Individual submission M. Kucherawy
Internet-Draft March 12, 2013
Obsoletes: 5451, 6577
(if approved)
Intended status: Standards Track
Expires: September 13, 2013

Message Header Field for Indicating Message Authentication Status
draft-kucherawy-rfc5451bis-02

Abstract

This document specifies a header field for use with electronic mail messages to indicate the results of message authentication efforts. Any receiver-side software, such as mail filters or Mail User Agents (MUAs), can use this header field to relay that information in a convenient and meaningful way to users, or make sorting and filtering decisions.

This document is a candidate for Internet Standard status. [RFC Editor: Please delete this notation, as I imagine it will be indicated some other way.]

Table of Contents

1. Introduction

[AR-ORIG] defined a new header field for electronic mail messages that presents the results of a message authentication effort in a machine-readable format. This document revises that definition based on current use of various authentication protocols in use today and incorporates errate logged since the publication of the original specification.

The intent of the header field is to create a place to collect such data when message authentication mechanisms are in use so that a Mail User Agent (MUA) and downstream filters can make filtering decisions and/or provide a recommendation to the user as to the validity of the message's origin and possibly the safety and integrity of its content.

End users are not expected to be direct consumers of this header field. This header field is intended for consumption by programs that will then use or render such data in a human-usable form.

This document specifies both the format of this header field and discusses the implications of its presence or absence. However, it does not discuss how the data contained in the header field ought to be used (i.e. what filtering decisions are appropriate, or how an MUA might render those results) as these are local policy and/or user interface design questions that are not appropriate for this document.

At the time of publication of this document, [ADSP], [AUTH], [DKIM], [SPF], and [VBR] are published DNS domain-level email authentication methods in common use. [DOMAINKEYS] is also referenced here, though it has "Historic" status as it is no longer common. This proposal is not intended to be restricted to domain-based authentication, but this has proven to be a good starting point for implementations. As various methods emerge, it is necessary to prepare for their appearance and encourage convergence in the area of interfacing verifiers to filters and MUAs.

Although [SPF] defined a header field called Received-SPF and [DOMAINKEYS] defined one called DomainKey-Status for this purpose, those header fields are specific to the conveyance of their respective results only and thus are insufficient to satisfy the requirements enumerated below. In addition, many SPF implementations have adopted the header field specified below, and DomainKeys has been obsoleted by DKIM.

1.1. Purpose

The header field defined in this document is expected to serve several purposes:

1. Convey the results of various message authentication checks being applied by upstream filters and Mail Transfer Agents (MTAs) to MUAs and downstream filters within the same "trust domain", as such agents may wish to render those results to end users or use that data to apply more or less stringent content checks based on authentication results;

2. Provide a common location within a message for this data;

3. Create an extensible framework for reporting new authentication methods as they emerge.

In particular, the mere presence of this header field should not be construed as meaning that its data is valid, but rather that it is asserting some kind of validity based on one or more authentication schemes applied somewhere upstream. For an MUA or downstream filter to treat the assertions as actually valid, there must be an assessment of the trust relationship between such agents and the validating MTA.

1.2. Trust Boundary

This document makes several references to the "trust boundary" of an administrative management domain (ADMD). Given the diversity among existing mail environments, a precise definition of this term isn't possible.

Simply put, a transfer from the creator of the header field to the consumer must occur within a context of trust that the creator's information is correct. How this trust is obtained is outside the scope of this document. It is entirely a local matter.

Thus, this document defines a "trust boundary" as the delineation between "external" and "internal" entities; "external" here includes all hosts that do not deliberately provide some kind of messaging service for the receiving ADMD's users, and "internal" includes those hosts that do. By this definition, the hosts within a "trust boundary" may lie entirely within a receiving ADMD's direct control, or they can include hosts managed by another ADMD (such as an ISP or commercial filtering service) but that also provide services for the former.

1.3. Processing Scope

This specification is intended to address the needs of authenticating messages or properties of messages during their actual transport. It is not meant to address the security of messages that might be encapsulated within other messages, such as a message/rfc822 [MIME] part within a message.

1.4. Requirements

This document establishes no new requirements on existing protocols or servers.

In particular, this document establishes no requirement on MTAs to reject or filter arriving messages that do not pass authentication checks. The data conveyed by the specified header field's contents are for the information of MUAs and filters and are to be used at their discretion.

1.5. Definitions

This section defines various terms used throughout this document.

1.5.1. Key Words

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 [KEYWORDS].

1.5.2. Security

[SECURITY] discusses authentication and authorization and the conflation of the two concepts. The use of those terms within the context of recent message security work has given rise to slightly different definitions, and this document reflects those current usages, as follows:

- "Authorization" is the establishment of permission to use a resource or represent an identity. In this context, authorization indicates that a message from a particular ADMD arrived via a route the ADMD has explicitly approved.

- "Authentication" is the assertion of validity of a piece of data about a message (such as the sender's identity) or the message in its entirety.

As examples: [SPF] and [SENDERID] are authorization mechanisms in that they express a result that shows whether or not the ADMD that

apparently sent the message has explicitly authorized the connecting [SMTP] client to relay messages on its behalf, but do not actually validate any property of the message itself. By contrast, [DKIM] is agnostic as to the routing of a message but uses cryptographic signatures to authenticate agents claiming responsibility for the message (which implies authorization) and ensure it was not modified in transit. Since the signatures are not tied to SMTP connections, they can be added by either the ADMD of origin, intermediate ADMDs (such as a mailing list server), or both.

Rather than create a separate header field for each class of solution, this proposal groups them both into a single header field.

1.5.3. Email Architecture

- A "border MTA" is an MTA that acts as a gateway between the general Internet and the users within an organizational boundary. (See also Section 1.2.)

- A "delivery MTA" (or Mail Delivery Agent or MDA) is an MTA that actually enacts delivery of a message to a user's inbox or other final delivery.

- An "intermediate MTA" is an MTA that handles messages after a border MTA and before a delivery MTA.

The following diagram illustrates the flow of mail among these defined components:

```
                       +-----+   +-----+   +------------+
                       | MUA |-->| MSA |-->| Border MTA |
                       +-----+   +-----+   +------------+
                                                 |
                                                 |
                                                 V
                                            +----------+
                                            | Internet |
                                            +----------+
                                                 |
                                                 |
                                                 V
+-----+   +-----+   +-------------------+   +------------+
| MUA |<--| MDA |<--| Intermediate MTA |<--| Border MTA |
+-----+   +-----+   +-------------------+   +------------+
```

Generally, it is assumed that the work of applying message authentication schemes takes place at a border MTA or a delivery MTA. This specification is written with that assumption in mind. However,

there are some sites at which the entire mail infrastructure consists of a single host. In such cases, such terms as "border MTA" and "delivery MTA" might well apply to the same machine or even the very same agent. It is also possible that some message authentication tests could take place on an intermediate MTA. Although this document doesn't specifically describe such cases, they are not meant to be excluded.

See [EMAIL-ARCH] for further discussion on general email system architecture, and Appendix D of this document for discussion about the common aspects of email authentication in current environments.

