ï»¿

Internet Engineering Task Force (IETF)
Request for Comments: 8823
Category: Informational
ISSN: 2070-1721

A. Melnikov
Isode Ltd

April 2021

Extensions to Automatic Certificate Management Environment for End-User S/MIME Certificates

Abstract

This document specifies identifiers and challenges required to enable the Automated Certificate Management Environment (ACME) to issue certificates for use by email users that want to use S/MIME.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8823.

Table of Contents

1. Introduction

ACME [RFC8555] is a mechanism for automating certificate management on the Internet. It enables administrative entities to prove effective control over resources like domain names, and it automates the process of generating and issuing certificates.

This document describes an extension to ACME for use by S/MIME. Section 3 defines extensions for issuing end-user S/MIME [RFC8550] certificates.

This document aims to support both:

1. A Mail User Agent (MUA) that has a built-in ACME client that is aware of the extension described in this document. (We will call such MUAs "ACME-email-aware".) Such an MUA can present a nice user interface to the user and automate certificate issuance.

2. An MUA that is not ACME aware, with a separate ACME client implemented in a command-line tool or as a part of a website. While S/MIME certificate issuance is not going to be as painless as in the case of the ACME-email-aware MUA, the extra burden on a user is going to be minimal.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Use of ACME for Issuing End-User S/MIME Certificates

ACME [RFC8555] defines a "dns" identifier type that is used to verify that a particular entity has control over a domain or specific service associated with the domain. In order to be able to issue end-user S/MIME certificates, ACME needs a new identifier type that proves ownership of an email address.

This document defines a new identifier type, "email", which corresponds to an email address. The address can be all ASCII [RFC5321] or internationalized [RFC6531]; when an internationalized email address is used, the domain part can contain both U-labels and A-labels [RFC5890]. This can be used with S/MIME or another similar service that requires possession of a certificate tied to an email address.

Any identifier of type "email" in a newOrder request MUST NOT have a wildcard ("*") character in its value.

A new challenge type, "email-reply-00", is used with the "email" identifier type, which provides proof that an ACME client has control over an email address.

The process of issuing an S/MIME certificate works as follows. Note that the ACME client can be a standalone application (if the MUA is not ACME-email-aware) or can be a component of the MUA.

1. An end user initiates issuance of an S/MIME certificate for one of their email addresses. This might be done by using an email client UI, by running a command-line tool, by visiting a certificate authority web page, etc. This document doesn't prescribe a specific UI used to initiate S/MIME certificate issuance or where the ACME client is located.

2. The ACME-email-aware client component begins the certificate issuance process by sending a POST request to the server's newOrder resource, including the identifier of type "email". See Section 7.4 of [RFC8555] for more details.

3. The ACME server responds to the POST request, including an

"authorizations" URL for the requested email address. The ACME client then retrieves information about the corresponding "email-reply-00" challenge, as specified in Section 7.5 of [RFC8555]. The "token" field of the corresponding challenge object (from the "challenges" array) contains token-part2. token-part2 should contain at least 128 bits of entropy. The "type" field of the challenge object is "email-reply-00". The challenge object also contains the "from" field, with the email address that would be used in the From header field of the "challenge" email message (see the next step).

An example challenge object might look like this:

```
{
  "type": "email-reply-00",
  "url": "https://example.com/acme/chall/ABprV_B7yEyA4f",
  "from": "acme-challenge+2i211oi1204310@example.com",
  "token": "DGyRejmCefe7v4NfDGDKfA"
}
```

4. After responding to the authorization request, the ACME server generates another token and a "challenge" email message with the subject "ACME: <token-part1>", where <token-part1> is the base64url-encoded [RFC4648] form of the token. The ACME server MUST generate a fresh token for each S/MIME issuance request (authorization request), and token-part1 MUST contain at least 128 bits of entropy. The "challenge" email message structure is described in more details in Section 3.1.

5. The MUA retrieves and parses the "challenge" email message. If the MUA is ACME-email-aware, it ignores any "challenge" email that is not expected, e.g., if there is no ACME certificate issuance pending. The ACME-email-aware MUA also ignores any "challenge" email that has the Subject header field that indicates that it is an email reply, e.g., a subject starting with the reply prefix "Re:".

