

ACME Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 26 March 2022

A. Gable  
Internet Security Research Group  
22 September 2021

Automated Certificate Management Environment (ACME) Renewal Information  
(ARI) Extension  
draft-aaron-acme-ari-00

## Abstract

This document specifies how an ACME server may provide hints to ACME clients as to when they should attempt to renew their certificates. This allows servers to mitigate load spikes, and ensures clients do not make false assumptions about appropriate certificate renewal periods.

## Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Automated Certificate Management Environment Working Group mailing list ([acme@ietf.org](mailto:acme@ietf.org)), which is archived at <https://mailarchive.ietf.org/arch/browse/acme/>.

Source for this draft and an issue tracker can be found at <https://github.com/aarongable/draft-acme-ari>.

## Status of This Memo

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

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

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

This Internet-Draft will expire on 26 March 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Conventions and Definitions . . . . .	3
3. Extensions to the ACME Protocol: The "order" Resource . . . .	3
4. Extensions to the ACME Protocol: The "renewalInfo" Resource . . . . .	3
5. Security Considerations . . . . .	4
6. IANA Considerations . . . . .	5
6.1. New Registries . . . . .	5
6.2. ACME Resource Type . . . . .	5
6.3. ACME Order Object Fields . . . . .	5
6.4. ACME Renewal Info Object Fields . . . . .	5
7. Normative References . . . . .	6
Acknowledgments . . . . .	6
Author's Address . . . . .	6

## 1. Introduction

Most ACME clients today choose when to attempt to renew a certificate in one of three ways. They may be configured to renew at a specific interval (e.g. via "cron"); they may parse the issued certificate to determine its expiration date and renew a specific amount of time before then; or they may parse the issued certificate and renew when some percentage of its validity period has passed. The first two techniques create significant barriers against the issuing CA changing certificate lifetimes. All three techniques lead to load clustering for the issuing CA.

Being able to indicate to the client a period in which the issuing CA suggests renewal would allow both dynamic changes to the certificate validity period and proactive smearing of load. This document specifies a mechanism by which ACME servers may provide suggested renewal windows to ACME clients.

## 2. Conventions and Definitions

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. Extensions to the ACME Protocol: The "order" Resource

An ACME server which wishes to provide renewal information MUST include a new field, "renewalInfo", in finalized Order objects.

renewalInfo (optional, string): A URL for renewal information for the certificate that has been issued in response to this order.

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "status": "valid",
  "expires": "2021-01-20T14:09:07.99Z",

  "identifiers": [
    { "type": "dns", "value": "www.example.org" },
    { "type": "dns", "value": "example.org" }
  ],

  "notBefore": "2021-01-01T00:00:00Z",
  "notAfter": "2021-01-08T00:00:00Z",

  "authorizations": [
    "https://example.com/acme/authz/PAniVnsZcis",
    "https://example.com/acme/authz/r4HqLzrSrpI"
  ],

  "finalize": "https://example.com/acme/order/T0locE8rfgo/finalize",
  "certificate": "https://example.com/acme/cert/mAt3xBGaobw",
  "renewalInfo": "https://example.com/acme/renewal/eXoM9UwLgbL"
}
```

Conforming clients SHOULD store the "renewalInfo" URL locally so that they can poll it at any time during the lifetime of the certificate.

## 4. Extensions to the ACME Protocol: The "renewalInfo" Resource

We define a new resource type, the "renewalInfo" resource, as part of the ACME protocol.

The structure of an ACME `renewalInfo` resource is as follows:

`suggestedWindow` (object, required): A JSON object with two keys, `"start"` and `"end"`, whose values are timestamps, encoded in the format specified in [RFC3339], which bound the window of time in which the CA recommends renewing the certificate.

HTTP/1.1 200 OK  
Content-Type: application/json

```
{
  "suggestedWindow": {
    "start": "2021-01-03T00:00:00Z",
    "end": "2021-01-07T00:00:00Z"
  }
}
```

Conforming servers **MUST** provide the `renewalInfo` resource via POST-as-GET; they **SHOULD** provide it via unauthenticated GET as well. Conforming clients **SHOULD** use unauthenticated GET to request `renewalInfo` resources.

The server **SHOULD** include a `Retry-After` header indicating the polling interval that the ACME server recommends. Conforming clients **SHOULD** query the `"renewalInfo"` URL again after the `Retry-After` period has passed, as the server may provide a different `suggestedWindow`.

Conforming clients **SHOULD** select a random time within the suggested window to attempt to renew the certificate. If the selected time is in the past, the client **SHOULD** attempt renewal immediately.

## 5. Security Considerations

The extensions to the ACME protocol described in this document build upon the Security Considerations and threat model defined in Section 10.1 of [RFC8555].

This document specifies that `renewalInfo` resources should be exposed via unauthenticated GET requests, a departure from RFC8555's requirement that clients must send POST-as-GET requests to fetch resources from the server. This is because the information contained in `renewalInfo` resources is not considered confidential, and because allowing `renewalInfo` to be easily cached is advantageous to shed load from clients which do not respect the `Retry-After` header.

## 6. IANA Considerations

Draft note: The following changes to IANA registries have not yet been made.

### 6.1. New Registries

Within the "Automated Certificate Management Environment (ACME) Protocol" registry, IANA has created the new "ACME Renewal Info Object Fields" registry (Section 6.4).

### 6.2. ACME Resource Type

Within the "Automated Certificate Management Environment (ACME) Protocol" registry, the following entry has been added to the "ACME Resource Types" registry.

Field Name	Resource Type	Reference
renewalInfo	Renewal Info object	This draft

Table 1

### 6.3. ACME Order Object Fields

Within the "Automated Certificate Management Environment (ACME) Protocol" registry, the following entry has been added to the "ACME Order Object Fields" registry.

Field Name	Field Type	Configurable	Reference
renewalInfo	string	false	This draft

Table 2

### 6.4. ACME Renewal Info Object Fields

The "ACME Renewal Info Object Fields" registry lists field names that are defined for use in ACME renewal info objects.

Template:

- \* Field name: The string to be used as a field name in the JSON object

\* Field type: The type of value to be provided, e.g., string, boolean, array of string

\* Reference: Where this field is defined

Initial contents:

Field Name	Field type	Reference
suggestedWindow	object	This draft

Table 3

## 7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/info/rfc3339>>.
- [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>>.
- [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>>.

## Acknowledgments

TODO acknowledge.

## Author's Address

A. Gable  
Internet Security Research Group

Email: [aaron@letsencrypt.org](mailto:aaron@letsencrypt.org)

ACME Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 6 October 2022

A. Gable  
Internet Security Research Group  
4 April 2022

Automated Certificate Management Environment (ACME) Renewal Information  
(ARI) Extension  
draft-aaron-acme-ari-02

Abstract

This document specifies how an ACME server may provide hints to ACME clients as to when they should attempt to renew their certificates. This allows servers to mitigate load spikes, and ensures clients do not make false assumptions about appropriate certificate renewal periods.

Current Implementations

Draft note: this section will be removed by the editor before final publication.

Let's Encrypt's Staging environment (available at [lestaging], source at [boulder]) implements this draft specification.

Status of This Memo

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

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

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

This Internet-Draft will expire on 6 October 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Conventions and Definitions . . . . .	3
3. Extensions to the ACME Protocol: The "directory" Resource . .	3
4. Extensions to the ACME Protocol: The "renewalInfo" Resource . . . . .	3
4.1. Getting Renewal Information . . . . .	4
4.2. Updating Renewal Information . . . . .	5
5. Security Considerations . . . . .	7
6. IANA Considerations . . . . .	7
6.1. New Registries . . . . .	7
6.2. ACME Resource Type . . . . .	7
6.3. ACME Renewal Info Object Fields . . . . .	7
7. Normative References . . . . .	8
8. Informative References . . . . .	9
Appendix A. Example Certificates . . . . .	9
A.1. Example End-Entity Certificate . . . . .	9
Example CA Certificate . . . . .	9
Acknowledgments . . . . .	10
Author's Address . . . . .	10

## 1. Introduction

Most ACME [RFC8555] clients today choose when to attempt to renew a certificate in one of three ways. They may be configured to renew at a specific interval (e.g. via cron); they may parse the issued certificate to determine its expiration date and renew a specific amount of time before then; or they may parse the issued certificate and renew when some percentage of its validity period has passed. The first two techniques create significant barriers against the issuing CA changing certificate lifetimes. All three techniques lead to load clustering for the issuing CA.

Being able to indicate to the client a period in which the issuing CA suggests renewal would allow both dynamic changes to the certificate validity period and proactive smearing of load. This document specifies a mechanism by which ACME servers may provide suggested renewal windows to ACME clients.



## 2. Conventions and Definitions

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. Extensions to the ACME Protocol: The "directory" Resource

An ACME server which wishes to provide renewal information MUST include a new field, `renewalInfo`, in its directory object.

Field	URL in Value
<code>renewalInfo</code>	Renewal info

Table 1

HTTP/1.1 200 OK  
Content-Type: application/json

```
{
  "newNonce": "https://example.com/acme/new-nonce",
  "newAccount": "https://example.com/acme/new-account",
  "newOrder": "https://example.com/acme/new-order",
  "newAuthz": "https://example.com/acme/new-authz",
  "revokeCert": "https://example.com/acme/revoke-cert",
  "keyChange": "https://example.com/acme/key-change",
  "renewalInfo": "https://example.com/acme/renewal-info",
  "meta": {
    "termsOfService": "https://example.com/acme/terms/2021-10-05",
    "website": "https://www.example.com/",
    "caaIdentities": ["example.com"],
    "externalAccountRequired": false
  }
}
```

## 4. Extensions to the ACME Protocol: The "renewalInfo" Resource

The "renewalInfo" resource is a new resource type introduced to ACME protocol. This new resource both allows clients to query the server for suggestions on when they should renew certificates, and allows clients to inform the server when they have completed renewal (or otherwise replaced the certificate to their satisfaction).

#### 4.1. Getting Renewal Information

To request the suggested renewal information for a certificate, the client sends a GET request to a path under the server's renewalInfo URL.

The full request URL is computed by concatenating the renewalInfo URL from the server's directory with a forward slash and the base64url-encoded [RFC4648] bytes of a DER-encoded CertID ASN.1 sequence [RFC6960]. Trailing '=' characters MUST be stripped.

For example, to request renewal information for the end-entity certificate given in Appendix A.1, issued by the CA certificate given in Appendix A.2, using SHA256, the client would make the following request (the path has been split onto multiple lines for readability):

```
GET https://example.com/acme/renewal-info/  
MFswCwYJYIZIAWUDBAIBBCCeWLRusNLb--vmWOkxm34qDjTMWkc  
3utIhOMoMwKDqbgQg2iiKWYSZrD-6c88HMZ6vhIHZPamChLlzGH  
eZ7pTS8jYCCD6jRWhlRB8c
```

The ACME Server MAY restrict the hash algorithms which it accepts (for example, only allowing SHA256 to limit the number of potential cache keys); if it receives a request whose embedded signatureAlgorithm field contains an unacceptable OID, it SHOULD respond with HTTP status code 400 (Bad Request).

The structure of an ACME renewalInfo resource is as follows:

suggestedWindow (object, required): A JSON object with two keys, "start" and "end", whose values are timestamps, encoded in the format specified in [RFC3339], which bound the window of time in which the CA recommends renewing the certificate.

explanationURL (string, optional): A URL pointing to a page which may explain why the suggested renewal window is what it is. For example, it may be a page explaining the CA's dynamic load-balancing strategy, or a page documenting which certificates are affected by a mass revocation event. Conforming clients SHOULD provide this URL to their operator, if present.