1.6. Trust Environment

This header field permits one or more message validation mechanisms to communicate output to one or more separate assessment mechanisms. These mechanisms operate within a unified trust boundary that defines an Administrative Management Domain (ADMD). An ADMD contains one or more entities that perform validation and generate the header field, and one or more that consume it for some type of assessment. The field contains no integrity or validation mechanism of its own, so its presence must be trusted implicitly. Hence, use of the header field depends upon ensuring that mail entering the ADMD has instances of the header field claiming to be valid within its boundaries removed, so that occurrences of such header fields can be used safely by consumers.

The "authserv-id" token defined in Section 2.2 can be used to label an entire ADMD or a specific validation engine within an ADMD. Although the labeling scheme is left as an operational choice, some guidance for selecting a token is provided in later sections of this document.

2. Definition and Format of the Header Field

This section gives a general overview of the format of the header field being defined, and then provides more formal specification.

2.1. General Description

The header field specified here is called "Authentication-Results". It is a Structured Header Field as defined in [MAIL] and thus all of the related definitions in that document apply.

This new header field SHOULD be 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 therefore should also be treated as a Trace Field as defined in [MAIL], and thus all of the

related definitions in that document apply.

The value of the header field (after removing [MAIL] comments) consists of an authentication identifier, an optional version, and then a series of "method=result" statements indicating which authentication method(s) were applied and their respective results, and then, for each applied method, an optional "reason" string, plus optional "property=value" statements indicating which message properties were evaluated to reach that conclusion.

The header field MAY appear more than once in a single message, or more than one result MAY be represented in a single header field, or a combination of these MAY be applied.

2.2. Formal Definition

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

```
authres-header = "Authentication-Results:" [CFWS] authserv-id
         [ CFWS version ]
         ( [CFWS] ";" [CFWS] "none" / 1*resinfo ) [CFWS] CRLF
       ; the special case of "none" is used to indicate that no
       ; message authentication is performed

authserv-id = dot-atom
            ; see below for a description of this element

version = 1*DIGIT [CFWS]
        ; indicates which version of this specification is in use;
        ; this specification is version "1", and the absence of a
        ; version implies this version of the specification

resinfo = [CFWS] ";" methodspec [ CFWS reasonspec ]
          *( CFWS propspec )

methodspec = [CFWS] method [CFWS] "=" [CFWS] result
           ; indicates which authentication method was evaluated
           ; and what its output was

reasonspec = "reason" [CFWS] "=" [CFWS] value
           ; a free-form comment on the reason the given result
           ; was returned

propspec = ptype [CFWS] "." [CFWS] property [CFWS] "=" pvalue
         ; an indication of which properties of the message
         ; were evaluated by the authentication scheme being
         ; applied to yield the reported result, and would be
         ; useful to reveal to end users as authenticated
```

```
     method = dot-atom [ [CFWS] "/" [CFWS] version ]
            ; a method indicates which method's result is
            ; represented by "result", and is one of the methods
            ; explicitly defined as valid in this document
            ; or is an extension method as defined below

     result = dot-atom
            ; indicates the results of the attempt to authenticate
            ; the message; see below for details

     ptype = "smtp" / "header" / "body" / "policy"
           ; indicates whether the property being evaluated was
           ; a parameter to an [SMTP] command, or was a value taken
           ; from a message header field, or was some property of
           ; the message body, or some other property evaluated by
           ; the receiving MTA

     property = dot-atom
              ; if "ptype" is "smtp", this indicates which [SMTP]
              ; command provided the value that was evaluated by the
              ; authentication scheme being applied; if "ptype" is
              ; "header", this indicates from which header field the
              ; value being evaluated was extracted; if "ptype" is
              ; "body", this indicates where in the message body
              ; a value being evaluated can be found (e.g., a specific
              ; offset into the message or a MIME part);
              ; if "ptype" is "policy" then this indicates the name
              ; of the policy that caused this header field to be
              ; added (see below)

     pvalue = [CFWS] ( value / [ [ local-part ] "@" ] domain-name )
              [CFWS]
            ; the value extracted from the message property defined
            ; by the "ptype.property" construction; if the value
            ; identifies something intended to be an e-mail identity,
            ; then it MUST use the right hand portion of this ABNF
            ; definition
```

The "local-part" is as defined in Section 3.4.1, and "dot-atom" is defined in Section 3.2.3, of [MAIL].

The "value" is as defined in Section 5.1 of [MIME].

The "domain-name" is as defined in Section 3.5 of [DKIM].

The "dot-atom" used in a "result" above is further constrained by the necessity of being enumerated in Section 2.4 or an amendment to it.

See Section 2.3 for a description of the "authserv-id" element.

The list of commands eligible for use with the "smtp" ptype can be found in [SMTP] and subsequent amendments.

"CFWS" is as defined in Section 3.2.2 of [MAIL].

The "propspec" may be omitted if, for example, the method was unable to extract any properties to do its evaluation yet has a result to report.

The "ptype" and "property" values used by each authentication method should be defined in the specification for that method (or its amendments).

The "ptype" and "property" are case-insensitive.

A "ptype" value of "policy" indicates a policy decision about the message not specific to a property of the message that could be extracted. For example, if a method would normally report a "ptype.property" of "header.From" and no From: header field was present, the method can use "policy" to indicate that no conclusion about the authenticity of the message could be reached.

2.3. Authentication Identifier Field

Every Authentication-Results header field has an authentication service identifier field ("authserv-id" above). This is similar in syntax to a fully-qualified domain name.

The authentication service identifier field provides a unique identifier that refers to the authenticating service within a given ADMD. The uniqueness of the identifier MUST be guaranteed by the ADMD that generates it and MUST pertain to exactly that one ADMD. This identifier is intended to be machine-readable and not necessarily meaningful to users. MUAs or downstream filters SHOULD use this identifier to determine whether or not the data contained in an Authentication-Results header field ought to be used or ignored.

For simplicity and scalability, the authentication service identifier SHOULD be a common token used throughout the ADMD, such as the DNS domain name used by or within that ADMD.

For tracing and debugging purposes, the authentication identifier MAY instead be the hostname of the MTA performing the authentication check whose result is being reported. This is also useful for another purpose, as described in Section 4. Moreover, some implementations have considered appending a delimiter such as "/" and

following it with useful transport tracing data such as the [SMTP] queue ID or a timestamp.

It should be noted, however, that using a local, relative identifier like a single hostname, rather than a hierarchical and globally unique ADMD identifier like a DNS domain name, makes configuration more difficult for large sites. The hierarchical identifier permits aggregating related, trusted systems together under a single, parent identifier, which in turn permits assessing the trust relationship with a single reference. The alternative is a flat namespace requiring individually listing each trusted system. Since consumers will use the identifier to determine whether to use the contents of the header field:

o Changes to the identifier impose a large, centralized administrative burden.

o Ongoing administrative changes require constantly updating this centralized table, making it difficult to ensure that an MUA or downstream filter will have access to accurate information for assessing the usability of the header field's content. In particular, consumers of the header field will need to know not only the current identifier(s) in use, but previous ones as well to account for delivery latency or later re-assessment of the header field's contents.

Examples of valid authentication identifiers are "example.com", "mail.example.org", "ms1.newyork.example.com", and "example-auth".

