any intermediaries between the source system and ultimate receiver system must preserve the validity of the DKIM signature; however, there are common email practices which break the DKIM validation ([DMARC-INTEROP]). This proposal is intended to define an Authenticated Received Chain (ARC) to address the problems with the untrustworthiness of the standard Received header sequence so that receivers can develop a more nuanced interpretation to guide any local policies related to messages which arrive with broken domain authentication.

Forgery of the Received headers is a common tactic for bad actors. One of the goals of this proposal is to define a comparable set of trace headers which can be relied upon by receivers in so far as all ADMD (ADministrative Management Domain) handlers of a message participate in the ARC chain.
2. Requirements

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

2.1. Primary Design Criteria

- Provide a method by which a "chain of custody" can be documented for email messages
- Not require changes for senders of email
- Support the complete verification of the ARC header set by each hop in the handling chain
- Work at internet scale

2.2. Out of Scope

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

2.3. Utility

The ARC-related set of headers can be used (when validated) to determine the path that an email message has taken between the sending system and receiver. Subject to the cautions mentioned below under Section 9, this information can assist in determining any local policy overrides to for violations of sender domain authentication policies.

3. 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].

Readers are encouraged 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].
4. Overview

When an email is received without a properly validated domain author, the inability to believe the accuracy of a series of Received headers prevents receiving systems from having a way to infer anything about the handling of the message by looking at the ADMD’s through which the message has traveled.

With the implementation of this proposal, participating ADMDs would be able to securely register their handling of an email message. If all intermediaries participate in the ARC process, then receivers will be able to rely upon the chain and make local policy decisions informed by that information.

The ARC set of headers provides a method by which participating intermediaries can indicate the hand-offs for email messages.

5. Definition

This proposal 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: OAR-Dev Group
  
  Specification document(s): [I-D.ARC]
  
  Related information: [RFC6376]

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

- Header field name: ARC-Authentication-Results
5.1. Description of the new headers

5.1.1. ARC-Seal

ARC-Seal is a Structured Header Field as defined in Internet Message Format ([RFC5321]), and thus all of the related definitions in that document apply.

This header field is added at the top of the message as it transits MTAs that do authentication checks, so some idea of how far away the checks were done can be inferred. It is therefore considered to be a trace field as defined in [RFC5321], and thus all of the related definitions in that document apply.

The ARC-Seal makes use of the same "tag=value" construction as defined in [RFC6376], section 3.2.

The value of the header field consists of an authentication identifier, and a series of statements and supporting data. The statements are of the form "tag=value" and indicate relevant data about the signing of the ARC set of headers. The header field can appear more than once in a single message.

5.1.1.1. Tags in the ARC-Seal header

5.1.1.1.1. Mandatory

  o i = instance or sequence number; monotonically increasing at each "sealing" entity beginning with ‘1’
  o a = hash algorithm (SHA256 as example) (as per [RFC6376] "a" tag)
  o t = timestamp (seconds since Unix epoch) (as per [RFC6376] "t" tag)
  o s = Selector for key ("s=seal2015") (as per [RFC6376] "s" tag)
  o d = domain for key ("d=example.com") (as per [RFC6376] "d" tag)
o  k = selector key path (specifies which ARC-Message-Signature is being asserted)

o  b = signature of the header hash (as per [RFC6376] "b" tag)

o  cv = chain validation status: values = ** 'V' = valid chain received; ** 'N' = no pre-existing chain; ** 'P' = permanent error, the chain as received does not validate; ** 'T' = temporary error, such as a DNS lookup error

5.1.1.2. Differences between DKIM-Signature and ARC-Seal

No 'bh' value is defined for ARC-Seal.

ARC-Seal does not use the 'h' list of headers that is defined for DKIM-Signatures because the list of applicable headers is fully determined by the construction rules (see Section 5.2).

ARC-Seal does not use the 'c' (canonicalization) tag because only 'relaxed' canonicalization [RFC6376] is allowed for ARC-Seal headers.

