DMARC Working Group   K. Andersen
Internet-Draft   LinkedIn
Intended status: Experimental   S. Blank, Ed.
Expires: September 8, 2019   ValiMail
                                                          J. Levine, Ed.
                                                    Taughannock Networks
                                                          March 7, 2019

Using Multiple Signing Algorithms with the ARC (Authenticated Received Chain) Protocol
draft-ietf-dmarc-arc-multi-03

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.

   Initial development of ARC has been done with a single allowed
   signing algorithm, but parallel work in the DCRUP working group
   (https://datatracker.ietf.org/wg/dcrup/about/) is expanding the
   supported algorithms. This specification defines how to extend ARC
   for multiple signing algorithms.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at https://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on September 8, 2019.

Copyright Notice

Table of Contents

1. Introduction

The Authenticated Received Chain (ARC) protocol adds a traceable
chain of signatures that cover the handling of an email message
through a chain of intermediary handlers.

Initial development of ARC has been done with a single allowed
signing algorithm, but parallel work in the DCRUP working group
(https://datatracker.ietf.org/wg/dcrup/about/) is expanding the
supported algorithms. This specification defines how to extend ARC
for multiple signing algorithms.

2. Overview

In order to phase in new signing algorithms, this specification
identifies how signers and validators process ARC sets found in email
messages.

3. Definitions and Terminology

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

The capitalized 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 and the problems [RFC7960] created by the
initial DMARC [RFC7489] .

4. Supporting Alternate Signing Algorithms

During a period where multiple algorithms are allowed, all of the
statements in the ARC 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 set per instance per algorithm".

5. General Approach

5.1. Signers

There is a separate independent signing chain for each signing
algorithm. Hence, when creating an ARC signature, a signer MUST
include only other signatures that use the same algorithm as the
signature being created.

Wnen signing a message with no previous ARC signatures, signers MUST
sign using all supported algorithms.

A signer MUST continue the longest ARC chain(s) in a message with all
algorithms that it supports. That is, if at least one of the longest
chains uses an algorithm that a signer supports, the signer continues
the chain(s). If none of the longest chains in a message use an
algorithm supported by a signer, the signer MUST NOT extend any
chains, even if a shorter chain does use a supported algorithm.

5.2. Validators

A validator MUST use the longest ARC chain(s) on the message. If a
validator cannot interpret the signing algorithm on any of the
longest chains, validation fails, evven if a shorter chain does use a
supported algorithm.

If there is more than one longest chain, the overall result reported
can be that of of any of the validations. The result used when
extending an ARC chain MUST be the result from validating that chain.

6. Phases of Algorithm Evolution

6.1. 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.

6.2. 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.

6.3. 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.

6.4. 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.

7. Privacy Considerations

No unique privacy considerations are introduced by this specification
beyond those of the base [ARC-DRAFT-23] protocol.

8. IANA Considerations

   No new IANA considerations are introduced by this specification.

9. Security Considerations

   No new security considerations are introduced by this specification
   beyond those of the base [ARC-DRAFT-23] protocol.

10. References

10.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

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

10.2. Informative References

   [ARC-DRAFT-23]
              Andersen, K., Long, B., and S. Jones, "Authenticated
              Received Chain (ARC) Protocol (I-D-23)", n.d.,
              <https://tools.ietf.org/html/
              draft-ietf-dmarc-arc-protocol-23>.

   [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>.

10.3. URIs

   [1] mailto:dmarc@ietf.org

Appendix A. Acknowledgements

   This draft is the work of DMARC Working Group.

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

Appendix B. Comments and Feedback

   Please address all comments, discussions, and questions to
   dmarc@ietf.org [1].

Authors' Addresses

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

   Email: kurta@linkedin.com

   Seth Blank (editor)
   ValiMail
   Montgomery
   San Francisco, California
   US

   Email: seth@valimail.com

   John Levine (editor)
   Taughannock Networks
   PO Box 727
   Trumansburg, New York
   US

   Email: standards@taugh.com