2.4. Result Values

Each individual authentication method returns one of a set of specific result values. The subsections below define these results for the authentication methods specifically supported by this document, and verifiers SHOULD use these values as described below. New methods not specified in this document intended to be supported by the header field defined here MUST include a similar result table either in its defining document or in a supplementary one.

2.4.1. DKIM and DomainKeys Results

The result values used by [DKIM] and [DOMAINKEYS] are as follows:

none: The message was not signed.

pass: The message was signed, the signature or signatures were acceptable to the verifier, and the signature(s) passed verification tests.

fail: The message was signed and the signature or signatures were acceptable to the verifier, but they failed the verification test(s).

policy: The message was signed but the signature or signatures were not acceptable to the verifier.

neutral: The message was signed but the signature or signatures contained syntax errors or were not otherwise able to be processed. This result SHOULD also be used for other failures not covered elsewhere in this list.

temperror: The message could not be verified due to some error that is likely transient in nature, such as a temporary inability to retrieve a public key. A later attempt may produce a final result.

permerror: The message could not be verified due to some error that is unrecoverable, such as a required header field being absent. A later attempt is unlikely to produce a final result.

A signature is "acceptable to the verifier" if it passes local policy checks (or there are no specific local policy checks). For example, a verifier might require that the signature(s) on the message be added using the DNS domain present in the From: header field of the message, thus making third-party signatures unacceptable.

[DKIM] advises that if a message fails verification, it should be treated as an unsigned message. A report of "fail" here permits the receiver of the report to decide how to handle the failure. A report of "neutral" or "none" preempts that choice, ensuring the message will be treated as if it had not been signed.

2.4.2. SPF and Sender-ID Results

The result values are used by [SPF] and [SENDERID] as follows:

none: No policy records were published at the sender's DNS domain.

neutral: The sender's ADMD has asserted that it cannot or does not want to make a statement as to whether the sending IP address is authorized to send mail using the sender's DNS domain.

pass: The client is authorized by the sender's ADMD to inject or relay mail on behalf of the sender's DNS domain.

policy: The client is authorized to inject or relay mail on behalf of the sender's DNS domain according to the authentication method's algorithm, but local policy dictates that the result is unacceptable.

fail: This client is explicitly not authorized to inject or relay mail using the sender's DNS domain.

softfail: The sender's ADMD believes the client was not authorized to inject or relay mail using the sender's DNS domain, but is unwilling to make a strong assertion to that effect.

temperror: The message could not be verified due to some error that is likely transient in nature, such as a temporary inability to retrieve a policy record from DNS. A later attempt may produce a final result.

permerror: The message could not be verified due to some error that is unrecoverable, such as a required header field being absent or a syntax error in a retrieved DNS TXT record. A later attempt is unlikely to produce a final result.

The distinction between and interpretation of "none" and "neutral" under these methods is discussed further in [SPF].

The "policy" result would be returned if, for example, [SPF] returned as "pass" result, but a local policy check matches the sending DNS domain to one found in an explicit list of unacceptable DNS domains (e.g., spammers).

If the retrieved sender policies used to evaluate [SPF] and [SENDERID] do not contain explicit provisions for authenticating the local-part (see Section 3.4.1 of [MAIL]) of an address, the "pvalue" reported along with results for these mechanisms SHOULD NOT include the local-part.

2.4.3. "iprev" Results

The result values are used by the "iprev" method, defined in Section 3, are as follows:

pass: The DNS evaluation succeeded, i.e., the "reverse" and "forward" lookup results were returned and were in agreement.

fail: The DNS evaluation failed. In particular, the "reverse" and "forward" lookups each produced results but they were not in agreement, or the "forward" query completed but produced no result, e.g., a DNS RCODE of 3, commonly known as NXDOMAIN, or an RCODE of 0 (NOERROR) in a reply containing no answers, was returned.

temperror: The DNS evaluation could not be completed due to some error that is likely transient in nature, such as a temporary DNS error, e.g., a DNS RCODE of 2, commonly known as SERVFAIL, or other error condition resulted. A later attempt may produce a final result.

permerror: The DNS evaluation could not be completed because no PTR data are published for the connecting IP address, e.g., a DNS RCODE of 3, commonly known as NXDOMAIN, or an RCODE of 0 (NOERROR) in a reply containing no answers, was returned. This prevented completion of the evaluation. A later attempt is unlikely to produce a final result.

There is no "none" for this method since any TCP connection delivering email has an IP address associated with it, so some kind of evaluation will always be possible.

For discussion of the format of DNS replies, see [DNS].

2.4.4. SMTP AUTH Results

The result values are used by the [AUTH] method are as follows:

none: SMTP authentication was not attempted.

pass: The SMTP client had authenticated to the server reporting the result using the protocol described in [AUTH].

fail: The SMTP client had attempted to authenticate to the server using the protocol described in [AUTH] but was not successful, yet continued to send the message about which a result is being reported.

temperror: The SMTP client attempted to authenticate using the protocol described in [AUTH] but was not able to complete the attempt due to some error which is likely transient in nature, such as a temporary Lightweight Directory Access Protocol (LDAP) lookup error. A later attempt may produce a final result.

permerror: The SMTP client attempted to authenticate using the protocol described in [AUTH] but was not able to complete the attempt due to some error that is likely not transient in nature, such as a permanent LDAP lookup error. A later attempt is not likely produce a final result.

Note that an agent making use of the data provided by this header field SHOULD consider "fail" and "temperror" to be the synonymous in terms of message authentication, i.e., the client did not authenticate.

2.4.5. Extension Result Codes

Additional result codes (extension results) might be defined in the future by later revisions or extensions to this specification. Result codes MUST be registered with the Internet Assigned Numbers Authority (IANA) and preferably published in an RFC. See Section 6 for further details.

Extension results MUST only be used within ADMDs that have explicitly consented to use them. These results and the parameters associated with them are not formally documented. Therefore, they are subject to change at any time and not suitable for production use. Any MTA, MUA or downstream filter intended for production use SHOULD ignore or delete any Authentication-Results header field that includes an extension result.

2.5. Authentication Methods

This section defines the supported authentication methods and discusses the proper means for applying experimental and other extension methods.

2.5.1. Definition of Initial Methods

As they are currently existing specifications for message authentication, it is appropriate to define an authentication method identifier for each of [ADSP], [ATPS], [AUTH], [DKIM], [DOMAINKEYS], [SENDERID], [SPF], and [VBR]. Therefore, the authentication method identifiers "dkim-adsp", "dkim-atps", "auth", "dkim", "domainkeys", "sender-id", "spf", and "vbr", respectively are defined for MTAs applying those specifications for email message authentication.

Furthermore, method "iprev" is defined in Section 3.

See Section 6 for details.

2.5.2. Extension Methods

Additional authentication method identifiers (extension methods) may be defined in the future by later revisions or extensions to this specification. Method identifiers MUST be registered with the Internet Assigned Numbers Authority (IANA) and, preferably, published in an RFC. See Section 6 for further details.

Extension methods can be defined for the following reasons:

1. To allow additional information from new authentication systems to be communicated to MUAs or downstream filters. The names of such identifiers should reflect the name of the method being defined, but should not be needlessly long.