```
HTTP/1.1 200 OK
Content-Type: application/json
Retry-After: 21600
```

```
{
  "suggestedWindow": {
    "start": "2021-01-03T00:00:00Z",
    "end": "2021-01-07T00:00:00Z"
  },
  "explanationURL": "https://example.com/docs/example-mass-reissuance-event"
}
```

The server SHOULD include a Retry-After header indicating the polling interval that the ACME server recommends. Conforming clients SHOULD query the renewalInfo URL again after the Retry-After period has passed, as the server may provide a different suggestedWindow.

Conforming clients MUST select a uniform random time within the suggested window to attempt to renew the certificate. If the selected time is in the past, the client SHOULD attempt renewal immediately. If the selected time is in the future, but before the next time that the client would wake up normally, the client MAY attempt renewal immediately. In all cases, renewal attempts are subject to the client's existing error backoff and retry intervals.

In particular, cron-based clients may find they need to increase their run frequency to check ARI more frequently. Those clients will need to store information about failures so that increasing their run frequency doesn't lead to retrying failures without proper backoff. Typical information stored should include: number of failures for a given order (defined by the set of names on the order), and time of the most recent failure.

If the client receives no response or a malformed response (e.g. an end timestamp which precedes the start timestamp), it SHOULD make its own determination of when to renew the certificate, and MAY retry the renewalInfo request with appropriate exponential backoff behavior.

#### 4.2. Updating Renewal Information

To update the renewal status of a certificate, the client sends a POST request to the server's renewalInfo URL.

The body of the POST is a JWS object which is authenticated to an account as defined in [RFC8555], Section 6.2, and whose JSON payload has the following structure:

**certID** (required, string): The CertID of the certificate whose renewal information should be updated, in the base64url-encoded version of the DER format with trailing "=" stripped. Note: this is identical to the final path component constructed for GET requests above.

**replaced** (required, boolean): Whether or not the client considers the certificate to have been replaced. A certificate is considered replaced when its revocation would not disrupt any ongoing services, for instance because it has been renewed and the new certificate is in use, or because it is no longer in use. Clients SHOULD NOT send a request where this value is false.

```
POST /acme/renewal-info HTTP/1.1
Host: example.com
Content-Type: application/jose+json
```

```
{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/evOfKhNU60wg",
    "nonce": "JHb54aT_KTXBWQOzGYkt9A",
    "url": "https://example.com/acme/renewal-info"
  }),
  "payload": base64url({
    "certID": "MFswCwYJ...RWhlRB8c",
    "replaced": true
  }),
  "signature": "Q1bURgJoEslbD1c5...3pYdSMLio57mQNN4"
}
```

The server MUST verify that the request is signed by the account key of the Subscriber to which the certificate was originally issued. If the server accepts the request and the update succeeds, it responds with HTTP status code 200 (OK). If the update is rejected or fails, for example because the certificate has already been marked as replaced, the server returns an error.

The server might use this renewal update to inform a number of processes, such as: not sending renewal reminder notifications for certificates that have been marked as replaced; sending empty or error responses to subsequent requests for the certificate's renewal information; or confidently revoking certificates subject to a mass revocation without fear of disrupting the Subscriber's operations.

## 5. Security Considerations

The extensions to the ACME protocol described in this document build upon the Security Considerations and threat model defined in [RFC8555], Section 10.1.

This document specifies that renewalInfo resources MUST be exposed and accessed via unauthenticated GET requests, a departure from RFC8555's requirement that clients must send POST-as-GET requests to fetch resources from the server. This is because the information contained in renewalInfo resources is not considered confidential, and because allowing renewalInfo to be easily cached is advantageous to shed load from clients which do not respect the Retry-After header.

## 6. IANA Considerations

Draft note: The following changes to IANA registries have not yet been made.

### 6.1. New Registries

Within the "Automated Certificate Management Environment (ACME) Protocol" registry, IANA has created the new "ACME Renewal Info Object Fields" registry (Section 6.4).

### 6.2. ACME Resource Type

Within the "Automated Certificate Management Environment (ACME) Protocol" registry, the following entry has been added to the "ACME Resource Types" registry.

Field Name	Resource Type	Reference
renewalInfo	Renewal Info object	This document

Table 2

### 6.3. ACME Renewal Info Object Fields

The "ACME Renewal Info Object Fields" registry lists field names that are defined for use in ACME renewal info objects.

Template:

- \* Field name: The string to be used as a field name in the JSON object
- \* Field type: The type of value to be provided, e.g., string, boolean, array of string
- \* Reference: Where this field is defined

Initial contents:

Field Name	Field type	Reference
suggestedWindow	object	This document

Table 3

## 7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/info/rfc3339>>.
- [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>>.
- [RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <<https://www.rfc-editor.org/info/rfc6960>>.
- [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>>.
- [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>>.

## 8. Informative References

- [boulder] Internet Security Research Group, "Boulder", 2022,  
<<https://github.com/letsencrypt/boulder>>.
- [lestaging] Internet Security Research Group, "Let's Encrypt Staging Environment", 2022,  
<<https://acme-staging-v02.api.letsencrypt.org/directory>>.

## Appendix A. Example Certificates

## A.1. Example End-Entity Certificate