6. The ACME client concatenates "token-part1" (received over email) and "token-part2" (received over HTTPS [RFC2818]) to create the ACME "token" and calculates keyAuthorization (as per Section 8.1 of [RFC8555]). Then, it returns the base64url-encoded SHA-256 digest [RFC6234] of the key authorization. The MUA returns the base64url-encoded SHA-256 digest obtained from the ACME client in the body of a "response" email message. The "response" email message structure is described in more details in Section 3.2. If the MUA is ACME-email-aware, it MUST NOT respond to the same "challenge" email more than once.

7. Once the MUA sends the "response" email, the ACME client notifies the ACME server by POST to the challenge URL ("url" field).

8. The ACME client can start polling the authorization URL (using POST-as-GET requests) to see if the ACME server received and validated the "response" email message. (See Section 7.5.1 of [RFC8555] for more details.) If the "status" field of the challenge switches to "valid", then the ACME client can proceed with request finalization. The Certificate Signing Request (CSR) MUST indicate the exact same set of requested identifiers as the initial newOrder request. For an identifier of type "email", the PKCS#10 [RFC2986] CSR MUST contain the requested email address in an extensionRequest attribute [RFC2985] requesting a subjectAltName extension. The email address MUST also match the From header field value of the "response" email message.

9. In order to request generation of signing-only or encryption-only S/MIME certificates (as opposed to requesting generation of S/MIME certificates suitable for both), the CSR needs to include the key usage extension (see Section 4.4.2 of [RFC8550]). This is described in more details in Section 3.3.

10. If a request to finalize an order is successful, the ACME server will return a 200 (OK) with an updated order object. If the certificate is issued successfully, i.e., if the order "status" is "valid", then the ACME client can download the issued S/MIME certificate from the URL specified in the "certificate" field.

3.1. ACME "Challenge" Email

A "challenge" email message MUST have the following structure:

1. The Subject header field has the following syntax: "ACME: <token-part1>", where the prefix "ACME:" is followed by folding white space (FWS; see [RFC5322]) and then by <token-part1>, which is the base64url-encoded first part of the ACME token that MUST be at least 128 bits long after decoding. Due to the recommended 78-octet line-length limit in [RFC5322], the subject line can be folded, so white spaces (if any) within the <token-part1> MUST be ignored. [RFC2231] encoding of the Subject header field MUST be supported, and, when used, only the "UTF-8" and "US-ASCII" charsets are allowed; other charsets MUST NOT be used. The US-ASCII charset SHOULD be used.

2. The From header field MUST be the same email address as specified in the "from" field of the challenge object.

3. The To header field MUST be the email address of the entity that requested the S/MIME certificate to be generated.

4. The message MAY contain a Reply-To and/or CC header field.

5. The message MUST include the Auto-Submitted header field with the value "auto-generated" [RFC3834]. To aid in debugging (and, for some implementations, to make automated processing easier), the Auto-Submitted header field SHOULD include the "type=acme" parameter. It MAY include other optional parameters, as allowed by the syntax of the Auto-Submitted header field.

6. In order to prove authenticity of a challenge message, it MUST be signed using either DomainKeys Identified Mail (DKIM) [RFC6376] or S/MIME [RFC8551].

   * If DKIM signing is used, the resulting DKIM-Signature header field MUST contain the "h=" tag that includes at least the From, Sender, Reply-To, To, CC, Subject, Date, In-Reply-To, References, Message-ID, Auto-Submitted, Content-Type, and Content-Transfer-Encoding header fields. The DKIM-Signature header field's "h=" tag SHOULD also include the Resent-Date, Resent-From, Resent-To, Resent-Cc, List-Id, List-Help, List-Unsubscribe, List-Subscribe, List-Post, List-Owner, List-Archive, and List-Unsubscribe-Post header fields. The domain from the "d=" tag of the DKIM-Signature header field MUST be the same as the domain from the From header field of the "challenge" email.

   * If S/MIME signing is used, the certificate corresponding to the signer MUST have an rfc822Name subjectAltName extension with the value equal to the From header field email address of the "challenge" email.