2. To allow the creation of "sub-identifiers" that indicate different levels of authentication and differentiate between their relative strengths, e.g., "auth1-weak" and "auth1-strong".

Authentication method implementers are encouraged to provide adequate information, via [MAIL] comments if necessary, to allow an MUA developer to understand or relay anciliary details of authentication results. For example, if it might be of interest to relay what data was used to perform an evaluation, such information could be relayed as a comment in the header field, such as:

```
     Authentication-Results: example.com;
               foo=pass bar.baz=blob (2 of 3 tests OK)
```

Experimental method identifiers MUST only be used within ADMDs that have explicitly consented to use them. These method identifiers and the parameters associated with them are not documented in RFCs. Therefore, they are subject to change at any time and not suitable for production use. Any MTA, MUA, or downstream filter intended for production use SHOULD ignore or delete any Authentication-Results header field that includes an experimental (unknown) method identifier.

3. The "iprev" Authentication Method

This section defines an additional authentication method called "iprev".

In general, "iprev" is an attempt to verify that a client appears to be valid based on some DNS queries. Upon receiving a session initiation of some kind from a client, the IP address of the client peer is queried for matching names (i.e., a number-to-name translation, also known as a "reverse lookup" or a "PTR" record

query). Once that result is acquired, a lookup of each of the names (i.e., a name-to-number translation, or an "A" or "AAAA" record query) thus retrieved is done. The response to this second check should result in at least one mapping back to the client's IP address.

Expressed as an algorithm: If the client peer's IP address is I, the list of names to which I maps (after a "PTR" query) is the set N, and the union of IP addresses to which each member of N maps (after corresponding "A" and "AAAA" queries) is L, then this test is successful if I is an element of L.

The response to a PTR query could contain multiple names. To prevent heavy DNS loads, agents performing these queries MUST be implemented such that the number of names evaluated by generation of corresponding A or AAAA queries is finite, though it MAY be configurable by an administrator. As an example, Section 5.5 of [SPF] chose a limit of 10 for its implementation of this algorithm.

[DNS-IP6] discusses the query formats for the IPv6 case.

A successful test using this algorithm constitutes a result of "pass" since the ADMD in which the client's PTR claims it belongs has confirmed that claim by including corresponding data in its DNS domain. A failure to match constitutes a "fail". There is no case in which a "neutral" result can be returned. The remaining "temperror" and "permerror" cases refer, respectively, to temporary and permanent DNS query errors.

There is some contention regarding the wisdom and reliability of this test. For example, in some regions it can be difficult for this test ever to pass because the practice of arranging to match the forward and reverse DNS is infrequently observed. Therefore, the precise implementation details of how a verifier performs an "iprev" test are not specified here. The verifier MAY report a successful or failed "iprev" test at its discretion having done some kind of check of the validity of the connection's identity using DNS. It is incumbent upon an agent making use of the reported "iprev" result to understand what exactly that particular verifier is attempting to report.

Extensive discussion of reverse DNS mapping and its implications can be found in [DNSOP-REVERSE]. In particular, it recommends that applications avoid using this test as a means of authentication or security. Its presence in this document is not an endorsement, but is merely acknowledgement that the method remains common and provides the means to relay the results of that test.

4. Adding the Header Field to A Message

This specification makes no attempt to evaluate the relative strengths of various message authentication methods that may become available. As such, the order of the presented authentication methods and results MUST NOT be used either to imply or infer the importance or strength of any given method over another. Instead, the MUA or downstream filter consuming this header field is to interpret the result of each method based on its own knowledge of what that method evaluates.

Each "method" MUST refer to an authentication method declared in the IANA registry, or an extension method as described in Section 2.5.2, and each "result" MUST refer to a result code declared in the IANA registry, or an extension result code as defined in Section 2.4.5. See Section 6 for further information about the registered methods and result codes.

An MTA compliant with this specification MUST add this header field (after performing one or more message authentication tests) to indicate which MTA or ADMD performed the test, which test got applied and what the result was. If an MTA applies more than one such test, it MUST add this header field either once per test, or once indicating all of the results. An MTA MUST NOT add a result to an existing header field.

An MTA MAY add this header field containing only the authentication identifier portion and the "none" token (see Section 2.2) to indicate explicitly that no message authentication schemes were applied prior to delivery of this message.

An MTA adding this header field MUST take steps to identify it as legitimate to the MUAs or downstream filters that will ultimately consume its content. One REQUIRED process to do so is described in Section 5. Further measures may be necessary in some environments. Some possible solutions are enumerated in Section 7.1. This document does not mandate any specific solution to this issue as each environment has its own facilities and limitations.

For MTAs that add this header field, adding header fields in order (at the top), per Section 3.6 of [MAIL], is particularly important. Moreover, this header field SHOULD be inserted above any other trace header fields such MTAs might prepend. This allows easy detection of header fields that can be trusted.

End users making direct use of this header field might inadvertently trust information that has not been properly vetted. If, for example, a basic [SPF] result were to be relayed that claims an

authenticated addr-spec, the local-part of that addr-spec has actually not been authenticated. Thus, an MTA adding this header field SHOULD NOT include any data that has not been authenticated by the method(s) being applied. Moreover, MUAs SHOULD NOT render to users such information if it is presented by a method known not to authenticate it.

4.1. Header Field Position and Interpretation

In order to ensure non-ambiguous results and avoid the impact of false header fields, MUAs and downstream filters SHOULD NOT interpret this header field unless specifically instructed to do so by the user or administrator. That is, this interpretation should not be "on by default". Naturally then, users or administrators ought not activate such a feature unless they are certain the header field will be added by the border MTA that accepts the mail that is ultimately read by the MUA, and instances of the header field appearing to originate within the ADMD but are actually added by foreign MTAs will be removed before delivery.

Furthermore, MUAs and downstream filters SHOULD NOT interpret this header field unless the authentication service identifier it bears appears to be one used within its own ADMD as configured by the user or administrator.

MUAs and downstream filters MUST ignore any result reported using a "result" not specified in the result code registry, or a "ptype" not listed in the corresponding registry for such values as defined in Section 6. Moreover, such agents MUST ignore a result indicated for any "method" they do not specifically support.

An MUA SHOULD NOT reveal these results to end users unless the results are accompanied by, at a minimum, some associated reputation data about the authenticated origin identifiers within the message. For example, an attacker could register examp1e.com (note the digit "one") and send signed mail to intended victims; a verifier would detect that the signature was valid and report a "pass" even though it's clear the DNS domain name was intended to mislead. See Section 7.2 for further discussion.

As stated in Section 2.1, this header field SHOULD be treated as though it were a trace header field as defined in Section 3.6.7 of [MAIL], and hence MUST NOT be reordered and MUST be prepended to the message, so that there is generally some indication upon delivery of where in the chain of handling MTAs the message authentication was done.

MUAs SHOULD ignore instances of this header field discovered within

message/rfc822 [MIME] attachments.

Further discussion of this can be found in Section 7 below.

4.2. Local Policy Enforcement

If a site's local policy is to consider a non-recoverable failure result (e.g., "fail" for DKIM, "hardfail" for SPF) for any particular authentication method as justification to reject the message completely, the border MTA SHOULD issue an [SMTP] rejection response to the message rather than adding this header field with the failure result and allowing it to proceed toward delivery. This is more desirable than allowing the message to reach an internal host's MTA or spam filter, thus possibly generating a local rejection such as a [DSN] to a forged originator. Such generated rejections are colloquially known as "backscatter".