5.1.2. ARC-Message-Signature

The ARC-Message-Signature header is a special variant of a DKIM-Signature [RFC6376], using the relaxed header canonicalization rules specified in [RFC6376] but with a different rules for the header (h=) list to prevent mis-appropriation as a DKIM-Signature.

5.1.2.1. Differences between DKIM-Signature and ARC-Message-Signature

5.1.2.1.1. 'h' value

When constructing the 'h' list for the ARC-Message-Signature, the following headers must be implicitly signed, but not listed within the list: 1. Message-ID 2. Date 3. From 4. To 5. Subject

This provides a unique h list which prevents re-use or misappropriation of the ARC-Message-Signature as if it was a DKIM-Signature.

5.1.2.1.2. 'i' value

For the ARC-Message-Signature, the 'i' value is the corresponding instance which matches the 'i' value of the related ARC-Seal (see Section 5.1.1.1.1).
5.1.2.1.3. ‘v’

‘v’ is not defined for an ARC-Message-Signature and is not allowed.

5.1.3. ARC-Authentication-Results

ARC-Authentication-Results is a direct copy of the Authentication-Results header [RFC7601] created for archival purposes by the first MTA outside of the trust boundary of the originating system. (See also [OAR] for a similar usage.)

The value of the header field (after removing comments) consists of an authentication identifier, and then a series of statements and supporting data. The statements are of the form "method=result" and indicate which authentication method(s) were applied and their respective results. For each such statement, the supporting data can include a "reason" string and one or more "property=value" statements indicating which message properties were evaluated to reach that conclusion. The header field can appear only once in a single message.

5.2. Constructing the ARC-Seal Header Set

The ARC-Seal is built in the same fashion as the analogous DKIM-Signature [RFC6376], using the relaxed header canonicalization rules specified in [RFC6376] but with a strict ordering component for the headers which are covered by the cryptographic signature:

1. The ARC-Seal headers MUST be ordered in descending instance (i=) order.

2. The referenced ARC-Message-Signatures (via k= tags) MUST immediately follow the ARC-Seal instance which included the reference.

3. The ARC-Authentication-Results header MUST be the last item in the list.

5.2.1. Handling Violations in the ARC Header Set

When ordering the ARC-Seal headers, if there are gross violations of this protocol, such as duplicated instance numbers, such headers (and any associated headers) shall be ordered as follows:

- Any headers which are complete duplicates shall be deduplicated - leaving only one instance of each unique header; then any remaining order dependencies shall be ordered as follows:
1. (First) By descending order of t=
2. (Second) By ascending order of d=
3. (Third) By ascending order of s=
4. (Fourth) By ascending order of b=
5. (Finally) By ascending UTF-7 sort order for the entire canonicalized header

5.3. Constructing the ARC-Message-Signature header

The ARC-Message-Signature header is built in the same fashion as a DKIM-Signature [RFC6376], using the relaxed header canonicalization rules specified in [RFC6376] but with a different rules for the header (h=) list to prevent mis-appropriation as a DKIM-Signature. ARC-Message-Signatures (via k= tags) MUST immediately follow the ARC-Seal instance which includes the reference.

5.4. Key Management and Binding

The public keys for ARC-Seals follow the same requirements and semantics as those for DKIM-Signatures [RFC6376]. Operators may use distinct selectors for the ARC-Seals at their own discretion.

5.4.1. Namespace

All ARC-Seal keys are stored in the same subdomain as DKIM keys [RFC6376]: ".domainkey". Given an ARC-Seal field with a "d=" tag of "example.com" and an "s=" tag of "foo.bar", the DNS query will be for "foo.bar._domainkey.example.com".

6. Usage

For a more thorough treatment of the recommended usage of the ARC headers for both intermediaries and end receivers, please consult [ARC-USAGE].

6.1. Participation

The inclusion of additional ARC-Seal headers should be done whenever a trust boundary is crossed and especially when prior DKIM-Signatures may not survive the handling which is being performed (such as some mailing lists which modify the content of messages or some gateway transformations). Note that trust boundaries may or may not exactly correspond with ADMD boundaries.
Each participating ADMD MUST validate the preceding ARC set of headers as a part of asserting their own seal. Even if the set is determined to be invalid, a participating ADMD SHOULD apply their own seal because this can help in analysis of breakage points in the chain.