7. The body of the challenge message is not used for automated processing, so it can be any media type. (However, there are extra requirements on S/MIME signing, if used. See below.) Typically, it is text/plain or text/html containing a human-readable explanation of the purpose of the message. If S/MIME signing is used to prove authenticity of the challenge message, then the multipart/signed or "application/pkcs7-mime; smime-type=signed-data;" media type should be used. Either way, it MUST use S/MIME header protection.

An email client compliant with this specification that detects that a

particular "challenge" email fails the validation described above MUST ignore the challenge and thus will not generate a "response" email. To aid in debugging, such failed validations SHOULD be logged.

Here is an example of an ACME "challenge" email (note that, for simplicity, DKIM-related header fields are not included).

```
Auto-Submitted: auto-generated; type=acme
Date: Sat, 5 Dec 2020 10:08:55 +0100
Message-ID: <A2299BB.FF7788@example.org>
From: acme-generator@example.org
To: alexey@example.com
Subject: ACME: LgYemJLy3F1LDkiJrdIGbEzyFJyOyf6vBdyZ1TG3sME=
Content-Type: text/plain
MIME-Version: 1.0

This is an automatically generated ACME challenge for email address
"alexey@example.com". If you haven't requested an S/MIME
certificate generation for this email address, be very afraid.
If you did request it, your email client might be able to process
this request automatically, or you might have to paste the first
token part into an external program.
```

Figure 1

3.2. ACME "Response" Email

A valid "response" email message MUST have the following structure:

1. The Subject header field is formed as a reply to the ACME "challenge" email (see Section 3.1). Its syntax is the same as that of the challenge message except that it may be prefixed by a US-ASCII reply prefix (typically "Re:") and FWS (see [RFC5322]), as is normal in reply messages. When parsing the subject, ACME servers MUST decode [RFC2231] encoding (if any), and then they can ignore any prefix before the "ACME:" label.

2. The From header field contains the email address of the user that is requesting S/MIME certificate issuance.

3. The To header field of the response contains the value from the Reply-To header field from the challenge message (if set). Otherwise, it contains the value from the From header field of the challenge message.

4. The Cc header field is ignored if present in the "response" email message.

5. The In-Reply-To header field SHOULD be set to the Message-ID header field of the challenge message according to rules in Section 3.6.4 of [RFC5322].

6. List-* header fields [RFC4021][RFC8058] MUST be absent (i.e., the reply can't come from a mailing list).

7. The media type of the "response" email message is either text/plain or multipart/alternative [RFC2046], containing text/plain as one of the alternatives. (Note that the requirement to support multipart/alternative is to allow use of ACME-unaware MUAs, which can't always generate pure text/plain, e.g., if they reply to a text/html). The text/plain body part (whether or not it is inside multipart/alternative) MUST contain a block of lines starting with the line "-----BEGIN ACME RESPONSE-----", followed by one or more lines containing the base64url-encoded SHA-256 digest [RFC6234] of the key authorization, calculated from concatenated token-part1 (received over email) and token-part2 (received over HTTPS), as outlined in the 5th bullet in Section 3. (Note that each line of text/plain is terminated by

CRLF. Bare LFs or bare CRs are not allowed.) Due to historical line-length limitations in email, line endings (CRLFs) can be freely inserted in the middle of the encoded digest, so they MUST be ignored when processing it. The final line of the encoded digest is followed by a line containing:

-----END ACME RESPONSE-----

Any text before and after this block is ignored. For example, such text might explain what to do with it for ACME-unaware clients.

8. There is no need to use any Content-Transfer-Encoding other than 7bit for the text/plain body part. Use of quoted-printable or base64 in a "response" email message is not necessary and should be avoided, though it is permitted.

9. In order to prove authenticity of a response message, it MUST be DKIM [RFC6376] signed. The resulting DKIM-Signature header field MUST contain the "h=" tag that includes at least the From, Sender, Reply-To, To, CC, Subject, Date, In-Reply-To, References, Message-ID, Content-Type, and Content-Transfer-Encoding header fields. The DKIM-Signature header field's "h=" tag SHOULD also include the Resent-Date, Resent-From, Resent-To, Resent-Cc, List-Id, List-Help, List-Unsubscribe, List-Subscribe, List-Post, List-Owner, List-Archive, and List-Unsubscribe-Post header fields. The domain from the "d=" tag of DKIM-Signature header field MUST be the same as the domain from the From header field of the "response" email.