The same MAY also be done for local policy decisions overriding the results of the authentication methods (e.g., the "policy" result codes described in Section 2.4).

Such rejections at the SMTP protocol level are not possible if local policy is enforced at the MUA and not the MTA.

5. Removing the Header Field

For security reasons, any MTA conforming to this specification MUST delete any discovered instance of this header field that claims, by virtue of its authentication service identifier, to have been added within its trust boundary and that did not come directly from another trusted MTA. For example, an MTA (border or otherwise) for example.com receiving a message MUST delete any instance of this header field bearing an authentication service identifier indicating the header field was added within example.com prior to adding its own header fields. This could mean each MTA will have to be equipped with a list of internal MTAs known to be compliant (and hence trustworthy).

For simplicity and maximum security, a border MTA MAY remove all instances of this header field on mail crossing into its trust boundary. However, this may conflict with the desire to access authentication results performed by trusted external service providers. It may also invalidate signed messages whose signatures cover external instances of this header field. A more robust border MTA could allow a specific list of authenticating MTAs whose information should be admitted, removing all others.

As stated in Section 1.2, a formal definition of "trust boundary" is

deliberately not made here. It is entirely possible that a border MTA for example.com will explicitly trust authentication results asserted by upstream host example.net even though they exist in completely disjoint administrative boundaries. In that case, the border MTA MAY elect not to delete those results; moreover, the upstream host doing some authentication work could apply a signing technology such as [DKIM] on its own results to assure downstream hosts of their authenticity. An example of this is provided in Appendix C.

Similarly, in the case of messages signed using [DKIM] or other message signing methods that sign header fields, this removal action could invalidate one or more signatures on the message if they covered the header field to be removed. This behavior can be desirable since there's little value in validating the signature on a message with forged header fields. However, signing agents MAY therefore elect to omit these header fields from signing to avoid this situation.

An MTA SHOULD remove any instance of this header field bearing a version (express or implied) that it does not support. However, an MTA MUST remove such a header field if the [SMTP] connection relaying the message is not from a trusted internal MTA.

6. IANA Considerations

IANA has registered the defined header field and created two tables as described below. These registry actions were originally defined by [AR-ORIG] and are repeated here to provide a single, current reference.

6.1. The Authentication-Results Header Field

Per [IANA-HEADERS], the "Authentication-Results" header field has been added to the IANA Permanent Message Header Field Registry. The following is the updated registration template:

```
Header field name: Authentication-Results
Applicable protocol: mail ([MAIL])
Status: Standard
Author/Change controller: IETF
Specification document(s): [this memo]
Related information:
  Requesting review of any proposed changes and additions to
  this field is recommended.
```

6.2. Email Authentication Method Name Registry

Names of message authentication methods supported by this specification are to be registered with IANA, with the exception of experimental names as described in Section 2.5.2. A registry was created by [AR-ORIG] for this purpose. This document changes the registration rules governing this registry.

New entries are assigned only for values that have received Expert Review, per [IANA-CONSIDERATIONS]. The Designated Expert shall be appointed by the IESG. The Designated Expert has discretion to request that a publication be referenced if a clear, concise definition of the authentication method cannot be provided such that interoperability is assured. Registrations should otherwise be permitted.

Each method must register a name, the specification that defines it, and zero or more "ptype" values appropriate for use with that method, which "property" value(s) should be reported by that method, and a description of the "value" to be used with each.

All existing registry entries that reference [AR-ORIG] are to be updated to eference this document.

6.3. Email Authentication Result Name Registry

Names of message authentication result codes supported by this specification must be registered with IANA, with the exception of experimental codes as described in Section 2.4.5. A registry was created by [AR-ORIG] for this purpose. This document changes the registration rules governing this registry.

New entries are assigned only for values that have received Expert Review, per [IANA-CONSIDERATIONS]. The Designated Expert shall be appointed by the IESG. The Designated Expert has discretion to request that a publication be referenced if a clear, concise definition of the authentication result cannot be provided such that interoperability is assured. Registrations should otherwise be permitted.

All existing registry entries that reference [AR-ORIG] are to be updated to reference this document.

7. Security Considerations

The following security considerations apply when adding or processing the "Authentication-Results" header field:

7.1. Forged Header Fields

An MUA or filter that accesses a mailbox whose mail is handled by a non-conformant MTA, and understands Authentication-Results header fields, could potentially make false conclusions based on forged header fields. A malicious user or agent could forge a header field using the DNS domain of a receiving ADMD as the authserv-id token in the value of the header field, and with the rest of the value claim that the message was properly authenticated. The non-conformant MTA would fail to strip the forged header field, and the MUA could inappropriately trust it.

It is for this reason an MUA should not have processing of the "Authentication-Results" header field enabled by default; instead it should be ignored, at least for the purposes of enacting filtering decisions, unless specifically enabled by the user or administrator after verifying that the border MTA is compliant. It is acceptable to have an MUA aware of this specification, but have an explicit list of hostnames whose "Authentication-Results" header fields are trustworthy; however, this list should initially be empty.

Proposed alternate solutions to this problem are nascent:

1. Possibly the simplest is a digital signature protecting the header field, such as using [DKIM], that can be verified by an MUA by using a posted public key. Although one of the main purposes of this document is to relieve the burden of doing message authentication work at the MUA, this only requires that the MUA learn a single authentication scheme even if a number of them are in use at the border MTA. Note that [DKIM] requires that the From header field be signed, although in this application, the signing agent (a trusted MTA) likely cannot authenticate that value, so the fact that it is signed should be ignored.

2. Another would be a means to interrogate the MTA that added the header field to see if it is actually providing any message authentication services and saw the message in question, but this isn't especially palatable given the work required to craft and implement such a scheme.

3. Yet another might be a method to interrogate the internal MTAs that apparently handled the message (based on Received: header fields) to determine whether any of them conform to Section 5 of this memo. This, too, has potentially high barriers to entry.

4. Extensions to [IMAP], [SMTP], and [POP3] could be defined to allow an MUA or filtering agent to acquire the "authserv-id" in

use within an ADMD, thus allowing it to identify which Authentication-Results header fields it can trust.

5. On the presumption that internal MTAs are fully compliant with Section 3.6 of [MAIL], and the compliant internal MTAs are using their own host names or the ADMD's DNS domain name as the "authserv-id" token, the header field proposed here should always appear above a Received: header added by a trusted MTA. This can be used as a test for header field validity.

Support for some of these is being considered for future work.

In any case, a mechanism needs to exist for an MUA or filter to verify that the host that appears to have added the header field (a) actually did so, and (b) is legitimately adding that header field for this delivery. Given the variety of messaging environments deployed today, consensus appears to be that specifying a particular mechanism for doing so is not appropriate for this document.