6.2. Relationship between DKIM Signatures and ARC Headers

Any DKIM-Signatures SHOULD not include any of the ARC-Seal, ARC-Message-Signature, or ARC-Authentication-Results headers in the scope of their header list.

ARC-Message-Signatures SHOULD include all DKIM-Signatures within their scope.

6.3. Relationship of ARC-Message-Signatures and ARC-Seals

The ARC-Message-Signature(s) should not include any of the ARC-Seal headers in their coverage scope in order maintain a separation of responsibilities. When adding an ARC-Authentication-Results header, it should be added before computing the ARC-Message-Signature. When "sealing" the message, an operator must create the ARC-Message-Signature before the ARC-Seal in order to reference it and embed the ARC-Message-Signature within the ARC-Seal signature scope.

Each ARC-Seal ties into its respective ARC-Message-Signature through the k= and i= fields.

6.4. Validating the ARC set of headers

Validation of the ARC headers can be performed step-wise by building up the sequence in order as defined in Section 5.2 and evaluating the correctness of the b= signature at each step. If a violation of the construction rules is found, for instance missing or repeated instance numbers or an otherwise invalid ARC-Seal header, validation fails and should be indicated as 'P'(ermanent error).

6.5. Assessing violations of ARC set validity

There are a wide variety of ways in which the ARC set of headers can be broken. Receivers should be wary of ascribing motive to such breakage although patterns of common behaviour may provide some basis for adjusting local policy decisions.

This proposal is exclusively focused on well-behaved, participating intermediaries that result in a valid chain of ARC-related headers. The presence of such a well-formed valid chain should also not be over-interpreted since malicious content can be easily introduced by
otherwise well-intended senders through machine or account compromises. All normal content-based analysis should still be performed on any messages bearing an ARC sequence.

6.6. Reporting violations of ARC set validity

If a receiver determines that the ARC set of headers has a permanent error, the receiver MAY signal the breakage through the extended SMTP response code X.7.7 [RFC3463] "message integrity failure" [ENHANCED-STATUS].

6.7. Recording results of ARC evaluation

Receivers may add an "arc=pass" or "arc=fail" method annotation into their local Authentication-Results [RFC7601] header.

7. Privacy Considerations

The ARC-Seal chain provides a verifiable record of the handlers for a message. Anonymous remailers will probably not find this to match their operating goals.

8. IANA Considerations

This proposal adds three new headers as defined above.

This proposal adds a new method to the [RFC7601] header.

9. Security Considerations

Recipients are cautioned to treat messages bearing ARC-Seal chains 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-Seal chain may be used to provide input to local policy engines in cases where the sending system’s DKIM-Signature does not validate.

9.1. Preventing Repurposing of ARC Headers

The ARC headers have been designed in such a way that they can not be re-used as standard DKIM-Signatures to prevent mis-use.

9.2. Messages Which Transit the Same ADMD More Than Once

Messages which loop in and out of an ADMD may lead to confusion about the scope of a particular set of ARC headers. The use of coordinated instance (i=) values and the non-confusability of the ARC-Message-
Signature vs. a DKIM-Signature are designed to prevent misunderstandings.

10. References

11. References

11.1. Normative References


11.2. Informative References

[ARC-USAGE]

[DMARC-INTEROP]

[ENHANCED-STATUS]
11.3. URIs

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

Appendix A. Appendix A - Example Usage

A.1. Example 1: Simple mailing list

A.1.1. Here’s the message as it exits the Origin:

Return-Path: &lt;jqd@d1.example&gt;
Received: from \[10.10.10.131\] (w-x-y-z.dsl.static.isp.com \[w.x.y.z]\)
 by segv.d1.example with ESMTP id t0FN4a80084569;
 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=EoJqaaRvhrngQxmQ3VnRIIMRBgecuKflpdktfGyWaU=;
 h=Message-ID:Date:From:MIME-Version:To:CC:Subject:Content-Type:
 Content-Transfer-Encoding;
 b=HxsvPubDE+R96v9dM9Y7V3dJUXvajd6rvF5ec5BPe/vpVBRJnD4I2weEIyYijrvQw
 bv9uUA1t94kMNQOQ+haFo6h1QPknkuDxku5+oxy2WOqtNH7CTMgcBWWtP4QD4d3TRJ1
 gotsX4RkbNcUlHfncQ0p+CywWjieI8aR6e6f6WDQ=;
Message-ID: &lt;54B84785.1060301@d1.example&gt;
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe &lt;jqd@d1.example&gt;
To: arc@dmarc.org
Subject: Example 1

Hey gang,
This is a test message.
--J.
A.1.2. Message is then received at example.org

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

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

Here’s the message as it exits example.org:
Hey gang,
This is a test message.
--J.
A.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 Auth-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: &lt;jqd@d1.example&gt;
Received: from example.org (example.org [208.69.40.157])
    by clothilde.example.com with ESMTP id
d20mr22663000yk8.93.1421363207
    for &lt;fmartin@example.com&gt;; 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;
    b=pClwlQxgfs9ElqnyvZ+cITE3KaxHgAjWWzz+2RjuOBceSluwIgOPkk+3RZH/kaiz61
    TX6RTVe4gs49Stnp41K7mujl0R5R6Q6lalhlQJZ/7ydE3NIuC52gFWLjUD7L69
    EU8ZyjpfkUhsccqXjOjgDwjI5ekBNN0fha3Jv+V8hQ3rVFCw0A=
ARC-Message-Signature: v=1; a=rsa-sha256; c=simple/simple;
    d=example.org; s=clochette; t=1421363105;
b=FiQQym3HhxStuzauxV4UcO2o55ExATNF4uBvEoy7k3s=;
    h=Date:To:Subject:List-Id:List-Archive:List-Post:
    List-Reply-To:List-Unsubscribe:List-Subscribe:From:Reply-To;
    b=Wbe4EIvAnvA8obWwrKpmlhmmdlvj0dvop0sikaiGOOug321Tnc74/IWv1PxfP
    1F5yYVF0m5wcmKoa82tKktU0O3yintAkqanly7GJuFCDeSAIfQHhStVV7BzAr3
    a=m4bwa6R1Dgr3OPJi1678dZHfztFWyjwIUxR5Ajxj/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 &lt;arc@example.org&gt;; Thu, 14 Jan 2015 15:01:30 -0800 (PST)
    (envelope-from jqd@d1.example)
ARC-Authentication-Results: lists.example.org;
    spf=pass smtp.mfrom=jqd@d1.example;
    dkim=pass (1024-bit key) header.i=0@example.org; 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 t0FN4a80084569;
    Thu, 14 Jan 2015 15:00:01 -0800 (PST)
    (envelope-from jqd@d1.example)
Hey gang,
This is a test message.
--J.

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

Here's the message as it exits the Origin:

Hey gang,
This is a test message.
--J.
A.2.2. Message is then received at example.org

A.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 Auth-Results or ARC-Seal headers are present * example.org adds ARC-Auth-Results header * example.org adds usual Received: header * example.org adds a ARC-Seal header