```
-----BEGIN CERTIFICATE-----
MIIDMDCCAhiGAWIBAgIIPqNFaGVEHxwwDQYJKoZIhvcNAQELBQAwIDEEeMBwGA1UE
AxMVbWluaWNhIHJvb3QgY2EgM2ExMzU2MB4XDTIyMDMxNzE3NTEwOVVoXDTI0MDQx
NjE3NTEwOVowFjEUMBIGA1UEAxMLZXhhbXBsZS5jb20wggeiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQCgm9K/c+il2Pf0f8qhgx9SKqXq88cOm9ov9AVRbPA
OWAAewqX2yUAWI4LZBGEgzGzTATkiXfoJ3cN3k39cH6tBbb3iSPuEn7OZpIk9D+e
3Q9/hX+N/jlWkaTB/FNA+7aE5IVWhmdczYilXa10V9r+RcvACJt0gsipBVVJ4jfJ
HnWJJGRZzzxqG/xkQmpXxZO7nOPFc8SxYKWdfcgp+rjR2ogYhSz7BfKoVakGPbpX
vZOuT9z4kkHra/WjwlkQhtHoTXdAxH3qC2UjMzO57Tx+otj0CxAv9O7CTJXISyWB
vEVcmTSzkHS3eZtvvIwPx7I30ITRkYk/tLl1MbyB3SiZAgMBAAGjeDB2MA4GA1Ud
DwEB/wQEAwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAWIwDAYDVR0T
AQH/BAIwADAFBgNVHSMEDAwgBQ4zzDRUaXHVkq1STWkULGU4zGZpTAWBgNVHREE
DzANggtleGFtcGxlLmNvbTANBgkqhkiG9w0BAQsFAAOCAQEAX0aYvmCk7JYGNEXe
+hrOfKawkHYzWvA92cI/Oi6h+oSdHZ2UKzwFNf37cVKZ37FCrrv5pFP/xhhHvrNV
EnOx4IaF7OrnaTu5miZiUWuvRQP7ZGmGNFYbLTEF6/dj+WqyYdVaWzxRqHFulptC
TXysJCeyiGnr+KOOjOOQ9ZlO5JUK3OE4hagPLfaIpDDy6RXQt3ss0iNLuB1+IOtp
1URpvfflLZQ8xPsEgOZyPWocabTwJrtqBwily+lwPFn2mChUx846LwQfxtsXU/lJg
HX2RteNJx7YYNeX3Uf960mgo5an6vE8QNAsIoNHyrGyEmXDhTRe9mCHyiW2S7fZq
o9q12g==
-----END CERTIFICATE-----
```

Example CA Certificate

-----BEGIN CERTIFICATE-----

MIIDSzCCAjOgAwIBAgIIOhNWtJ7Igr0wDQYJKoZIhvcNAQELBQAwIDEeMBwGA1UE  
AxMVbWluaWNhIHJvb3QgY2EgM2ExMzU2MCAXDTIyMDMxNzE3NTEwOV0YDzIxMjIw  
MzE3MTc1MTA5WjAgMR4wHAYDVQQDEXVtaW5pY2Egcm9vdCBjYSAzYTEzNTYwggEi  
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCd3P6cxcCZ7FQOQrYuigReSa8T  
IOPNkmlmX9OrTkPwjThiMNEETYK0lea99yXPK36LUHC6OLmZ9jVQW2NylqwQCOy6  
TrquhnwKgtkBMdAZBLySSEXydkL3r0jA4sf1W130/OLwhstU/yv0J8+pj7eSVOR3  
zJBnYd1AqnXHRswQm299KXgqema7uwsa8cgjrXsBzAhrwrvY1VhpWFSv3lQRDFQg  
c5Z/ZDV9i26qiaJsCCmdisJZWN7N2luUgxdRqzZ4Cr2Xoilg3T+hkb2y/d6ttsPA  
kaSA+pq3q6Qa7/qfGdT5WuUkcHpvKNRWqnwT9rCYlmG00r3hGgc42D/z1VvfAgMB  
AAGjgYyWgYmWdgYDVR0PAAQH/BAQDAgKEMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggr  
BgEFBQcDAjASBgNVHRMBAf8ECDAGAQH/AgEAMB0GA1UdDgQWBBQ4zzDRUaXHVKq1  
STWkULGU4zGZpTafBgNVHSMEGDAWgBQ4zzDRUaXHVKq1STWkULGU4zGZpTANBgkq  
hkiG9w0BAQsFAAOCAQEArbDHhEjGedjb/YjU80aFTPWOMRjgyfQaPPgyxwX6Dsid  
1i2H1x4ud4ntz3sTZzxdQIrOqt1IWTWVCjpStwGxAc+38SdreiTTwy/nikXGa/6W  
ZyQRppR3agh/pl5LHVO6GsJz3YHa7wQhEhj3xsRwa9VrRXgHbLGBPOFVRTHPjaPg  
Gtsv2PN3f67DsPHF47ASqyOIRpLZPQmZIw6D3isJwfl+8CzvlB1ve00Q3uh08IJc  
fspYQXvFBzYa64uKxNAJMi4Pby8cf4r36Wnb7cL4ho3fOHgAltxdW8jgibRzqZpQ  
QKyx2jX7kxeUDt0hFDJE8lOrhP73m66eBNzxe//FQ==

-----END CERTIFICATE-----

#### Acknowledgments

TODO acknowledge.

#### Author's Address

A. Gable  
Internet Security Research Group  
Email: aaron@letsencrypt.org



Automated Certificate Management Environment  
Internet-Draft  
Intended status: Experimental  
Expires: 16 April 2022

B. Sipos  
RKF Engineering  
13 October 2021

Automated Certificate Management Environment (ACME) Delay-Tolerant  
Networking (DTN) Node ID Validation Extension  
draft-ietf-acme-dtnnodeid-06

## Abstract

This document specifies an extension to the Automated Certificate Management Environment (ACME) protocol which allows an ACME server to validate the Delay-Tolerant Networking (DTN) Node ID for an ACME client. The DTN Node ID is encoded as a certificate Subject Alternative Name (SAN) of type otherName with a name form of BundleEID and as an ACME Identifier type "bundleEID".

## Status of This Memo

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

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

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

This Internet-Draft will expire on 16 April 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
1.1. Scope . . . . .	3
1.2. Authorization Strategy . . . . .	5
1.3. Use of CDDL . . . . .	6
1.4. Terminology . . . . .	7
2. Bundle Endpoint ID ACME Identifier . . . . .	7
3. DTN Node ID Validation . . . . .	8
3.1. DTN Node ID Challenge Request Object . . . . .	11
3.2. DTN Node ID Challenge Response Object . . . . .	12
3.3. ACME Node ID Validation Challenge Bundles . . . . .	13
3.3.1. Challenge Bundle Checks . . . . .	14
3.4. ACME Node ID Validation Response Bundles . . . . .	14
3.4.1. Response Bundle Checks . . . . .	16
3.5. Multi-Perspective Validation . . . . .	16
4. Bundle Integrity Gateway . . . . .	17
5. Certificate Request Profile . . . . .	17
5.1. Multiple Identity Claims . . . . .	18
5.2. Generating Encryption-only or Signing-only Bundle Security Certificates . . . . .	18
6. Implementation Status . . . . .	18
7. Security Considerations . . . . .	19
7.1. Threat: Passive Leak of Validation Data . . . . .	19
7.2. Threat: BP Node Impersonation . . . . .	20
7.3. Threat: Bundle Replay . . . . .	20
7.4. Threat: Denial of Service . . . . .	20
7.5. Inherited Security Considerations . . . . .	21
7.6. Out-of-Scope BP Agent Communication . . . . .	21
8. IANA Considerations . . . . .	22
8.1. ACME Identifier Types . . . . .	22
8.2. ACME Validation Methods . . . . .	22
8.3. Bundle Administrative Record Types . . . . .	22
9. Acknowledgments . . . . .	23
10. References . . . . .	23
10.1. Normative References . . . . .	23
10.2. Informative References . . . . .	25
Appendix A. Administrative Record Types CDDL . . . . .	27
Appendix B. Example Authorization . . . . .	27

B.1. Challenge Bundle . . . . .	27
B.2. Response Bundle . . . . .	28
Author's Address . . . . .	29

## 1. Introduction

Although the original purpose of the Automatic Certificate Management Environment (ACME) [RFC8555] was to allow Public Key Infrastructure Using X.509 (PKIX) certificate authorities to validate network domain names of clients, the same mechanism can be used to validate any of the subject claims supported by the PKIX profile [RFC5280].

In the case of this specification, the claim being validated is a Subject Alternative Name (SAN) of type `otherName` with a name form of `BundleEID`, which used to represent an Endpoint ID (EID) for a Delay-Tolerant Networking (DTN) bundle. Currently the URI schemes `"dtn"` and `"ipn"` as defined in [I-D.ietf-dtn-bpbis] are valid for an Endpoint ID. A DTN Node ID is an Endpoint ID with scheme-specific restrictions to identify it as such; currently the `"dtn"` scheme uses an empty demux part and the `"ipn"` scheme uses service number zero.

Because the `BundleEID` claim is new to ACME, a new ACME Identifier type `"bundleEID"` is needed to manage this claim within ACME messaging. A `"bundleEID"` claim can be part of a pre-authorization or as one of the authorizations of an order containing any number of claims.

Once an ACME server validates a Node ID, either as a pre-authorization of the `"bundleEID"` or as one of the authorizations of an order containing a `"bundleEID"`, the client can finalize the order using an associated certificate signing request (CSR). Because a single order can contain multiple identifiers of multiple types, there can be operational issues for a client attempting to, and possibly failing to, validate those multiple identifiers as described in Section 5.1. Once a certificate is issued for a Node ID, how the ACME client configures the Bundle Protocol (BP) agent with the new certificate is an implementation matter.

The scope and behavior of this validation mechanism is similar to that of secured email validation of [RFC8823].

### 1.1. Scope

This document describes the ACME messages, BPv7 payloads, and BPSec requirements needed to validate Node ID ownership. This document does not address:

- \* Mechanisms for communication between ACME client or ACME server and their associated BP agent(s). This document only describes exchanges between ACME client--server pairs and between their BP agents.
- \* Specific BP extension blocks or BPsec security contexts necessary to fulfill the security requirements of this protocol. The exact security context needed, and their parameters, are network-specific.
- \* Policies or mechanisms for defining or configuring bundle integrity gateways, or trusting integrity gateways on an individual entity or across a network.
- \* Mechanisms for locating or identifying other bundle entities (peers) within a network or across an internet. The mapping of Node ID to potential convergence layer (CL) protocol and network address is left to implementation and configuration of the BP Agent and its various potential routing strategies.
- \* Logic for routing bundles along a path toward a bundle's endpoint. This protocol is involved only in creating bundles at a source and handling them at a destination.
- \* Logic for performing rate control and congestion control of bundle transfers. The ACME server is responsible for rate control of validation requests.
- \* Policies or mechanisms for provisioning, deploying, or accessing certificates and private keys; deploying or accessing certificate revocation lists (CRLs); or configuring security parameters on an individual entity or across a network.
- \* Policies or mechanisms for an ACME server to handle mixed-use certificate requests. This specification is focused only on single-use DTN-specific PKIX profiles.

## 1.2. Authorization Strategy

The basic unit of data exchange in a DTN is a Bundle [I-D.ietf-dtn-bpbis], which consists of a data payload with accompanying metadata. An Endpoint ID is used as the destination of a Bundle and can indicate both a unicast or a multicast destination. A Node ID is used to identify the source of a Bundle and is used for routing through intermediate nodes, including the final node(s) used to deliver a Bundle to its destination endpoint. A Node ID can also be used as an endpoint for administrative bundles. More detailed descriptions of the rationale and capabilities of these networks can be found in "Delay-Tolerant Network Architecture" [RFC4838].

When an ACME client requests a pre-authorization or an order with a "bundleEID" identifier type having a value consistent with a Node ID (see Section 4.2.5 of [I-D.ietf-dtn-bpbis]), the ACME server offers a "dtn-nodeid-01" challenge type to validate that Node ID. If the ACME client attempts the authorization challenge to validate a Node ID, the ACME server sends an ACME Node ID Validation Challenge Bundle with a destination of the Node ID being validated. The BP agent on that node receives the Challenge Bundle, generates an ACME key authorization digest, and sends an ACME Node ID Validation Response Bundle in reply. An Integrity Gateway on the client side of the DTN can be used to attest to the source of the Response Bundle. Finally, the ACME server receives the Response Bundle and checks that the digest was generated for the associated ACME challenge and from the client account key associated with the original request. This workflow is shown in the diagram of Figure 1 and defined in Section 3.

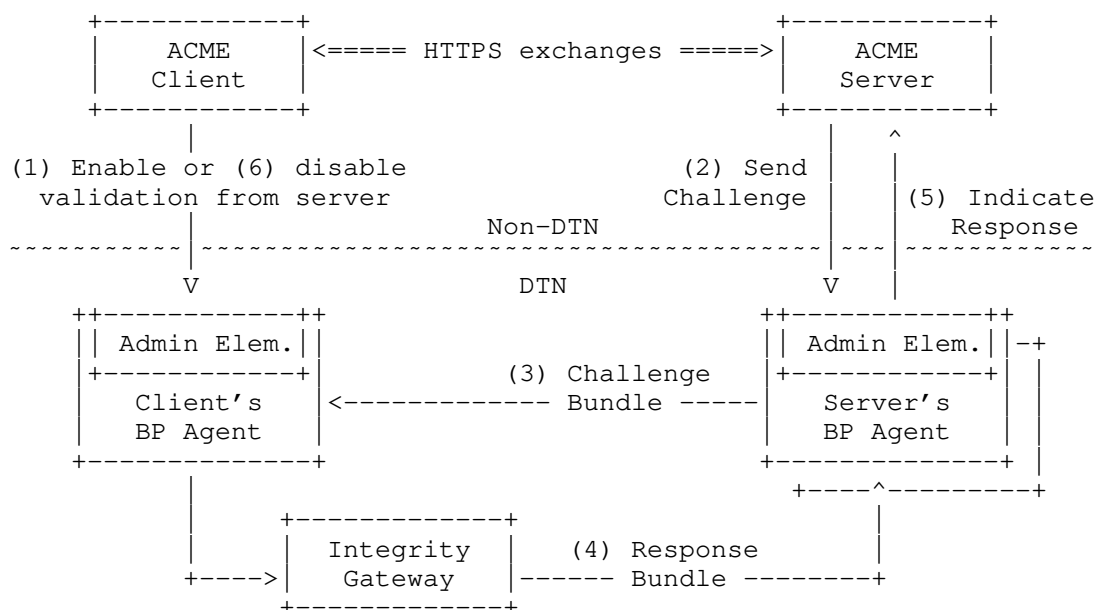


Figure 1: The relationships and flows between Node ID Validation entities

Because the DTN Node ID is used both for routing bundles between BP agents and for multiplexing administrative services within a BP agent, there is no possibility to separate the ACME validation of a Node ID from normal bundle handling for that same Node ID. This leaves administrative record types as a way to leave the Node ID unchanged while disambiguating from other service data bundles.

There is nothing in this protocol which requires network-topological co-location of either the ACME client or ACME server with their associated BP agent. While ACME requires a low-enough latency network to perform HTTPS exchanges between ACME client and server, the client's BP agent (the one being validated) could be on the far side of a long-delay or multi-hop DTN network. The means by which the ACME client or server communicates with its associated BP agent is an implementation matter.

### 1.3. Use of CDDL

This document defines CBOR structure using the Concise Data Definition Language (CDDL) of [RFC8610]. The entire CDDL structure can be extracted from the XML version of this document using the XPath expression:

```
'//sourcecode[@type="cddl"]'
```

The following initial fragment defines the top-level symbols of this document's CDDL, which includes the example CBOR content.

```
start = acme-record / bundle / tstr
```

#### 1.4. Terminology

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.

In this document, several terms are shortened for the sake of terseness. These terms are:

**Challenge Request:** This is a shortened form of the full "DTN Node ID Challenge Request Object". It is a JSON object created by the ACME server for challenge type "dtn-nodeid-01".

**Challenge Response:** This is a shortened form of the full "DTN Node ID Challenge Response Object". It is a JSON object created by the ACME client to authorize a challenge type "dtn-nodeid-01".

**Challenge Bundle:** This is a shortened form of the full "ACME Node ID Validation Challenge Bundle". It is a Bundle created by the BP agent managed by the ACME server to challenge a Node ID claim.

**Response Bundle:** This is a shortened form of the full "ACME Node ID Validation Response Bundle". It is a Bundle created by the BP agent managed by the ACME client to validate a Node ID claim.

#### 2. Bundle Endpoint ID ACME Identifier

This specification is the first to make use of an Bundle Endpoint ID to identify a claim for a certificate request in ACME. In this document, the only purpose for which an Bundle Endpoint ID ACME identifier is validated is as a DTN Node ID (see Section 3), but other specifications can define challenge types for other Endpoint ID uses.

Identifiers of type "bundleEID" in certificate requests MUST appear in an extensionRequest attribute [RFC2985] containing a subjectAltName extension of type otherName with a name form of BundleEID, identified by id-on-bundleEID of [IANA-SMI], consistent with the requirements of Section 4.4.2.1 of [I-D.ietf-dtn-tcpclv4].

Because the BundleEID is encoded as an IA5String it SHALL be treated as being in the percent-encoded form of Section 2.1 of [RFC3986]. Any "bundleEID" identifier which fails to properly percent-decode SHALL be rejected with an ACME error type of "malformed".

The ACME server SHALL decode and normalize (based on scheme-specific syntax) all received identifiers of type "bundleEID". Any "bundleEID" identifier request which uses a scheme not handled by the ACME server or for which the EID does not match the scheme-specific syntax SHALL be rejected with an ACME error type of "rejectedIdentifier".

When an ACME server needs to request proof that a client controls a BundleEID, it SHALL create an authorization with the identifier type "bundleEID". The ACME server SHALL NOT attempt to dereference the EID value on its own. It is the responsibility of a validation method to ensure the EID ownership via scheme-specific means authorized by the ACME client.

An identifier for the Node ID of "dtn://example/" would be formatted as:

```
{
  "type": "bundleEID",
  "value": "dtn://example/"
}
```

### 3. DTN Node ID Validation

The DTN Node ID validation method proves control over a Node ID by requiring the ACME client to configure a BP agent to respond to specific Challenge Bundles sent from the ACME server. The ACME server validates control of the Node ID by verifying that received Response Bundles correspond with the BP Node and client account key being validated.

Similar to the ACME use case for validating email address ownership [RFC8823], this challenge splits the token into several parts, each being transported by a different channel, and the Key Authorization result requires combining all parts of the token. The token parts are:

token-chal This token is unique to, and identifies, each ACME authorization. It is contained in the Challenge Object of Section 3.1 as well as the Challenge Bundle of Section 3.3 and Response Bundle of Section 3.4. Each authorization can consist of multiple Challenge Bundles (e.g. taking different routes), but they all share the same token-chal value. This ensures that the



Key Authorization is bound to the specific ACME challenge (and parent ACME authorization) and also allows the ACME client's BP agent to filter-in only valid Challenge Bundles. This token is also accessible to DTN on-path eavesdroppers.

**token-bundle** This token is unique to each Challenge Bundle sent by the ACME server. It is contained in the Challenge Bundle of Section 3.3 and Response Bundle of Section 3.4. This ensures that the Key Authorization is bound to the ability to receive the Challenge Bundle and not just have access to the ACME Challenge Object. This token is also accessible to DTN on-path eavesdroppers.

For each ACME server, the pair of token-chal and token-bundle values is the unique correlator between Challenge and Response bundles. Because multiple Challenge Bundles can be sent to validate the same Node ID, the token-bundle in the response is needed to correlate with the expected Key Authorization digest.

The DTN Node ID Challenge SHALL only be allowed for an EID usable as a DTN Node ID, which [I-D.ietf-dtn-bpbis]. When an ACME server supports Node ID validation, the ACME server SHALL define a challenge object in accordance with Section 3.1. Once this challenge object is defined, the ACME client may begin the validation.

To initiate a Node ID validation, the ACME client performs the following steps:

1. The ACME client POSTs a newOrder or newAuthz request including the identifier of type "bundleEID" for the desired Node ID. From either of these entry points an authorization for the "bundleEID" type is indicated by the ACME server. See Section 7.4 of [RFC8555] for more details.
2. The ACME client obtains the challenge source Node ID and token-chal from the challenge object in accordance with Section 3.1.
3. The ACME client indicates to the administrative element of its BP agent the source Node ID and challenge token-chal which is authorized for use and the associated client account key thumbprint. The ACME client SHALL wait, if necessary, until the agent is configured before proceeding to the next step.
4. The ACME client POSTs a challenge response to the challenge URL on the ACME server accordance with Section 7.5.1 of [RFC8555]. The payload object is constructed in accordance with Section 3.2.

5. The administrative element waits for a Challenge Bundle to be received with the authorized ACME parameters, including its token-bundle payload, in accordance with Section 3.3.
6. The administrative element concatenates token-bundle with token-chal (each as base64url-encoded text strings) and computes the Key Authorization in accordance with Section 8.1 of [RFC8555] using the full token and client account key thumbprint.
7. The administrative element computes the SHA-256 digest of the Key Authorization result and generates a Response Bundle to send back to the ACME server in accordance with Section 3.4.
8. The ACME client waits for the authorization to be finalized on the ACME server in accordance with Section 7.5.1 of [RFC8555].
9. Once the challenge is completed (successfully or not), the ACME client indicates to the BP agent that the validation source and token-chal is no longer usable. If the authorization fails, the ACME client MAY retry the challenge from Step 3.

The ACME server verifies the client's control over a Node ID by performing the following steps:

1. The ACME server receives a newOrder or newAuthz request including the identifier of type "bundleEID", where the URI value is a Node ID.
2. The ACME server generates an authorization for the Node ID with challenge type "dtn-nodeid-01" in accordance with Section 3.1.
3. The ACME server receives a POST to the challenge URL indicated from the authorization object. The payload is handled in accordance with Section 3.2.
4. The ACME server sends, via the administrative element of its BP agent, one or more Challenge Bundles in accordance with Section 3.3. Each challenge bundle SHALL contain a distinct token-bundle to be able to correlate with a response bundle. Computing an expected Key Authorization digest is not necessary until a response is received.
5. The ACME server waits for Response Bundle(s) for a limited interval of time (based on the challenge response object of Section 3.2). Responses are encoded in accordance with Section 3.4.

6. Once received and decoded, the ACME server checks the contents of each Response Bundle in accordance with Section 3.4.1. After all Challenge Bundles have either been responded to or timed-out, the ACME server policy (see Section 3.5) determines whether the validation is successful. If validation is not successful, a client may retry the challenge which starts in Step 3.

When responding to a Challenge Bundle, a BP agent SHALL send a single Response Bundle in accordance with Section 3.4. A BP agent SHALL respond to ACME challenges only within the interval of time, only for the Node ID, and only for the token-chal indicated by the ACME client. A BP agent SHALL respond to multiple Challenge Bundles with the same ACME parameters but different bundle identities (source Node ID and timestamp); these correspond with the ACME server validating via multiple routing paths.

### 3.1. DTN Node ID Challenge Request Object

The DTN Node ID Challenge request object is defined by the ACME server when it supports validating Node IDs.

The DTN Node ID Challenge request object has the following content:

type (required, string): The string "dtn-nodeid-01".

source (required, array of string): An unordered list of possible source Node ID of bundles originating at the BP agent(s) of the ACME server. See Section 3.5 for a discussion of multi-perspective validation using multiple sources. The array SHALL be non-empty. The array MAY contain Node IDs which are not actually used as a challenge bundle source.

token-chal (required, string): A random value that uniquely identifies the challenge. This value MUST have at least 128 bits of entropy. It MUST contain any characters outside the base64url alphabet as described in Section 5 of [RFC4648]. Trailing '=' padding characters MUST be stripped. See [RFC4086] for additional information on randomness requirements.

```
{
  "type": "dtn-nodeid-01",
  "url": "https://example.com/acme/chall/prV_B7yEyA4",
  "source": ["dtn://acme-server/"],
  "token-chal": "tPUZNY4ONik6LxErRFEjVw"
}
```

The token-chal value included in this object is fixed for the entire challenge, and may correspond with multiple separate token-bundle values when multiple Challenge Bundles are sent for a single validation.

### 3.2. DTN Node ID Challenge Response Object

The DTN Node ID Challenge response object is defined by the ACME client when it authorizes validation of a Node ID. Because a DTN has the potential for significantly longer delays than a non-DTN network, the ACME client is able to inform the ACME server if a particular validation round-trip is expected to take longer than normal network delays (on the order of seconds).

The DTN Node ID Challenge response object has the following content:

rtt (optional, number): An expected round-trip time (RTT), in seconds, between sending a Challenge Bundle and receiving a Response Bundle. This value is a hint to the ACME server for how long to wait for responses but is not authoritative. The minimum RTT value SHALL be zero. There is no special significance to zero-value RTT, it simply indicates the response is expected in less than the least significant unit used by the ACME client.