Here is an example of an ACME "response" email (note that, for simplicity, DKIM-related header fields are not included).

```
Date: Sat, 5 Dec 2020 12:01:45 +0100
Message-ID: <111-22222-3333333@example.com>
In-Reply-To: <A2299BB.FF7788@example.org>
From: alexey@example.com
To: acme-generator@example.org
Subject: Re: ACME: LgYemJLy3F1LDkiJrdIGbEzyFJyOyf6vBdyZ1TG3sME=
Content-Type: text/plain
MIME-Version: 1.0

-----BEGIN ACME RESPONSE-----
LoqXcYV8q5ONbJQxbmR7SCTNo3tiAXDfowy
jxAjEuX0=
-----END ACME RESPONSE-----
```

Figure 2

3.3. Generating Encryption-Only or Signing-Only S/MIME Certificates

ACME extensions specified in this document can be used to request signing-only or encryption-only S/MIME certificates.

In order to request signing-only S/MIME certificates, the CSR MUST include the key usage extension with digitalSignature and/or nonRepudiation bits set and no other bits set.

In order to request encryption-only S/MIME certificates, the CSR MUST include the key usage extension with keyEncipherment or keyAgreement bits set and no other bits set.

Presence of both of the above sets of key usage bits in the CSR, as well as absence of the key usage extension in the CSR, signals to the ACME server to issue an S/MIME certificate suitable for both signing and encryption.

4. Internationalization Considerations

[RFC8616] updated/clarified use of DKIM with internationalized email

addresses [RFC6531]. Please consult [RFC8616] in regards to any changes that need to be implemented.

Use of non-ASCII characters in left-hand sides of internationalized email addresses requires putting internationalized email addresses in X.509 certificates [RFC8398].

5. IANA Considerations

5.1. ACME Identifier Type

IANA has registered a new identifier type in the "ACME Identifier Types" registry defined in Section 9.7.7 of [RFC8555] with Label "email" and a Reference to this document, [RFC5321], and [RFC6531]. The new identifier type corresponds to an (all ASCII) email address [RFC5321] or internationalized email addresses [RFC6531].

5.2. ACME Challenge Type

IANA has registered a new entry in the "ACME Validation Methods" registry defined in Section 9.7.8 of [RFC8555]. This entry is as follows:

Label	Identifier Type	ACME	Reference
email-reply-00	email	Y	RFC 8823

Table 1

6. Security Considerations

Please see the Security Considerations section of [RFC8555] for general security considerations related to the use of ACME. This challenge/response protocol demonstrates that an entity that controls the private key (corresponding to the public key in the certificate) also controls the named email account. The ACME server is confirming that the requested email address belongs to the entity that requested the certificate, but this makes no claim to address correctness or fitness for purpose. If such claims are needed, they must be obtained by some other mechanism.

The security of the "email-reply-00" challenge type depends on the security of the email system. A third party that can read and reply to user's email messages (by possessing a user's password or a secret derived from it that can give read and reply access, such as "password equivalent" information, or by being given permissions to act on a user's behalf using email delegation features common in some email systems) can request S/MIME certificates using the protocol specified in this document and is indistinguishable from the email account owner. This has several possible implications:

1. An entity that compromised an email account would be able to request S/MIME certificates using the protocol specified in this document, and such entity couldn't be distinguished from the legitimate email account owner (unless some external sources of information are consulted).

2. For email addresses with legitimate shared access/control by multiple users, any such user would be able to request S/MIME certificates using the protocol specified in this document; such requests can't be attributed to a specific user without consulting external systems (such as IMAP/SMTP access logs).

3. The protocol specified in this document is not suitable for use with email addresses associated with mailing lists [RFC5321]. While it is not always possible to guarantee that a particular S/MIME certificate request is not from a mailing list address, prohibition on inclusion of List-* header fields helps certificate issuers to handle most common cases.