Here’s the message as it exits Step A:
Hey gang,
This is a test message.
--J.
A.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 Auth-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-Auth-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=V; k=20120806;
b=sjHDMr1RZOMu15eVE0GscRHWbQHcy971vrdHuQ8h+f2CFIrU1KOE44x3LQwDWR
Ybdjf5fcM9Mdlc0AhC+cP59Bq9Y9DHwMDzwR7n7Nvb4kY+tSaVnLoIoA9l1F/sut
xO+RRNrf0CFw==
ARC-Message-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
d=gmail.com; s=20120806;
h=mime-version:date:message-id:subject:from:to:content-type
 :x-original-sender:x-original-authentication-results:precedence:
 :mailing-list:listed:listed-post:listed-help:listed-archive:sender
 :list-unsubscribe:reply-to;
bh=2+g2zwliUK2VJ7bpoO2MrU19WhcA4JnjiohFm9Z/g=
b=cWq3Qgqs9lEatnNYZ+cTTF3KHgAjWwZz++Rju0BceSiruwIgPkk+3RZH/kai61
TX6VRT6E4gs49Stp41K7mu1OR5R6Q61lahL1QJ2/YfDZ3NmCU52gFWLUD7L69
EUB7zypkFuUscqXJ0jGw1jceBNNOfh3y+Vh8QrZVFCwoA801lebYV/h1Bmfrs
LF1E80SmCpijONTPQ66gShoh/k66r2fqr6as56gL/WA3+3Id48EhXv1GcJRF8w
KdJqi65xqPfTRE+Ben5e6lZg66kr265NTDAU8p8fQNUfz49+MA+QwDBJtXw
bQoZyRtbB6Xq60MyszUB8kw==
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: gmail.com; spf=fail
smtp.from=jqd@d1.example; dkim=pass (1024-bit key)
header.i=@example.org; dmarc=fail; arc=pass
ARC-Seal: v=1; i=1; a=rsa-sha256; t=1421363107;
s=seal2015; d=example.org; c=none;
b=cWq3Qgqs91QnyNZ+cTTF3KHgAjWwZz++Rju0BceSiruwIgPkk+3RZH/kai61
TX6VRT6E4gs49Stp41K7mu1OR5R6Q61lahL1QJ2/YfDZ3NmCU52gFWLUD7L69
```

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

A.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 Auth-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:

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

Return-Path: &lt;jqd@d1.example&gt;
Received: from mail-ob0-f188.google.com (mail-ob0-f188.google.com [208.69.40.157]) by clothilde.example.com with ESMTP id d200mr22663000yk9.14.1421363268 for &lt;fmartin@example.com&gt;; 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=V; k=20120806;
b=sjHDmRi2R0Mu15eVEOGscRHHwbQHy971vdhuHQ8h+f2CFtIrXuKOE44k3LQwDWR
YbDj5fcM9Mdc1AhC+cP59BQ9q9DHWMDzwRTnM7NVb4kY+TsAvNLoI0aP91F/sut
tx0+RRNn0FCFw==
ARC-Message-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
d=gmail.com; s=20120806;
h=mime-version:date:message-id:content-type
:x-original-sender:x-original-authentication-results:precedence
:mailing-list:list-id:post:list-help:list-archive:sender
:list-unsubscribe:reply-to;
bh=2+gZWzHUK2V7JbpoO2MTu19WvhcA4JnjiohFm9Zz/g=;
b=pCw3Qxgfs9ElqnyN2+cTTF3KhgAjWw2z++Rju0BceSiuwIg0Pkk+3RZ/khiaz61
TX6RVT6E4gs49Stsp41K7muj10R5R6q61ahL1Q7JZ/YfD33NImCU52gFELUD7L69
EUB7zyfpkUhsqcXj0JgDwj1cBeBNNOf3j3y+yVhQZBvFwC0Ab8O11e1bYV/hIBmfsH
LF1E80hMPCmJOnFTQBG65Hoh/kE6N2g6paSng/LWA3+g3Id8ElhXHvIGcJRFEM
KdJjIw5cxdqPTRW+BnR5ee6Tzt6Q6krz265NTDIAU8p8fQNu1fzj49yMA+QwDBjTjxw
bQo3yRtbB6xq60mYaszUB8kw==
Received: by mail-yk0-f179.google.com with SMTP id 19so2728865ykq.10
for &lt;mailbox@gmail.com&gt;; Thu, 14 Jan 2015 15:02:45 -0800 (PST)
Authentication-Results: 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=N; k=clochette;