```
{  
  "rtt": 300.0  
}
```

A challenge response is not sent until the BP agent has been configured to properly respond to the challenge, so the RTT value is meant to indicate any node-specific path delays expected to encountered from the ACME server. Because there is no requirement on the path (or paths) which bundles may traverse between the ACME server and the BP agent, and the ACME server can attempt some path diversity, the RTT value SHOULD be pessimistic.

A default bundle response interval, used when the object does not contain an RTT, SHOULD be a configurable parameter of the ACME server. If the ACME client indicated an RTT value in the object, the response interval SHOULD be twice the RTT (with limiting logic applied as described below). The lower limit on response interval is network-specific, but SHOULD NOT be shorter than one second. The upper limit on response interval is network-specific, but SHOULD NOT be longer than one minute (60 seconds) for a terrestrial-only DTN.

### 3.3. ACME Node ID Validation Challenge Bundles

Each ACME Node ID Validation Challenge Bundle SHALL be structured and encoded in accordance with [I-D.ietf-dtn-bpbis].

Each Challenge Bundle has parameters as listed here:

**Bundle Processing Control Flags:** The primary block flags SHALL indicate that the payload is an administrative record. The primary block flags SHALL indicate that user application acknowledgement is requested; this flag distinguishes the Challenge Bundle from the Response Bundle. The primary block flags MAY indicate that status reports are requested; such status can be helpful to troubleshoot routing issues.

**Destination EID:** The Destination EID SHALL be the ACME-server-normalized Node ID being validated.

**Source Node ID:** The Source Node ID SHALL indicate the Node ID of the BP agent of the ACME server performing the challenge. The challenge bundle source SHALL be present in the "source" array of the challenge object (see Section 3.1)

**Creation Timestamp and Lifetime:** The Creation Timestamp SHALL be set to the time at which the challenge was generated. The Lifetime SHALL indicate the response interval (of Section 3.2) for which ACME server will accept responses to this challenge.

**Administrative Record Type Code:** Set to the ACME Node ID Validation type code defined in Section 8.3.

**Administrative Record Content:** The Challenge Bundle administrative record content SHALL consist of a CBOR map containing two pairs:

- \* One pair SHALL consist of key 1 with value of ACME challenge token-chal, copied from the challenge object, represented as a CBOR byte string.
- \* One pair SHALL consist of key 2 with value of ACME challenge token-bundle, represented as a CBOR byte string. The token-bundle is a random value that uniquely identifies the challenge bundle. This value MUST have at least 128 bits of entropy. See [RFC4086] for additional information on randomness requirements.

This structure is part of the extension CDDL in Appendix A. An example full Challenge Bundle is included in Appendix B.1

If the BP agent generating a Challenge Bundle does not have a well-synchronized clock or the agent responding to the challenge is expected to not have a well-synchronized clock, the bundle SHALL include a Bundle Age extension block.

Challenge Bundles SHALL include a Block Integrity Block (BIB) in accordance with Section 4 or with a Security Source identical to the bundle Source Node. Challenge Bundles SHALL NOT be directly encrypted by Block Confidentiality Block (BCB) or any other method (see Section 7.1).

### 3.3.1. Challenge Bundle Checks

A proper Challenge Bundle meets all of the following criteria:

- \* The Challenge Bundle was received within the time interval allowed for the challenge. The allowed interval starts at the Creation Timestamp and extends for the Lifetime of the Challenge Bundle.
- \* The Challenge Bundle Source Node ID is identical to the Node ID indicated in the ACME challenge object. The comparison of Node IDs SHALL use the comparison logic of the NODE-ID from Section 4.4.1 of [I-D.ietf-dtn-tcpclv4].
- \* The Challenge Bundle contains a BIB which covers at least the primary block and payload. That BIB has a security source which is trusted and passes security-context-specific validation (i.e. MAC or signature verification).
- \* The challenge payload contains the token-chal as indicated in the ACME challenge object. The challenge payload contains a token-bundle meeting the definition in Section 3.3.

Any of the failures above SHALL cause the challenge bundle to be deleted and otherwise ignored by the BP agent. The BP agent MAY send status reports about the deletion if allowed by security policy.

### 3.4. ACME Node ID Validation Response Bundles

Each ACME Node ID Validation Response Bundle SHALL be structured and encoded in accordance with [I-D.ietf-dtn-bpbis].

Each Response Bundle has parameters as listed here:

Bundle Processing Control Flags: The primary block flags SHALL indicate that the payload is an administrative record. The primary block flags SHALL NOT indicate that user application acknowledgement is requested; this flag distinguishes the Response

Bundle from the Challenge Bundle. The primary block flags MAY indicate that status reports are requested; such status can be helpful to troubleshoot routing issues.

**Destination EID:** The Destination EID SHALL be identical to the Source Node ID of the Challenge Bundle to which this response corresponds.

**Source Node ID:** The Source Node ID SHALL be identical to the Destination EID of the Challenge Bundle to which this response corresponds.

**Creation Timestamp and Lifetime:** The Creation Timestamp SHALL be set to the time at which the response was generated. The response Lifetime SHALL indicate the response interval remaining if the Challenge Bundle indicated a limited Lifetime.

**Administrative Record Type Code:** Set to the ACME Node ID Validation type code defined in Section 8.3.

**Administrative Record Content:** The Response Bundle administrative record content SHALL consist of a CBOR map containing three pairs:

- \* One pair SHALL consist of key 1 with value of ACME challenge token-chal, copied from the Request Bundle, represented as a CBOR byte string.
- \* One pair SHALL consist of key 2 with value of ACME challenge token-bundle, copied from the Request Bundle, represented as a CBOR byte string.
- \* One pair SHALL consist of key 3 with value of the SHA-256 digest [FIPS180-4] of the ACME Key Authorization in accordance with Section 8.1 of [RFC8555], represented as a CBOR byte string.

This structure is part of the extension CDDL in Appendix A. An example full Response Bundle is included in Appendix B.2

If the BP agent responding to a Challenge Bundle does not have a well-synchronized clock, it SHALL use any information about last-hop delays and (if present) Bundle Age extension data to infer the age of the Challenge Bundle and lifetime of the Response Bundle.

Response Bundles SHALL include a BIB in accordance with Section 4. Response Bundles SHALL NOT be directly encrypted by BCB or any other method (see Section 7.1 for explanation).

### 3.4.1. Response Bundle Checks

A proper Response Bundle meets all of the following criteria:

- \* The Response Bundle was received within the time interval allowed for the challenge. The allowed interval starts at the Creation Timestamp and extends for the Lifetime of the associated Challenge Bundle. The interval of the Challenge Bundle is used here because the interval of the Response Bundle could be made to disagree with the Challenge Bundle.
- \* The Response Bundle Source Node ID is identical to the Node ID being validated. The comparison of Node IDs SHALL use the comparison logic of the NODE-ID from Section 4.4.1 of [I-D.ietf-dtn-tcpclv4].
- \* The Response Bundle contains a BIB which covers at least the primary block and payload. That BIB has a security source which is trusted and passes security-context-specific validation.
- \* The response payload contains the same token-chal and token-bundle as sent in the Challenge Bundle (this is also how the two bundles are correlated). The response payload contains the expected Key Authorization digest computed by the ACME server.

Any of the failures above SHALL cause that single-perspective validation to fail. Any of the failures above SHOULD be distinguished as subproblems to the ACME client. The lack of a response within the expected response interval, as defined in Section 3.2, SHALL also be treated as a validation failure.

### 3.5. Multi-Perspective Validation

To avoid possible on-path attacks in certain networks, an ACME server can perform a single validation using multiple challenge bundle sources or via multiple routing paths. This technique is called multi-perspective validation as recommended in Section 10.2 of [RFC8555] and an implementation used by Let's Encrypt is described in [LE-multi-perspective].

When required by policy, an ACME server SHALL send multiple challenge bundles from different sources in the DTN network. When multiple Challenge Bundles are sent for a single validation, it is a matter of ACME server policy to determine whether or not the validation as a whole is successful. The result of each single-source validation is defined as success or failure in Section 3.4.1.



A RECOMMENDED validation policy is to succeed if the challenge from a primary bundle source is successful and if there are no more than one failure from a secondary source. The determination of which perspectives are considered primary or secondary is an implementation matter.

Regardless of whether a validation is single- or multi-perspective, a validation failure SHALL be indicated by an ACME error type of "incorrectResponse". Each specific perspective failure SHOULD be indicated to the ACME client as a validation subproblem.

#### 4. Bundle Integrity Gateway

This section defines a BIB use which closely resembles the function of DKIM email signing [RFC6376]. In this mechanism a routing node in a DTN sub-network attests to the origination of a bundle by adding a BIB before forwarding it. The bundle receiver then need not trust the source of the bundle, but only trust this security source node. The receiver needs policy configuration to know which security sources are permitted to attest for which bundle sources.

An integrity gateway SHALL validate the Source Node ID of a bundle, using local-network-specific means, before adding a BIB to the bundle. The exact means by which an integrity gateway validates a bundle's source is network-specific, but could use physical-layer, network-layer or BP-convergence-layer authentication. The bundle source could also add its own BIB with a local-network-specific security context or local-network-specific key material (i.e. a group key shared within the local network).

When an integrity gateway adds a BIB it SHALL be in accordance with [I-D.ietf-dtn-bpsec]. The BIB targets SHALL cover both the payload block and the primary block (either directly as a target or as additional authenticated data for the payload block MAC/signature). The Security Source of this BIB SHALL be either the bundle source Node ID itself or a routing node trusted by the destination node (see Section 7.2).

#### 5. Certificate Request Profile

The ultimate purpose of this ACME validation is to allow a CA to issue certificates following the profiles of Section 4.4.2 of [I-D.ietf-dtn-tcpclv4], [I-D.sipos-dtn-udpcl], and [I-D.bsipos-dtn-bpsec-cose]. These purposes are referred to here as bundle security certificates.

One defining aspect of bundle security certificates is the Extended Key Usage key purpose id-kp-bundleSecurity of [IANA-SMI]. When requesting a certificate which includes a Node ID SAN, the CSR SHOULD include an Extended Key Usage of id-kp-bundleSecurity. When a bundle security certificate is issued based on a validated Node ID SAN, the certificate SHALL include an Extended Key Usage of id-kp-bundleSecurity.

#### 5.1. Multiple Identity Claims

A single bundle security CSR MAY contain a mixed set of SAN claims, including combinations of "ip", "dns", and "bundleEID" claims. There is no restriction on how a certificate combines these claims, but each claim MUST be validated by an ACME server to issue such a certificate as part of an associated ACME order. This is no different than the existing behavior of [RFC8555] but is mentioned here to make sure that CA policy handles such situations; especially related to validation failure of an identifier in the presence of multiple identifiers. The specific use case of [I-D.ietf-dtn-tcpclv4] allows, and for some network policies requires, that a certificate authenticate both the DNS name of an entity as well as the Node ID of the entity.

#### 5.2. Generating Encryption-only or Signing-only Bundle Security Certificates

ACME extensions specified in this document can be used to request encryption-only or signing-only bundle security certificates.

In order to request signing only bundle security certificate, 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 bundle security certificate, 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 key usage extension in the CSR, signals to ACME server to issue a bundle security certificate suitable for both signing and encryption.

### 6. Implementation Status

This section is to be removed before publishing as an RFC.

[NOTE to the RFC Editor: please remove this section before publication, as well as the reference to [RFC7942] and [github-dtn-demo-agent] and [github-dtn-wireshark].]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations can exist.

An example implementation of the this draft of ACME Node ID Validation has been created as a GitHub project [github-dtn-demo-agent] and is intended to use as a proof-of-concept and as a possible source of interoperability testing.

A Wireshark dissector for of the this draft of ACME Node ID Validation has been created as a GitHub project [github-dtn-wireshark] and is intended to be used to inspect and troubleshoot implementations.

## 7. Security Considerations

This section separates security considerations into threat categories based on guidance of BCP 72 [RFC3552].

### 7.1. Threat: Passive Leak of Validation Data

Because this challenge mechanism is used to bootstrap security between DTN Nodes, the challenge and its response are likely to be transferred in plaintext. The only ACME data present on-the-wire is a random token and a cryptographic digest, so there is no sensitive data to be leaked within the Node ID Validation bundle exchange. Because each challenge uses a separate token, there is no value in an on-path attacker seeing the tokens from past challenges and/or responses.

It is possible for intermediate BP nodes to encapsulate-and-encrypt Challenge and/or Response Bundles while they traverse untrusted networks, but that is a DTN configuration matter outside of the scope of this document.

## 7.2. Threat: BP Node Impersonation

As described in Section 8.1 of [RFC8555], it is possible for an active attacker to alter data on both ACME client channel and the DTN validation channel.

The primary mitigation is to delegate bundle integrity sourcing to a trusted routing node near, in the sense of bundle routing topology, to the bundle source node as defined in Section 4. This is functionally similar to DKIM signing of [RFC6376] and provides some level of bundle origination.

Another way to mitigate single-path on-path attacks is to attempt validation of the same Node ID from multiple sources or via multiple bundle routing paths, as defined in Section 3.5. It is not a trivial task to guarantee bundle routing though, so more advanced techniques such as onion routing (using bundle-in-bundle encapsulation [I-D.ietf-dtn-bibect]) could be employed.

## 7.3. Threat: Bundle Replay

It is possible for an on-path attacker to replay both Challenge Bundles or Response Bundles. Even in a properly-configured DTN it is possible that intermediate bundle routers to use multicast forwarding of a unicast-destination bundle.

Ultimately, the point of the ACME bundle exchange is to derive a Key Authorization and its cryptographic digest and communicate it back to the ACME server for validation, so the uniqueness of the Key Authorization directly determines the scope of replay validity. The uniqueness of each token-bundle to each challenge bundle ensures that the Key Authorization is unique to the challenge bundle. The uniqueness of each token-chal to the ACME challenge ensures that the Key Authorization is unique to that ACME challenge.

Having each bundle's primary block and payload block covered by a BIB from a trusted security source (see Section 4) ensures that a replayed bundle cannot be altered in the blocks used by ACME. All together, these properties mean that there is no degraded security caused by replay of either a Challenge Bundle or a Response Bundle even in the case where the primary or payload block is not covered by a BIB. The worst that can come of bundle replay is the waste of network resources as described in Section 7.4.

## 7.4. Threat: Denial of Service

The behaviors described in this section all amount to a potential denial-of-service to a BP agent.

A malicious entity can continually send Challenge Bundles to a BP agent. The victim BP agent can ignore Challenge Bundles which do not conform to the specific time interval and challenge token for which the ACME client has informed the BP agent that challenges are expected. The victim BP agent can require all Challenge Bundles to be BIB-signed to ensure authenticity of the challenge.

A malicious entity can continually send Response Bundles to a BP agent. The victim BP agent can ignore Response Bundles which do not conform to the specific time interval or Source Node ID or challenge token for an active Node ID validation.

Similar to other validation methods, an ACME server validating a DTN Node ID could be used as a denial of service amplifier. For this reason any ACME server can rate-limit validation activities for individual clients and individual certificate requests.

#### 7.5. Inherited Security Considerations

Because this protocol relies on ACME for part of its operation, the security considerations of [RFC8555] apply to all ACME client--server exchanges during Node ID validation.

Because this protocol relies on BPv7 for part of its operation, the security considerations of [I-D.ietf-dtn-bpbis] and [I-D.ietf-dtn-bpsec] apply to all BP messaging during Node ID validation.

#### 7.6. Out-of-Scope BP Agent Communication

Although messaging between an ACME client or ACME server and its associated BP agent are out-of-scope for this document, both of those channels are critical to Node ID validation security. Either channel can potentially leak data or provide attack vectors if not properly secured. These channels need to protect against spoofing of messaging in both directions to avoid interruption of normal validation sequencing and to prevent false validations from succeeding.

The ACME server and its BP agent exchange the outgoing token-chal, token-bundle, and Key Authorization digest but these values do not need to be confidential (they are also in plaintext over the BP channel).

Depending on implementation details, the ACME client might transmit the client account key thumbprint to its BP agent to allow computing the Key Authorization digest on the BP agent. If an ACME client does transmit its client account key thumbprint to a BP agent, it is

important that this data is kept confidential because it provides the binding of the client account key to the Node ID validation (as well as for all other types of ACME validation). Avoiding this transmission would require a full round-trip between BP agent and ACME client, which can be undesirable if the two are separated by a long-delay network.

## 8. IANA Considerations

This specification adds to the ACME registry and BP registry for this behavior.

### 8.1. ACME Identifier Types

Within the "Automated Certificate Management Environment (ACME) Protocol" registry [IANA-ACME], the following entry has been added to the "ACME Identifier Types" sub-registry.

Label	Reference
uri	This specification and [RFC3986]

Table 1

### 8.2. ACME Validation Methods

Within the "Automated Certificate Management Environment (ACME) Protocol" registry [IANA-ACME], the following entry has been added to the "ACME Validation Methods" sub-registry.

Label	Identifier Type	ACME	Reference
dtm-nodeid-01	uri	Y	This specification

Table 2

### 8.3. Bundle Administrative Record Types

Within the "Bundle Protocol" registry [IANA-BP], the following entries have been added to the "Bundle Administrative Record Types" sub-registry.

[NOTE to the RFC Editor: For [RFC5050] compatibility the AR-TBD value needs to be no larger than 15, but such compatibility is not needed. For BPbis the AR-TBD value needs to be no larger than 65535 as defined by [I-D.sipos-bpv7-admin-iana].]

Bundle Protocol Version	Value	Description	Reference
7	AR-TBD	ACME Node ID Validation	This specification

Table 3

## 9. Acknowledgments

This specification is based on DTN use cases related to PKIX certificate issuance.

The workflow and terminology of this validation method was originally copied from the work of Alexey Melnikov in [RFC8823].

## 10. References

### 10.1. Normative References

- [FIPS180-4] National Institute of Standards and Technology, "Secure Hash Standard (SHS)", FIPS PUB 180-4, August 2015, <<https://csrc.nist.gov/publications/detail/fips/180/4/final>>.
- [IANA-ACME] IANA, "Automated Certificate Management Environment (ACME) Protocol", <<https://www.iana.org/assignments/acme/>>.
- [IANA-BP] IANA, "Bundle Protocol", <<https://www.iana.org/assignments/bundle/>>.
- [IANA-SMI] IANA, "Structure of Management Information (SMI) Numbers", <<https://www.iana.org/assignments/smi-numbers/>>.
- [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>>.

- [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>>.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <<https://www.rfc-editor.org/info/rfc4086>>.
- [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>>.
- [RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, DOI 10.17487/RFC4838, April 2007, <<https://www.rfc-editor.org/info/rfc4838>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [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>>.
- [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>>.



[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <<https://www.rfc-editor.org/info/rfc8610>>.

[I-D.ietf-dtn-bpbis] Burleigh, S., Fall, K., and E. J. Birrane, "Bundle Protocol Version 7", Work in Progress, Internet-Draft, draft-ietf-dtn-bpbis-31, 25 January 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-dtn-bpbis-31>>.

[I-D.ietf-dtn-bpsec] III, E. J. B. and K. McKeever, "Bundle Protocol Security Specification", Work in Progress, Internet-Draft, draft-ietf-dtn-bpsec-27, 16 February 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-dtn-bpsec-27>>.

## 10.2. Informative References

[RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, DOI 10.17487/RFC5050, November 2007, <<https://www.rfc-editor.org/info/rfc5050>>.

[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>>.

[RFC8823] Melnikov, A., "Extensions to Automatic Certificate Management Environment for End-User S/MIME Certificates", RFC 8823, DOI 10.17487/RFC8823, April 2021, <<https://www.rfc-editor.org/info/rfc8823>>.

[I-D.ietf-dtn-bibect] Burleigh, S., "Bundle-in-Bundle Encapsulation", Work in Progress, Internet-Draft, draft-ietf-dtn-bibect-03, 18 February 2020, <<https://datatracker.ietf.org/doc/html/draft-ietf-dtn-bibect-03>>.

- [I-D.ietf-dtn-tcpclv4]  
Sipos, B., Demmer, M., Ott, J., and S. Perreault, "Delay-Tolerant Networking TCP Convergence Layer Protocol Version 4", Work in Progress, Internet-Draft, draft-ietf-dtn-tcpclv4-28, 6 October 2021,  
<<https://datatracker.ietf.org/doc/html/draft-ietf-dtn-tcpclv4-28>>.
- [I-D.sipos-dtn-udpcl]  
Sipos, B., "Delay-Tolerant Networking UDP Convergence Layer Protocol", Work in Progress, Internet-Draft, draft-sipos-dtn-udpcl-01, 26 March 2021,  
<<https://datatracker.ietf.org/doc/html/draft-sipos-dtn-udpcl-01>>.
- [I-D.sipos-bpv7-admin-iana]  
Sipos, B., "Bundle Protocol Version 7 Administrative Record Types Registry", Work in Progress, Internet-Draft, draft-sipos-bpv7-admin-iana-00, 13 October 2021,  
<<https://datatracker.ietf.org/doc/html/draft-sipos-bpv7-admin-iana-00>>.
- [I-D.bsipos-dtn-bpsec-cose]  
Sipos, B., "DTN Bundle Protocol Security COSE Security Context", Work in Progress, Internet-Draft, draft-bsipos-dtn-bpsec-cose-06, 3 June 2021,  
<<https://datatracker.ietf.org/doc/html/draft-bsipos-dtn-bpsec-cose-06>>.
- [github-dtn-demo-agent]  
Sipos, B., "Python implementation of basic BPv7-related protocols",  
<<https://github.com/BSipos-RKF/dtn-demo-agent/>>.
- [github-dtn-wireshark]  
Sipos, B., "Wireshark Dissectors for BPv7-related Protocols",  
<<https://github.com/BSipos-RKF/dtn-wireshark/>>.
- [LE-multi-perspective]  
Aas, J., McCarney, D., and R. Shoemaker, "Multi-Perspective Validation Improves Domain Validation Security", 19 February 2020,  
<<https://letsencrypt.org/2020/02/19/multi-perspective-validation.html>>.

## Appendix A. Administrative Record Types CDDL

[NOTE to the RFC Editor: The "0xFFFF" in this CDDL is replaced by the "ACME Node ID Validation" administrative record type code.]

The CDDL extension of BP [I-D.ietf-dtn-bpbis] for the ACME bundles is:

```
; All ACME records have the same structure
$admin-record /= [0xFFFF, acme-record]
acme-record = {
    token-chal,
    token-bundle,
    ? key-auth-digest ; present for Response Bundles
}
token-chal = (1 => bstr)
token-bundle = (2 => bstr)
key-auth-digest = (3 => bstr)
```

## Appendix B. Example Authorization

[NOTE to the RFC Editor: The "0xFFFF" in these examples are replaced by the "ACME Node ID Validation" administrative record type code.]

This example is a bundle exchange for the ACME server with Node ID "dtn://acme-server/" performing a verification for ACME client Node ID "dtn://acme-client/". The example bundles use no block CRC or BPsec integrity, which is for simplicity and is not recommended for normal use. The provided figures are extended diagnostic notation [RFC8610].

For this example the ACME client key thumbprint has the base64url encoded value of:

"LPJNul-wow4m6DsrxbninhsWHlwfp0JecwQzYpOLmCQ"

And the ACME-server generated token-chal has the base64url-encoded value of:

"tPUZNY4ONIk6LxErRFEjVw"

## B.1. Challenge Bundle

For the single challenge bundle in this example, the token-bundle (transported as byte string via BP) has the base64url-encoded value of:

"p3yRYFU4KxwQaHQjJ2RdiQ"

The minimal-but-valid Challenge Bundle is shown in Figure 2. This challenge requires that the ACME client respond within a 60 second time window.

```
[
  [
    7, / BP version /
    0x22, / flags: user-app-ack, payload-is-an-admin-record /
    0, / CRC type: none /
    [1, "//acme-client/"], / destination /
    [1, "//acme-server/"], / source /
    [1, "//acme-server/"], / report-to /
    [1000000, 0], / timestamp: 2000-01-01T00:16:40+00:00 /
    60000 / lifetime: 60s /
  ],
  [
    1, / block type code /
    1, / block number /
    0, / flags /
    0, / CRC type: none /
    <<[ / type-specific data /
      0xFFFF, / record-type-code /
      { / record-content /
        1: b64'tPUZNY4ONIk6LxErRFEjVw' / token-chal /
        2: b64'p3yRYFU4KxwQaHQjJ2RdiQ' / token-bundle /
      }
    ]>>
  ]
]
```

Figure 2: Example Challenge Bundle

## B.2. Response Bundle

When the tokens are combined with the key thumbprint, the full Key Authorization value (a single string split across lines for readability) is:

```
"p3yRYFU4KxwQaHQjJ2RdiQtPUZNY4ONIk6LxErRFEjVw."
"LPJNul-wow4m6DsqxnbnnhsWHlwfp0JecwQzYpOLmCQ"
```

The minimal-but-valid Response Bundle is shown in Figure 3. This response indicates that there is 30 seconds remaining in the response time window.