Mitigation of the forged header field attack can also be accomplished by moving the authentication results data into meta-data associated with the message. In particular, an [SMTP] extension could be established that is used to communicate authentication results from the border MTA to intermediate and delivery MTAs; the latter of these could arrange to store the authentication results as meta-data retrieved and rendered along with the message by an [IMAP] client aware of a similar extension in that protocol. The delivery MTA would be told to trust data via this extension only from MTAs it trusts, and border MTAs would not accept data via this extension from any source. There is no vector in such an arrangement for forgery of authentication data by an outside agent.

7.2. Misleading Results

Until some form of service for querying the reputation of a sending agent is widely deployed, the existence of this header field indicating a "pass" does not render the message trustworthy. It is possible for an arriving piece of spam or other undesirable mail to pass checks by several of the methods enumerated above (e.g., a piece of spam signed using [DKIM] by the originator of the spam, which might be a spammer or a compromised system). In particular, this issue is not resolved by forged header field removal discussed above.

Hence, MUAs and downstream filters must take some care with use of this header even after possibly malicious headers are scrubbed.

7.3. Header Field Position

Despite the requirements of [MAIL], header fields can sometimes be reordered enroute by intermediate MTAs. The goal of requiring header field addition only at the top of a message is an acknowledgement that some MTAs do reorder header fields, but most do not. Thus, in the general case, there will be some indication of which MTAs (if any) handled the message after the addition of the header field defined here.

7.4. Reverse IP Query Denial-of-Service Attacks

Section 5.5 of [SPF] describes a DNS-based denial-of-service attack for verifiers that attempt DNS-based identity verification of arriving client connections. A verifier wishing to do this check and report this information need to take care not to go to unbounded lengths to resolve "A" and "PTR" queries. MUAs or other filters making use of an "iprev" result specified by this document need to be aware of the algorithm used by the verifier reporting the result and, especially, its limitations.

7.5. Mitigation of Backscatter

Failing to follow the instructions of Section 4.2 can result in a denial-of-service attack caused by the generation of [DSN] messages (or equivalent) to addresses that did not send the messages being rejected.

7.6. Internal MTA Lists

Section 5 describes a procedure for scrubbing header fields that may contain forged authentication results about a message. A compliant installation will have to include, at each MTA, a list of other MTAs known to be compliant and trustworthy. Failing to keep this list current as internal infrastructure changes may expose an ADMD to attack.

7.7. Attacks against Authentication Methods

If an attack becomes known against an authentication method, clearly then the agent verifying that method can be fooled into thinking an inauthentic message is authentic, and thus the value of this header field can be misleading. It follows that any attack against the authentication methods supported by this document (and later amendments to it) is also a security consideration here.

7.8. Intentionally Malformed Header Fields

It is possible for an attacker to add an Authentication-Results header field that is extraordinarily large or otherwise malformed in an attempt to discover or exploit weaknesses in header field parsing code. Implementers must thoroughly verify all such header fields received from MTAs and be robust against intentionally as well as unintentionally malformed header fields.

7.9. Compromised Internal Hosts

An internal MUA or MTA that has been compromised could generate mail with a forged From header field and a forged Authentication-Results header field that endorses it. Although it is clearly a larger concern to have compromised internal machines than it is to prove the value of this header field, this risk can be mitigated by arranging that internal MTAs will remove this header field if it claims to have been added by a trusted border MTA (as described above), yet the [SMTP] connection is not coming from an internal machine known to be running an authorized MTA. However, in such a configuration, legitimate MTAs will have to add this header field when legitimate internal-only messages are generated. This is also covered in Section 5.

7.10. Encapsulated Instances

[MIME] messages can contain attachments of type "message/rfc822", which contain other [MAIL] messages. Such an encapsulated message can also contain an Authentication-Results header field. Although the processing of these is outside of the intended scope of this document (see Section 1.3), some early guidance to MUA developers is appropriate here.

Since MTAs are unlikely to strip Authentication-Results header fields after mailbox delivery, MUAs are advised in Section 4.1 to ignore such instances within [MIME] attachments. Moreover, when extracting a message digest to separate mail store messages or other media, such header fields should be removed so that they will never be interpreted improperly by MUAs that might later consume them.

7.11. Reverse Mapping

Although Section 3 of this memo includes explicit support for the "iprev" method, its value as an authentication mechanism is limited. Implementers of both this proposal and agents that use the data it relays are encouraged to become familiar with the issues raised by [DNSOP-REVERSE] when deciding whether or not to include support for "iprev".

8. References

8.1. Normative References

[ABNF] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[AR-ORIG] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", RFC 5451, April 2009.

[IANA-HEADERS] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[MAIL] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[MIME] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

8.2. Informative References

[ADSP] Allman, E., Fenton, J., Delany, M., and J. Levine, "DomainKeys Identified Mail (DKIM) Author Domain Signing Practices (ADSP)", RFC 5617, August 2009.

[ATPS] Kucherawy, M., "DomainKeys Identified Mail (DKIM) Authorized Third-Party Signatures", RFC 6541, February 2012.

[AUTH] Siemborski, R. and A. Melnikov, "SMTP Service Extension for Authentication", RFC 4954, July 2007.

[DKIM] Crocker, D., Hansen, T., and M. Kucherawy, "DomainKeys Identified Mail (DKIM) Signatures", RFC 6376, September 2011.

[DNS] Mockapetris, P., "Domain names -

implementation and specification", STD 13, RFC 1035, November 1987.

[DNS-IP6] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[DNSOP-REVERSE] Senie, D. and A. Sullivan, "Considerations for the use of DNS Reverse Mapping", Work in Progress, March 2008.

[DOMAINKEYS] Delany, M., "Domain-Based Email Authentication Using Public Keys Advertised in the DNS (DomainKeys)", RFC 4870, May 2007.

[DSN] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003.

[EMAIL-ARCH] Crocker, D., "Internet Mail Architecture", Work in Progress, October 2008.

[IANA-CONSIDERATIONS] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[IMAP] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[POP3] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[SECURITY] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[SENDERID] Lyon, J. and M. Wong, "Sender ID: Authenticating E-Mail", RFC 4406, April 2006.

[SMTP] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[SPF] Wong, M. and W. Schlitt, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1", RFC 4408, April 2006.

[VBR] Hoffman, P., Levine, J., and A. Hathcock, "Vouch By Reference", RFC 5518, April 2009.

Appendix A. Acknowledgements

The author wishes to acknowledge the following for their review and constructive criticism of this update: S. Moonesamy, Alessandro Vesely, [other names]

Appendix B. Legacy MUAs

Implementers of this protocol should be aware that many MUAs are unlikely to be retrofitted to support the new header field and its semantics. In the interests of convenience and quicker adoption, a delivery MTA might want to consider adding things that are processed by existing MUAs in addition to the Authentication-Results header field. One suggestion is to include a Priority header field, on messages that don't already have such a header field, containing a value that reflects the strength of the authentication that was accomplished, e.g., "low" for weak or no authentication, "normal" or "high" for good or strong authentication.

Some modern MUAs can already filter based on the content of this header field. However, there is keen interest in having MUAs make some kind of graphical representation of this header field's meaning to end users. Until this capability is added, other interim means of conveying authentication results may be necessary while this proposal and its successors are adopted.

Appendix C. Authentication-Results Examples

This section presents some examples of the use of this header field to indicate authentication results.