b=pCw3Qxgfs9ElqnyN2+cTTF3KhgAjWw2z++Rju0BceSiuwIg0Pkk+3RZ/khiaz61
TX6RVT6E4gs49Stsp41K7muj10R5R6q61ahL1Q7JZ/YfD33NImCU52gFELUD7L69
EUB7zyfpkUhsqcXj0JgDwj1cBeBNNOf3j3y+yVhQZBvFwC0Ab8O11e1bYV/hIBmfsH
LF1E80hMPCmJOnFTQBG65Hoh/kE6N2g6paSng/LWA3+g3Id8ElhXHvIGcJRFEM
KdJjIw5cxdqPTRW+BnR5ee6Tzt6Q6krz265NTDIAU8p8fQNu1fzj49yMA+QwDBjTjxw
bQo3yRtbB6xq60mYaszUB8kw==
Hey gang,
This is a test message.
--J.

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

A.3.1. Here’s the message as it exits the Origin:
Return-Path: &lt;jqd@d1.example&gt;
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 t0FN4a80084569;
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=N; k=20130426;
b=pCw3Qxgfs9E1qnyNZcTTF3KHgaJwWzZ3zz+Rju0BceSiuwIgi0Pkkk+3RZH/kaiz61T
X6RVrP6Es49Stsp41K7muji0R5r6Q61ahL1QJZ/YfD23NImCU529FWLUDe71L69EU
8TzyppfUhsCqXjoJgDwjJceBNNOFh3Jy+V8qQzrVFCwoA=
ARC-Message-Signature: v=1; a=rsa-sha256; c=relaxed/simple;
d=d1.example; s=20130426; t=1421363082;
bh=EoJqaaRvhrngQxmqQVnRIIMRBgecuKf1pdxtfGyWuaU=;
h=Message-ID:Date:From:MIME-Version:To:CC:Subject:Content-Type:
Content-Transfer-Encoding;
b=HxsvPublDE+R6v9dM9Y7V3sdJUXvajd6rv5ec5BPe/vpVBRJnD4I2weE1yY1jrv
Qwvb9uUAI9kMNNOQhA06hiQPkuDxku5+oxyZWOqfNH7CTMgcBWNTP4QD4Gd3
TRJ1jotsX4RkkNCUihfnc0q0p+CywWjieI8aR6eGf6WDQ=
Message-ID: &lt;54B84785.106030@18d1.example&gt;
Date: Thu, 14 Jan 2015 15:00:01 -0800
From: John Q Doe &lt;jqd@d1.example&gt;
To: arc@example.org
Subject: Example 1

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

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

A.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-Author-Results header - example.org adds usual Received: header - example.org adds a DKIM-Signature - example.org adds an 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") - link to DKIM-Signature ("k=clochette") - signature ("b=")

Here's the message as it exits Step A:
Hey gang,
This is a test message.
A.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 Auth-Results: and Received: header
- Determines that message fails DMARC - Checks for ARC-Seal: header
- 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-Authorization-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 - link to DKIM-Signature ("k=20120806")
- signature of the chain

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

Return-Path: &lt;jqd@d1.example&gt;
ARC-Seal: i=2; a=rsa-sha256; t=1421363253;
 s=notary01; d=gmail.com; cv=V; k=20120806;
b=sjHDMrirZ20Mu15eVEOGscRHWbQHCy971vruuHQ8h+f2CFIrxxUIkOE44x3LQwD
 WRyBdf5fcM9Mdc1ahC+CpP59BQ9Y9DHwMDzwRTnM7NVb4kY+TsVa1hLo10AP91F
 /suTx0+RRNRe0fCf8w==
ARC-Message-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
d=gmail.com; s=20120806;
h=mime-version:date:message-id:subject:from:to:content-type:
x-original-sender:x-original-authentication-results:precedence:
mailing-list:list-id:list-post:list-help:list-archive:sender:
    :unsubscribe:reply-to;
bh=2+g2wZhUK2V7Jbpo2MTrU19WvhcA4JnjiohFm92Z/g=;
b=pCW3QgxfS9ElqnyN2+CTTF3KHaJWzZz++RjU0BceSiwWlgPkkk+3RZH/kaiz6
  1TX6RVT6E4gs49SAtyp41K7muj1OR5Q6Q611ahL1QJZ/YFDZ3NIMICU5g2FVLUD7L