An email system in its turn depends on DNS. A third party that can manipulate DNS MX records for a domain might be able to redirect an email and can get (at least temporary) read and reply access to it. Similar considerations apply to DKIM TXT records in DNS. Use of DNSSEC by email system administrators is recommended to avoid making it easy to spoof DNS records affecting an email system. However, use of DNSSEC is not ubiquitous at the time of publishing of this document, so it is not required here. Also, many existing systems that rely on verification of ownership of an email address -- for example, 2-factor authentication systems used by banks or traditional certificate issuance systems -- send email messages to email addresses, expecting the owner to click on the link supplied in them (or to reply to a message), without requiring use of DNSSEC. So the risk of not requiring DNSSEC is presumed acceptable in this document.

An ACME email challenge message can be forged by an attacker. As per requirements on an ACME-email-aware MUA specified in Section 3, the MUA will not respond to requests it is not expecting. Even if the attacker causes the erroneous "response" email to go to an attacker-controlled email address, very little information is leaked -- the SHA-256 hash of the key authorization would be leaked, not the key authorization itself, so no parts of the token or the account key thumbprint are leaked.

An attacker that can read the "response" email has only one chance to guess the token-part2. Even if the attacker can guess it right, it still needs to know the ACME account key to be able to make use of the intercepted SHA-256 hash of the key authorization.

Also see the Security Considerations section of [RFC6376] for details on how DKIM depends on the DNS and the respective vulnerabilities this dependence has.

7. References

7.1. Normative References

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <https://www.rfc-editor.org/info/rfc2046>.

[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>.

[RFC2231] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, DOI 10.17487/RFC2231, November 1997, <https://www.rfc-editor.org/info/rfc2231>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>.

[RFC2985] Nystrom, M. and B. Kaliski, "PKCS #9: Selected Object Classes and Attribute Types Version 2.0", RFC 2985, DOI 10.17487/RFC2985, November 2000, <https://www.rfc-editor.org/info/rfc2985>.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, DOI 10.17487/RFC2986, November 2000, <https://www.rfc-editor.org/info/rfc2986>.

[RFC3834] Moore, K., "Recommendations for Automatic Responses to Electronic Mail", RFC 3834, DOI 10.17487/RFC3834, August 2004, <https://www.rfc-editor.org/info/rfc3834>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/info/rfc5321>.

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

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <https://www.rfc-editor.org/info/rfc5890>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <https://www.rfc-editor.org/info/rfc6376>.

[RFC6531] Yao, J. and W. Mao, "SMTP Extension for Internationalized Email", RFC 6531, DOI 10.17487/RFC6531, February 2012, <https://www.rfc-editor.org/info/rfc6531>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8398] Melnikov, A., Ed. and W. Chuang, Ed., "Internationalized Email Addresses in X.509 Certificates", RFC 8398, DOI 10.17487/RFC8398, May 2018, <https://www.rfc-editor.org/info/rfc8398>.

[RFC8550] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Certificate Handling", RFC 8550, DOI 10.17487/RFC8550, April 2019, <https://www.rfc-editor.org/info/rfc8550>.

[RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <https://www.rfc-editor.org/info/rfc8551>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/info/rfc8555>.

[RFC8616] Levine, J., "Email Authentication for Internationalized Mail", RFC 8616, DOI 10.17487/RFC8616, June 2019, <https://www.rfc-editor.org/info/rfc8616>.

7.2. Informative References

[RFC4021] Klyne, G. and J. Palme, "Registration of Mail and MIME Header Fields", RFC 4021, DOI 10.17487/RFC4021, March 2005, <https://www.rfc-editor.org/info/rfc4021>.

[RFC8058] Levine, J. and T. Herkula, "Signaling One-Click Functionality for List Email Headers", RFC 8058, DOI 10.17487/RFC8058, January 2017, <https://www.rfc-editor.org/info/rfc8058>.

Acknowledgements

Thank you to Andreas Schulze, Gerd v. Egidy, James A. Baker, Ben Schwartz, Peter Yee, Hilarie Orman, Michael Jenkins, Barry Leiba, Fraser Tweedale, Daniel Kahn Gillmor, and Benjamin Kaduk for their suggestions, comments, and corrections of this document.

Author's Address

Alexey Melnikov
Isode Ltd
14 Castle Mews
Hampton, Middlesex
TW12 2NP
United Kingdom

Email: alexey.melnikov@isode.com