C.1. Trivial Case; Header Field Not Present

The trivial case:

```
     Received: from mail-router.example.com
                   (mail-router.example.com [192.0.2.1])
               by server.example.org (8.11.6/8.11.6)
                   with ESMTP id g1G0r1kA003489;
               Fri, Feb 15 2002 17:19:07 -0800
     From: sender@example.com
     Date: Fri, Feb 15 2002 16:54:30 -0800
     To: receiver@example.org
     Message-Id: <12345.abc@example.com>
     Subject: here's a sample

     Hello!  Goodbye!
```

Example 1: Trivial case

The "Authentication-Results" header field is completely absent. The MUA may make no conclusion about the validity of the message. This could be the case because the message authentication services were not available at the time of delivery, or no service is provided, or the MTA is not in compliance with this specification.

C.2. Nearly Trivial Case; Service Provided, But No Authentication Done

A message that was delivered by an MTA that conforms to this specification but provides no actual message authentication service:

```
     Authentication-Results: example.org; none
     Received: from mail-router.example.com
                   (mail-router.example.com [192.0.2.1])
               by server.example.org (8.11.6/8.11.6)
                   with ESMTP id g1G0r1kA003489;
               Fri, Feb 15 2002 17:19:07 -0800
     From: sender@example.com
     Date: Fri, Feb 15 2002 16:54:30 -0800
     To: receiver@example.org
     Message-Id: <12345.abc@example.com>
     Subject: here's a sample

     Hello!  Goodbye!
```

Example 2: Header present but no authentication done

The "Authentication-Results" header field is present, showing that the delivering MTA conforms to this specification. It used its DNS

domain name as the authserv-id. The presence of "none" (and the absence of any method and result tokens) indicates that no message authentication was done.

C.3. Service Provided, Authentication Done

A message that was delivered by an MTA that conforms to this specification and applied some message authentication:

```
Authentication-Results: example.com;
          spf=pass smtp.mailfrom=example.net
Received: from dialup-1-2-3-4.example.net
              (dialup-1-2-3-4.example.net [192.0.2.200])
          by mail-router.example.com (8.11.6/8.11.6)
              with ESMTP id g1G0r1kA003489;
          Fri, Feb 15 2002 17:19:07 -0800
From: sender@example.net
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.com
Message-Id: <12345.abc@example.net>
Subject: here's a sample

Hello!  Goodbye!
```

Example 3: Header reporting results

The "Authentication-Results" header field is present, indicating that the border MTA conforms to this specification. The authserv-id is once again the DNS domain name. Furthermore, the message was authenticated by that MTA via the method specified in [SPF]. Note that since that method cannot authenticate the local-part, it has been omitted from the result's value. The MUA could extract and relay this extra information if desired.

C.4. Service Provided, Several Authentications Done, Single MTA

A message that was relayed inbound via a single MTA that conforms to this specification and applied three different message authentication checks:

```
Authentication-Results: example.com;
          auth=pass (cram-md5) smtp.auth=sender@example.com;
          spf=pass smtp.mailfrom=example.com
Authentication-Results: example.com;
          sender-id=pass header.from=example.com
Received: from dialup-1-2-3-4.example.net (8.11.6/8.11.6)
              (dialup-1-2-3-4.example.net [192.0.2.200])
          by mail-router.example.com (8.11.6/8.11.6)
              with ESMTP id g1G0r1kA003489;
          Fri, Feb 15 2002 17:19:07 -0800
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.net
From: sender@example.com
Message-Id: <12345.abc@example.com>
Subject: here's a sample

Hello!  Goodbye!
```

Example 4: Headers reporting results from one MTA

The "Authentication-Results" header field is present, indicating the delivering MTA conforms to this specification. Once again, the receiving DNS domain name is used as the authserv-id. Furthermore, the sender authenticated herself/himself to the MTA via a method specified in [AUTH], and both [SPF] and [SENDERID] checks were done and passed. The MUA could extract and relay this extra information if desired.

Two "Authentication-Results" header fields are not required since the same host did all of the checking. The authenticating agent could have consolidated all the results into one header field.

This example illustrates a scenario in which a remote user on a dialup connection (example.net) sends mail to a border MTA (example.com) using SMTP authentication to prove identity. The dialup provider has been explicitly authorized to relay mail as "example.com" resulting in passes by the SPF and SenderID checks.

C.5. Service Provided, Several Authentications Done, Different MTAs

A message that was relayed inbound by two different MTAs that conform to this specification and applied multiple message authentication checks:

```
Authentication-Results: example.com;
          sender-id=fail header.from=example.com;
          dkim=pass (good signature) header.i=sender@example.com
Received: from mail-router.example.com
              (mail-router.example.com [192.0.2.1])
          by auth-checker.example.com (8.11.6/8.11.6)
              with ESMTP id i7PK0sH7021929;
          Fri, Feb 15 2002 17:19:22 -0800
DKIM-Signature:  v=1; a=rsa-sha256; s=gatsby; d=example.com;
          t=1188964191; c=simple/simple; h=From:Date:To:Subject:
          Message-Id:Authentication-Results;
          bh=sEuZGD/pSr7ANysbY3jtdaQ3Xv9xPQtS0m70;
          b=EToRSuvUfQVP3Bkz ... rTB0t0gYnBVCM=
Authentication-Results: example.com;
          auth=pass (cram-md5) smtp.auth=sender@example.com;
          spf=fail smtp.mailfrom=example.com
Received: from dialup-1-2-3-4.example.net
              (dialup-1-2-3-4.example.net [192.0.2.200])
          by mail-router.example.com (8.11.6/8.11.6)
              with ESMTP id g1G0r1kA003489;
          Fri, Feb 15 2002 17:19:07 -0800
From: sender@example.com
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.com
Message-Id: <12345.abc@example.com>
Subject: here's a sample

Hello!  Goodbye!
```

Example 5: Headers reporting results from multiple MTAs

The "Authentication-Results" header field is present, indicating conformance to this specification. Once again, the authserv-id used is the recipient's DNS domain name. The header field is present twice because two different MTAs in the chain of delivery did authentication tests. The first, "mail-router.example.com" reports that [AUTH] and [SPF] were both used, and [AUTH] passed but [SPF] failed. In the [AUTH] case, additional data is provided in the comment field, which the MUA can choose to render if desired.

The second MTA, "auth-checker.example.com", reports that it did a [SENDERID] test (which failed) and a [DKIM] test (which passed).

Again, additional data about one of the tests is provided as a comment, which the MUA may choose to render. Also noteworthy here is the fact that there is a DKIM signature added by example.com that assured the integrity of the lower Authentication-Results field.

Since different hosts did the two sets of authentication checks, the header fields cannot be consolidated in this example.

This example illustrates more typical transmission of mail into "example.com" from a user on a dialup connection "example.net". The user appears to be legitimate as he/she had a valid password allowing authentication at the border MTA using [AUTH]. The [SPF] and [SENDERID] tests failed since "example.com" has not granted "example.net" authority to relay mail on its behalf. However, the [DKIM] test passed because the sending user had a private key matching one of "example.com"'s published public keys and used it to sign the message.