  69EUI8TYzfkyUhsckXjoJgDwj1CBeBNNOFEh3Jy+V8hQzrVFCW0Ab80i1eBYV/hIBm
  fsHSLP1E80hMPCmijONfTQB6g5Hoh/KE6N2fp6aSnqL/KA3+g3i8gk1hXHuGcJ
  RFeMeKdJqiW5xdqPTrw+rnRr5ee6Tzg606kr265NTDIAU8p8fQNuLjz49MMA+QwD
  BJTXwboQ0yRtb6X6q0mYaszUBbw==
Received: by mail-yk0-f179.google.com with SMTP id 19so2728865ykq.10
  for &lt;mailbox@gmail.com&gt;; Thu, 14 Jan 2015 15:02:45 -0800 (PST)
Authentication-Results: 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;
Hey gang,

This is a test message.

--J.
A.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 Auth-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 - 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 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: &lt;jqd@d1.example&gt;
Received: from mail-ob0-f188.google.com (mail-ob0-f188.google.com [208.69.40.157]) by clothilde.example.com with ESMTP id d200mr2263000ykb.93.1421363268 for &lt;fmartin@example.com&gt;; Thu, 14 Jan 2015 15:03:15 -0800 (PST)
Authentication-Results: clothilde.example.com; spf=fail
header.i=@gmail.com; dmarc=fail; arc=pass
ARC-Seal: i=2; a=rsa-sha256; t=1421363253;
s=notary01; d=example.com; k=20120806;
b=sjHDMrRZ0Mu15veVEOGscRHWbQHCY97lvruQhQ8+42CfIrxxUIk0E44x3LQwDW
RYbDj55cfC9MdcIahC+cP59BQ9Y9DiWMDzWRTm57NVb4KY+5sVnJo0aP9IF/s
uttxo+RRRn0fCFw==
ARC-Message-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
h=mime-version:date:message-id:subject:from:to:content-type
:x-original-sender:x-original-authentication-results:precedence
:mailing-list:list-id:list-post:list-help:list-archive:sender
:unsubscribe:reply-to;
bh=2+g22wzHUKz7V7JbpoOZMTRu199WvhcA4JnjihFm9Z2/g=;
b=pCw3Qxgfs9EIqnyN2+ITT3KhAgJwZz++Rju0BceSiuw1g0Pkk+3RZh/kaiz6
1TX6rVT6E4gs49Stkp4Ik7mujiR56Q611ahLkQJY/YFD3INiCMU52gFWLUD7L
69EU8TzyfBhUcsgxJjOjGwhrjCeBNNOf3Jy+V8h2rVFfCwOAbEo1iebYV/hIBm
fSfSLPfE80hMPCmiJOnfTQB6g5ho/KE6N2fpg6aSnqL/WA3+q31d8E1xXhVigCj
RFemKdJqiw5cxEdqTFRW+BnR5e67zg06kr265NTDIAn8p8fQNuLFzj49MA+QwD
BJmXwBOo2yRtb6X6Gq0mYiszUB8kw==
Received: by mail-yk0-f179.google.com with SMTP id 19so2728865ykg.10
for &lt;mailbox@gmail.com&gt;; Thu, 14 Jan 2015 15:02:45 -0800 (PST)
Authentication-Results: google.com; spf=fail
header.i=@example.org; dmarc=fail; arc=pass
ARC-Seal: i=2; a=rsa-sha256; t=1421363107;
```

Hey gang,

This is a test message.

--J.
Appendix B. Acknowledgements

This draft is the work of OAR-Dev Group.

The authors thank all of the OAR-Dev group for the ongoing help and thought-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.

Appendix C. Comments and Feedback

Please address all comments, discussions, and questions to arc-discuss@dmarc.org.

Appendix D. Historical Note

The ARC-Authentication-Results header is a direct copy of the normal Authentication-Results header ([RFC7601]) used in a similar fashion as that proposed in [OAR].

Authors’ Addresses

OAR-DEV Group
Email: arc-discuss@dmarc.org

Kurt Andersen
LinkedIn
2029 Stierlin Ct.
Mountain View, California  94043
USA

Email: kurta@linkedin.com

John Rae-Grant (editor)
Google

Email: johnrg@google.com

Brandon Long (editor)
Google

Email: blong@google.com