C.6. Service Provided, Multi-Tiered Authentication Done

A message that had authentication done at various stages, one of which was outside the receiving ADMD:

```
     Authentication-Results: example.com;
           dkim=pass reason="good signature"
             header.i=@mail-router.example.net;
           dkim=fail reason="bad signature"
             header.i=@newyork.example.com
     Received: from mail-router.example.net
               (mail-router.example.net [192.0.2.250])
           by chicago.example.com (8.11.6/8.11.6)
               for <recipient@chicago.example.com>
               with ESMTP id i7PK0sH7021929;
           Fri, Feb 15 2002 17:19:22 -0800
     DKIM-Signature: v=1; a=rsa-sha256; s=furble;
           d=mail-router.example.net; t=1188964198; c=relaxed/simple;
           h=From:Date:To:Message-Id:Subject:Authentication-Results;
           bh=ftA9J6GtX8OpwUECzHnCkRzKw1uk6FNiLfJl5Nmv49E=;
           b=oINEO8hgn/gnunsg ... 9n9ODSNFSDij3=
     Authentication-Results: example.net;
           dkim=pass (good signature) header.i=@newyork.example.com
     Received: from smtp.newyork.example.com
               (smtp.newyork.example.com [192.0.2.220])
           by mail-router.example.net (8.11.6/8.11.6)
               with ESMTP id g1G0r1kA003489;
           Fri, Feb 15 2002 17:19:07 -0800
     DKIM-Signature: v=1; a=rsa-sha256; s=gatsby;
           d=newyork.example.com;
           t=1188964191; c=simple/simple;
           h=From:Date:To:Message-Id:Subject;
           bh=sEu28nfs9fuZGD/pSr7ANysbY3jtdaQ3Xv9xPQtS0m7=;
           b=EToRSuvUfQVP3Bkz ... rTB0t0gYnBVCM=
     From: sender@newyork.example.com
     Date: Fri, Feb 15 2002 16:54:30 -0800
     To: meetings@example.net
     Message-Id: <12345.abc@newyork.example.com>
     Subject: here's a sample
```

Example 6: Headers reporting results from multiple MTAs in different ADMDs

In this example we see multi-tiered authentication with an extended trust boundary.

The message was sent from someone at example.com's New York office (newyork.example.com) to a mailing list managed at an intermediary.

The message was signed at the origin using [DKIM].

The message was sent to a mailing list service provider called example.net, which is used by example.com. There, meetings@example.net is expanded to a long list of recipients, one of that is at the Chicago office. In this example, we will assume that the trust boundary for chicago.example.com includes the mailing list server at example.net.

The mailing list server there first authenticated the message and affixed an Authentication-Results header field indicating such using its DNS domain name for the authserv-id. It then altered the message by affixing some footer text to the body, including some administrivia such as unsubscription instructions. Finally, the mailing list server affixes a second [DKIM] signature and begins distribution of the message.

The border MTA for chicago.example.com explicitly trusts results from mail-router.example.net so that header field is not removed. It performs evaluation of both signatures and determines that the first (most recent) is a "pass" but, because of the aforementioned modifications, the second is a "fail". However, the first signature included the Authentication-Results header added at mail-router.example.net that validated the second signature. Thus, indirectly, it can be determined that the authentications claimed by both signatures are indeed valid.

Note that two styles of presenting meta-data about the result are in use here. In one case, the "reason=" clause is present which is intended for easy extraction by parsers; in the other case, the CFWS production of the ABNF is used to include such data as a header field comment. The latter can be harder for parsers to extract given the varied supported syntaxes of mail header fields.

Appendix D. Operational Considerations about Message Authentication

This protocol is predicated on the idea that authentication (and presumably in the future, reputation) work is typically done by border MTAs rather than MUAs or intermediate MTAs; the latter merely make use of the results determined by the former. Certainly this is not mandatory for participation in electronic mail or message authentication, but this protocol and its deployment to date are based on that model. The assumption satisfies several common ADMD requirements:

1. Service operators prefer to resolve the handling of problem messages as close to the border of the ADMD as possible. This enables, for example, rejections of messages at the SMTP level

rather than generating a DSN internally. Thus, doing any of the authentication or reputation work exclusively at the MUA or intermediate MTA renders this desire unattainable.

2. Border MTAs are more likely to have direct access to external sources of authentication or reputation information since modern MUAs are more likely to be heavily firewalled. Thus, some MUAs might not even be able to complete the task of performing authentication or reputation evaluations without complex proxy configurations or similar burdens.

3. MUAs rely upon the upstream MTAs within their trust boundaries to make correct (as much as that is possible) evaluations about the message's envelope, header and content. Thus, MUAs don't need to know how to do the work that upstream MTAs do; they only need the results of that work.

4. Evaluations about the quality of a message, from simple token matching (e.g., a list of preferred DNS domains) to cryptanalysis (e.g., public/private key work), are at least a little bit expensive and thus should be minimized. To that end, performing those tests at the border MTA is far preferred to doing that work at each MUA that handles a message. If an ADMD's environment adheres to common messaging protocols, a reputation query or an authentication check performed by a border MTA would return the same result as the same query performed by an MUA. By contrast, in an environment where the MUA does the work, a message arriving for multiple recipients would thus cause authentication or reputation evaluation to be done more than once for the same message (i.e., at each MUA) causing needless amplification of resource use and creating a possible denial-of-service attack vector.

5. Minimizing change is good. As new authentication and reputation methods emerge, the list of methods supported by this header field would presumably be extended. If MUAs simply consume the contents of this header field rather than actually attempting to do authentication and/or reputation work, then MUAs only need to learn to parse this header field once; emergence of new methods requires only a configuration change at the MUAs and software changes at the MTAs (which are presumably fewer in number). When choosing to implement these functions in MTAs vs MUAs, the issues of individual flexibility, infrastructure inertia and scale of effort must be considered. It is typically easier to change a single MUA than an MTA because the modification affects fewer users and can be pursued with less care. However, changing many MUAs is more effort than changing a smaller number of MTAs.

6. For decisions affecting message delivery and display, assessment based on authentication and reputation is best performed close to the time of message transit, as a message makes its journey toward a user's inbox, not afterwards. DKIM keys and IP address reputations, etc., can change over time or even become invalid, and users can take a long time to read a message once delivered. The value of this work thus degrades, perhaps quickly, once the delivery process has completed. This seriously diminishes the value of this work when done other than at MTAs.

Many operational choices are possible within an ADMD, including the venue for performing authentication and/or reputation assessment. The current specification does not dictate any of those choices. Rather, it facilitates those cases in which information produced by one stage of analysis needs to be transported with the message to the next stage.

Appendix E. Changes since RFC5451

- Errata #2617 was addressed in RFC6577 and was incorporated here
- Request Internet Standard status
- Change IANA rules to Designated Expert from IETF Review
- Copy current IANA registry entries
- Add references to ADSP, ATPS, VBR
- Remove all the "X-" stuff, per BCP178
- Adjust language to indicate that this header field was already defined, and we're just refreshing and revising
- In a few places, RFC2119 language had been used in lowercase terms; fixed here
- Errata #2818 addressed
- Errata #3195 addressed
- Some minor wordsmithing

Author's Address

Murray S. Kucherawy
270 Upland Drive
San Francisco, CA 94127
US

EMail: superuser@gmail.com