```

[
  [
    7, / BP version /
    0x02, / flags: payload-is-an-admin-record /
    0, / CRC type: none /
    [1, "//acme-server/"], / destination /
    [1, "//acme-client/"], / source /
    [1, 0], / report-to: none /
    [1030000, 0], / timestamp: 2000-01-01T00:17:10+00:00 /
    30000 / lifetime: 30s /
  ],
  [
    1, / block type code /
    1, / block number /
    0, / flags /
    0, / CRC type: none /
    <<[ / block-type-specific data /
      0xFFFF, / record-type-code /
      { / record-content /
        1: b64'tPUZNY4ONIk6LxErRFEjVw' / token-chal /
        2: b64'p3yRYFU4KxwQaHQjJ2RdiQ' / token-bundle /
        3: b64'mVIOJEQZie8XpYM6MMVSQUiNPH64URnhM9niJ5XHrew'
          / key auth. digest /
      }
    ]>>
  ]
]

```

Figure 3: Example Response Bundle

## Author's Address

Brian Sipos  
 RKF Engineering Solutions, LLC  
 7500 Old Georgetown Road  
 Suite 1275  
 Bethesda, MD 20814-6198  
 United States of America  
  
 Email: brian.sipos+ietf@gmail.com

Automated Certificate Management Environment  
Internet-Draft  
Intended status: Experimental  
Expires: 3 September 2022

B. Sipos  
RKF Engineering  
2 March 2022

Automated Certificate Management Environment (ACME) Delay-Tolerant  
Networking (DTN) Node ID Validation Extension  
draft-ietf-acme-dtnnodeid-09

## Abstract

This document specifies an extension to the Automated Certificate Management Environment (ACME) protocol which allows an ACME server to validate the Delay-Tolerant Networking (DTN) Node ID for an ACME client. The DTN Node ID is encoded as a certificate Subject Alternative Name (SAN) of type otherName with a name form of BundleEID and as an ACME Identifier type "bundleEID".

## Status of This Memo

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

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

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

This Internet-Draft will expire on 3 September 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
1.1. Scope . . . . .	3
1.2. Authorization Strategy . . . . .	5
1.3. Use of CDDL . . . . .	6
1.4. Terminology . . . . .	7
2. Bundle Endpoint ID ACME Identifier . . . . .	8
2.1. Subsets of Endpoint ID . . . . .	9
3. DTN Node ID Validation . . . . .	9
3.1. DTN Node ID Challenge Request Object . . . . .	12
3.2. DTN Node ID Challenge Response Object . . . . .	13
3.3. ACME Node ID Validation Challenge Bundles . . . . .	14
3.3.1. Challenge Bundle Checks . . . . .	15
3.4. ACME Node ID Validation Response Bundles . . . . .	16
3.4.1. Response Bundle Checks . . . . .	17
3.5. Multi-Perspective Validation . . . . .	18
4. Bundle Integrity Gateway . . . . .	18
5. Certificate Request Profile . . . . .	19
5.1. Multiple Identity Claims . . . . .	19
5.2. Generating Encryption-only or Signing-only Bundle Security Certificates . . . . .	20
6. Implementation Status . . . . .	20
7. Security Considerations . . . . .	21
7.1. Threat: Passive Leak of Validation Data . . . . .	21
7.2. Threat: BP Node Impersonation . . . . .	21
7.3. Threat: Bundle Replay . . . . .	22
7.4. Threat: Denial of Service . . . . .	22
7.5. Inherited Security Considerations . . . . .	23
7.6. Out-of-Scope BP Agent Communication . . . . .	23
8. IANA Considerations . . . . .	23
8.1. ACME Identifier Types . . . . .	23
8.2. ACME Validation Methods . . . . .	24
8.3. Bundle Administrative Record Types . . . . .	24
9. References . . . . .	24
9.1. Normative References . . . . .	24
9.2. Informative References . . . . .	26
Appendix A. Administrative Record Types CDDL . . . . .	28
Appendix B. Example Authorization . . . . .	29

B.1. Challenge Bundle . . . . .	29
B.2. Response Bundle . . . . .	30
Acknowledgments . . . . .	31
Author's Address . . . . .	31

## 1. Introduction

Although the original purpose of the Automatic Certificate Management Environment (ACME) [RFC8555] was to allow Public Key Infrastructure Using X.509 (PKIX) certificate authorities to validate network domain names of clients, the same mechanism can be used to validate any of the subject claims supported by the PKIX profile [RFC5280].

In the case of this specification, the claim being validated is a Subject Alternative Name (SAN) of type `otherName` with a name form of `BundleEID`, which used to represent an Endpoint ID (EID) for a Delay-Tolerant Networking (DTN) bundle. Currently the URI schemes `"dtn"` and `"ipn"` as defined in [RFC9171] are valid for an Endpoint ID. A DTN Node ID is an Endpoint ID with scheme-specific restrictions to identify it as such; currently the `"dtn"` scheme uses an empty demux part and the `"ipn"` scheme uses service number zero.

Because the `BundleEID` claim is new to ACME, a new ACME Identifier type `"bundleEID"` is needed to manage this claim within ACME messaging. A `"bundleEID"` claim can be part of a pre-authorization or as one of the authorizations of an order containing any number of claims.

Once an ACME server validates a Node ID, either as a pre-authorization of the `"bundleEID"` or as one of the authorizations of an order containing a `"bundleEID"`, the client can finalize the order using an associated certificate signing request (CSR). Because a single order can contain multiple identifiers of multiple types, there can be operational issues for a client attempting to, and possibly failing to, validate those multiple identifiers as described in Section 5.1. Once a certificate is issued for a Node ID, how the ACME client configures the Bundle Protocol (BP) agent with the new certificate is an implementation matter.

The scope and behavior of this validation mechanism is similar to that of secured email validation of [RFC8823].

### 1.1. Scope

This document describes the ACME messages, BPv7 payloads, and BPsec requirements needed to validate Node ID ownership. This document does not address:



- \* Mechanisms for communication between ACME client or ACME server and their associated BP agent(s). This document only describes exchanges between ACME client--server pairs and between their BP agents.
- \* Specific BP extension blocks or BPSec security contexts necessary to fulfill the security requirements of this protocol. The exact security context needed, and their parameters, are network-specific.
- \* Policies or mechanisms for defining or configuring bundle integrity gateways, or trusting integrity gateways on an individual entity or across a network.
- \* Mechanisms for locating or identifying other bundle entities (peers) within a network or across an internet. The mapping of Node ID to potential convergence layer (CL) protocol and network address is left to implementation and configuration of the BP Agent and its various potential routing strategies.
- \* Logic for routing bundles along a path toward a bundle's endpoint. This protocol is involved only in creating bundles at a source and handling them at a destination.
- \* Logic for performing rate control and congestion control of bundle transfers. The ACME server is responsible for rate control of validation requests.
- \* Policies or mechanisms for an ACME server to choose a prioritized list of acceptable hash algorithms, or for an ACME client to choose a set of acceptable hash algorithms.
- \* Policies or mechanisms for provisioning, deploying, or accessing certificates and private keys; deploying or accessing certificate revocation lists (CRLs); or configuring security parameters on an individual entity or across a network.
- \* Policies or mechanisms for an ACME server to handle mixed-use certificate requests. This specification is focused only on single-use DTN-specific PKIX profiles.

## 1.2. Authorization Strategy

The basic unit of data exchange in a DTN is a Bundle [RFC9171], which consists of a data payload with accompanying metadata. An Endpoint ID is used as the destination of a Bundle and can indicate both a unicast or a multicast destination. A Node ID is used to identify the source of a Bundle and is used for routing through intermediate nodes, including the final node(s) used to deliver a Bundle to its destination endpoint. A Node ID can also be used as an endpoint for administrative bundles. More detailed descriptions of the rationale and capabilities of these networks can be found in "Delay-Tolerant Network Architecture" [RFC4838].

When an ACME client requests a pre-authorization or an order with a "bundleEID" identifier type having a value consistent with a Node ID (see Section 4.2.5 of [RFC9171]), the ACME server offers a "dtn-nodeid-01" challenge type to validate that Node ID. If the ACME client attempts the authorization challenge to validate a Node ID, the ACME server sends an ACME Node ID Validation Challenge Bundle with a destination of the Node ID being validated. The BP agent on that node receives the Challenge Bundle, generates an ACME key authorization digest, and sends an ACME Node ID Validation Response Bundle in reply. An Integrity Gateway on the client side of the DTN can be used to attest to the source of the Response Bundle. Finally, the ACME server receives the Response Bundle and checks that the digest was generated for the associated ACME challenge and from the client account key associated with the original request. This workflow is shown in the diagram of Figure 1 and defined in Section 3.

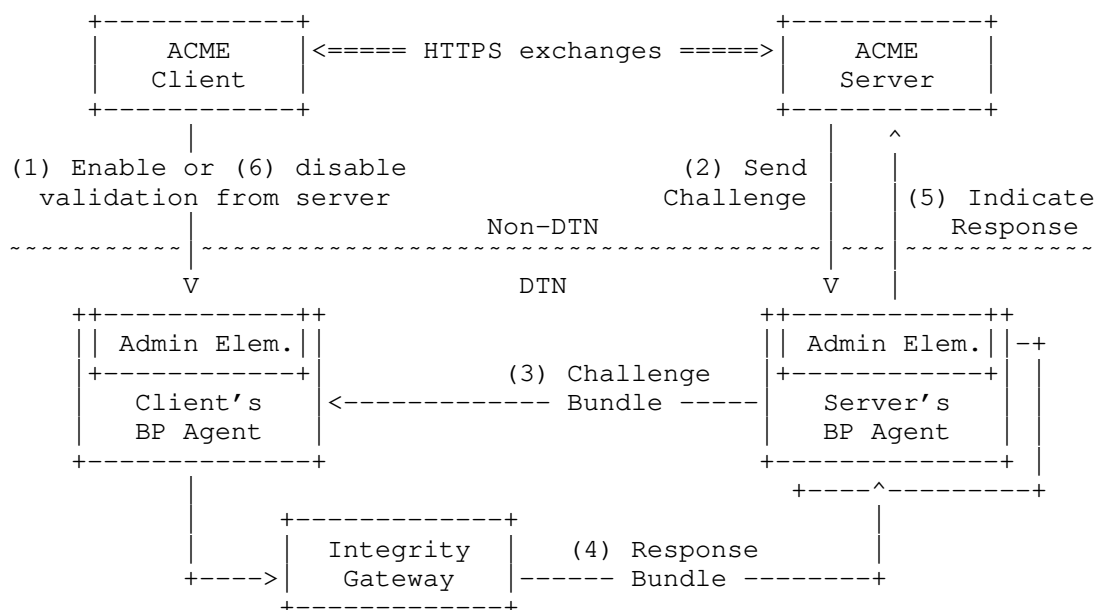


Figure 1: The relationships and flows between Node ID Validation entities

Because the DTN Node ID is used both for routing bundles between BP agents and for multiplexing administrative services within a BP agent, there is no possibility to separate the ACME validation of a Node ID from normal bundle handling for that same Node ID. This leaves administrative record types as a way to leave the Node ID unchanged while disambiguating from other service data bundles.

There is nothing in this protocol which requires network-topological co-location of either the ACME client or ACME server with their associated BP agent. While ACME requires a low-enough latency network to perform HTTPS exchanges between ACME client and server, the client's BP agent (the one being validated) could be on the far side of a long-delay or multi-hop DTN network. The means by which the ACME client or server communicates with its associated BP agent is an implementation matter.

### 1.3. Use of CDDL

This document defines CBOR structure using the Concise Data Definition Language (CDDL) of [RFC8610]. The entire CDDL structure can be extracted from the XML version of this document using the XPath expression:

```
'//sourcecode[@type="cddl"]'
```

The following initial fragment defines the top-level symbols of this document's CDDL, which includes the example CBOR content.

```
start = acme-record / bundle / tstr
```

#### 1.4. Terminology

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.

Because this document combines two otherwise unrelated contexts, ACME and DTN, when a protocol term applies to one of those areas and is used in the other its name is prefixed with either "ACME" or "DTN" respectively. Thus within the ACME context the term is "DTN Node ID" while in the DTN context the name is just "Node ID".

In this document, several terms are shortened for the sake of terseness. These terms are:

**Challenge Request:** This is a shortened form of the full "DTN Node ID Challenge Request Object". It is a JSON object created by the ACME server for challenge type "dtn-nodeid-01".

**Challenge Response:** This is a shortened form of the full "DTN Node ID Challenge Response Object". It is a JSON object created by the ACME client to authorize a challenge type "dtn-nodeid-01".

**Challenge Bundle:** This is a shortened form of the full "ACME Node ID Validation Challenge Bundle". It is a Bundle created by the BP agent managed by the ACME server to challenge a Node ID claim.

**Response Bundle:** This is a shortened form of the full "ACME Node ID Validation Response Bundle". It is a Bundle created by the BP agent managed by the ACME client to validate a Node ID claim.

Because this is an ACME document, the following DTN Bundle Protocol terms are defined here to clarify how they are used by this ACME identifier type and validation mechanism.

**Endpoint ID:** An Endpoint ID is an identifier for the ultimate

destination of a bundle, independent of any intermediate forwarding needed to reach that destination. An endpoint can be a singleton (unicast) or not (anycast or multicast) so an Endpoint ID can also represent a single entity or a set of entities. This is formally defined in Section 4.2.5.1 of [RFC9171].

**Node ID:** A Node ID is a (not guaranteed unique) identifier for a specific node in a network in the form of a singleton Endpoint ID. A single node can have any number of Node IDs but a typical (and expected) form of Node ID is the Administrative Endpoint ID (described below). This is formally defined in Section 4.2.5.2 of [RFC9171].

**Administrative Endpoint ID:** An Administrative Endpoint ID is unique for a node within a specific URI scheme. Although any Node ID can be a valid bundle Source and Destination, the Administrative Endpoint ID is a minimum required Node ID for any node operating in a particular URI scheme. For the "dtn" scheme this is the empty demux part and for the "ipn" scheme this is the service number zero. These is formally defined under Section 4.2.5.1 of [RFC9171].

## 2. Bundle Endpoint ID ACME Identifier

This specification is the first to make use of an Bundle Endpoint ID to identify a claim for a certificate request in ACME. In this document, the only purpose for which an Bundle Endpoint ID ACME identifier is validated is as a DTN Node ID (see Section 3), but other specifications can define challenge types for other Endpoint ID uses.

Identifiers of type "bundleEID" in certificate requests SHALL appear in an extensionRequest attribute [RFC2985] containing a subjectAltName extension of type otherName with a name form of BundleEID, identified by id-on-bundleEID of [IANA-SMI], consistent with the requirements of Section 4.4.2.1 of [RFC9174]. Because the BundleEID is encoded as an IA5String it SHALL be treated as being in the percent-encoded form of Section 2.1 of [RFC3986]. Any "bundleEID" identifier which fails to properly percent-decode SHALL be rejected with an ACME error type of "malformed".

The ACME server SHALL decode and normalize (based on scheme-specific syntax) all received identifiers of type "bundleEID". Any "bundleEID" identifier request which uses a scheme not handled by the ACME server or for which the EID does not match the scheme-specific syntax SHALL be rejected with an ACME error type of "rejectedIdentifier".

When an ACME server needs to request proof that a client controls a BundleEID, it SHALL create an authorization with the identifier type "bundleEID". The ACME server SHALL NOT attempt to dereference the EID value on its own. It is the responsibility of a validation method to ensure the EID ownership via scheme-specific means authorized by the ACME client.

An identifier for the Node ID of "dtn://example/" would be formatted as:

```
{
  "type": "bundleEID",
  "value": "dtn://example/"
}
```

### 2.1. Subsets of Endpoint ID

While the PKIX other name form of BundleEID can hold any Endpoint ID value, the certificate profile used by [RFC9174] and supported by this ACME validation method in Section 3 requires that the value hold a Node ID.

In addition to the narrowing of that certificate profile, this validation method requires that the client's BP agent responds to administrative records sent to the Node ID being validated. This typically is limited to an Administrative Endpoint ID, but there is no prohibition on the administrative element of a BP node from receiving administrative records for, and sending records from, other Node IDs in order to support this validation method.

Neither that certificate profile nor this validation method support operating on non-singleton Endpoint IDs, but other validation methods could be defined to do so in order to support other certificate profiles.

## 3. DTN Node ID Validation

The DTN Node ID validation method proves control over a Node ID by requiring the ACME client to configure a BP agent to respond to specific Challenge Bundles sent from the ACME server. The ACME server validates control of the Node ID by verifying that received Response Bundles correspond with the BP Node and client account key being validated.

Similar to the ACME use case for validating email address ownership [RFC8823], this challenge splits the token into several parts, each being transported by a different channel, and the Key Authorization result requires combining all parts of the token. A separate challenge identifier is used as a filter by BP agents similarly to the challenge "from" email address of [RFC8823].

The token parts are:

token-chal: This token is unique to each ACME authorization. It is contained in the Challenge Object of Section 3.1. Each authorization can consist of multiple Challenge Bundles (e.g. taking different routes), but they all share the same token-chal value. This ensures that the Key Authorization is bound to the specific ACME challenge (and parent ACME authorization). This token does not appear on the BP channel so that any eavesdropper knowing the client's account key thumbprint (e.g. from some other validation method) is not able to impersonate the client.

token-bundle: This token is unique to each Challenge Bundle sent by the ACME server. It is contained in the Challenge Bundle of Section 3.3 and Response Bundle of Section 3.4. This ensures that the Key Authorization is bound to the ability to receive the Challenge Bundle and not just have access to the ACME Challenge Object. This token is also accessible to DTN on-path eavesdroppers.

Because multiple Challenge Bundles can be sent to validate the same Node ID, the token-bundle in the response is needed to correlate with the expected Key Authorization digest.

The DTN Node ID Challenge SHALL only be allowed for an EID usable as a DTN Node ID, which is defined per-scheme in Section 4.2.5.1 of [RFC9171]. When an ACME server supports Node ID validation, the ACME server SHALL define a challenge object in accordance with Section 3.1. Once this challenge object is defined, the ACME client may begin the validation.

To initiate a Node ID validation, the ACME client performs the following steps:

1. The ACME client POSTs a newOrder or newAuthz request including the identifier of type "bundleEID" for the desired Node ID. From either of these entry points an authorization for the "bundleEID" type is indicated by the ACME server. See Section 7.4 of [RFC8555] for more details.

2. The ACME client obtains the id-chal and token-chal from the challenge object in accordance with Section 3.1.
3. The ACME client indicates to the administrative element of its BP agent the id-chal which is authorized for use along with the associated token-chal and client account key thumbprint. The ACME client SHALL wait, if necessary, until the agent is configured before proceeding to the next step.
4. The ACME client POSTs a challenge response to the challenge URL on the ACME server accordance with Section 7.5.1 of [RFC8555]. The payload object is constructed in accordance with Section 3.2.
5. The administrative element waits for a Challenge Bundle to be received with the authorized ACME parameters, including its id-chal payload, in accordance with Section 3.3.
6. The administrative element concatenates token-bundle with token-chal (each as base64url-encoded text strings) and computes the Key Authorization in accordance with Section 8.1 of [RFC8555] using the full token and client account key thumbprint.
7. The administrative element chooses the highest-priority hash algorithm supported by both the client and server, uses that algorithm to compute the digest of the Key Authorization result, and generates a Response Bundle to send back to the ACME server in accordance with Section 3.4.
8. The ACME client waits for the authorization to be finalized on the ACME server in accordance with Section 7.5.1 of [RFC8555].
9. Once the challenge is completed (successfully or not), the ACME client indicates to the BP agent that the id-chal is no longer usable. If the authorization fails, the ACME client MAY retry the challenge from Step 3.

The ACME server verifies the client's control over a Node ID by performing the following steps:

1. The ACME server receives a newOrder or newAuthz request including the identifier of type "bundleEID", where the URI value is a Node ID.
2. The ACME server generates an authorization for the Node ID with challenge type "dtn-nodeid-01" in accordance with Section 3.1.



3. The ACME server receives a POST to the challenge URL indicated from the authorization object. The payload is handled in accordance with Section 3.2.
4. The ACME server sends, via the administrative element of its BP agent, one or more Challenge Bundles in accordance with Section 3.3. Each challenge bundle SHALL contain a distinct token-bundle to be able to correlate with a response bundle. Computing an expected Key Authorization digest is not necessary until a response is received with a chosen hash algorithm.
5. The ACME server waits for Response Bundle(s) for a limited interval of time (based on the challenge response object of Section 3.2). Responses are encoded in accordance with Section 3.4.
6. Once received and decoded, the ACME server checks the contents of each Response Bundle in accordance with Section 3.4.1. After all Challenge Bundles have either been responded to or timed-out, the ACME server policy (see Section 3.5) determines whether the validation is successful. If validation is not successful, a client may retry the challenge which starts in Step 3.

When responding to a Challenge Bundle, a BP agent SHALL send a single Response Bundle in accordance with Section 3.4. A BP agent SHALL respond to ACME challenges only within the interval of time and only for the id-chal indicated by the ACME client. A BP agent SHALL respond to multiple Challenge Bundles with the same ACME parameters but different bundle identities (source Node ID and timestamp); these correspond with the ACME server validating via multiple routing paths.

### 3.1. DTN Node ID Challenge Request Object

The DTN Node ID Challenge request object is defined by the ACME server when it supports validating Node IDs.

The DTN Node ID Challenge request object has the following content:

type (required, string): The string "dtn-nodeid-01".

id-chal (required, string): This is a random value associated with a challenge which allows a client to filter valid Challenge Bundles. The same value is used for all Challenge Bundles associated with an ACME challenge, including multi-perspective validation using multiple sources as described in Section 3.5. This value SHALL have at least 128 bits of entropy. It SHALL NOT contain any characters outside the base64url alphabet as described in

Section 5 of [RFC4648]. Trailing '=' padding characters SHALL be stripped. See [RFC4086] for additional information on randomness requirements.

token-chal (required, string): This is a random value, used as part of the Key Authorization algorithm, which is communicated to the ACME client only over the ACME channel. This value SHALL have at least 128 bits of entropy. It SHALL NOT contain any characters outside the base64url alphabet as described in Section 5 of [RFC4648]. Trailing '=' padding characters SHALL be stripped. See [RFC4086] for additional information on randomness requirements.

```
{
  "type": "dtn-nodeid-01",
  "url": "https://example.com/acme/chall/prV_B7yEyA4",
  "id-chal": "dDtaviYTPUWFS3NK37YWfQ",
  "token-chal": "tPUZNY4ONIk6LxErRFEjVw"
}
```

The token-chal value included in this object applies to the entire challenge, and may correspond with multiple separate token-bundle values when multiple Challenge Bundles are sent for a single validation.

### 3.2. DTN Node ID Challenge Response Object

The DTN Node ID Challenge response object is defined by the ACME client when it authorizes validation of a Node ID. Because a DTN has the potential for significantly longer delays than a non-DTN network, the ACME client is able to inform the ACME server if a particular validation round-trip is expected to take longer than normal network delays (on the order of seconds).

The DTN Node ID Challenge response object has the following content:

rtt (optional, number): An expected round-trip time (RTT), in seconds, between sending a Challenge Bundle and receiving a Response Bundle. This value is a hint to the ACME server for how long to wait for responses but is not authoritative. The minimum RTT value SHALL be zero. There is no special significance to zero-value RTT, it simply indicates the response is expected in less than the least significant unit used by the ACME client.

```
{
  "rtt": 300.0
}
```

A challenge response is not sent until the BP agent has been configured to properly respond to the challenge, so the RTT value is meant to indicate any node-specific path delays expected to be encountered from the ACME server. Because there is no requirement on the path (or paths) which bundles may traverse between the ACME server and the BP agent, and the ACME server can attempt some path diversity, the RTT value SHOULD be pessimistic.

A default bundle response interval, used when the object does not contain an RTT, SHOULD be a configurable parameter of the ACME server. If the ACME client indicated an RTT value in the object, the response interval SHOULD be twice the RTT (with limiting logic applied as described below). The lower limit on response interval is network-specific, but SHOULD NOT be shorter than one second. The upper limit on response interval is network-specific, but SHOULD NOT be longer than one minute (60 seconds) for a terrestrial-only DTN.

### 3.3. ACME Node ID Validation Challenge Bundles

Each ACME Node ID Validation Challenge Bundle SHALL be structured and encoded in accordance with [RFC9171].

Each Challenge Bundle has parameters as listed here:

**Bundle Processing Control Flags:** The primary block flags SHALL indicate that the payload is an administrative record. The primary block flags SHALL indicate that user application acknowledgement is requested; this flag distinguishes the Challenge Bundle from the Response Bundle.

**Destination EID:** The Destination EID SHALL be the ACME-server-normalized Node ID being validated.

**Source Node ID:** The Source Node ID SHALL indicate the Node ID of a BP agent of the ACME server performing the challenge.

**Creation Timestamp and Lifetime:** The Creation Timestamp SHALL be set to the time at which the challenge was generated. The Lifetime SHALL indicate the response interval (of Section 3.2) for which ACME server will accept responses to this challenge.

**Administrative Record Type Code:** Set to the ACME Node ID Validation type code defined in Section 8.3.

**Administrative Record Content:** The Challenge Bundle administrative record content SHALL consist of a CBOR map containing two pairs:

- \* One pair SHALL consist of key 1 with value of ACME challenge id-chal, copied from the challenge object, represented as a CBOR byte string.
- \* One pair SHALL consist of key 2 with value of ACME challenge token-bundle, represented as a CBOR byte string. The token-bundle is a random value that uniquely identifies the challenge bundle. This value SHALL have at least 128 bits of entropy. See [RFC4086] for additional information on randomness requirements.
- \* One pair SHALL consist of key 4 with value of an array containing acceptable hash algorithm identifiers. The array SHALL be ordered in descending preference, with the first item being the most preferred. The array SHALL contain at least one item. Each algorithm identifier SHALL correspond to the Value column (integer or text string) of the algorithm registered in the "COSE Algorithms" registry of [IANA-COSE].

This structure is part of the extension CDDL in Appendix A. An example full Challenge Bundle is included in Appendix B.1

For interoperability, entities which use this validation method SHALL support the hash algorithm "SHA-256" of [I-D.ietf-cose-hash-algs], but can use other hash algorithms. This requirement allows for different implementations to be configured to use an interoperable algorithm, but does not preclude the use of other algorithms.

If the BP agent generating a Challenge Bundle does not have a well-synchronized clock or the agent responding to the challenge is expected to not have a well-synchronized clock, the bundle SHALL include a Bundle Age extension block.

Challenge Bundles SHALL include a Block Integrity Block (BIB) in accordance with Section 4 or with a Security Source identical to the bundle Source Node. Challenge Bundles SHALL NOT be directly encrypted by Block Confidentiality Block (BCB) or any other method (see Section 7.1).

### 3.3.1. Challenge Bundle Checks

A proper Challenge Bundle meets all of the following criteria:

- \* The Challenge Bundle was received within the time interval allowed for the challenge. The allowed interval starts at the Creation Timestamp and extends for the Lifetime of the Challenge Bundle.

- \* The Challenge Bundle contains a BIB which covers at least the primary block and payload. That BIB has a security source which is trusted and passes security-context-specific validation (i.e. MAC or signature verification).
- \* The challenge payload contains the id-chal as indicated in the ACME challenge object. The challenge payload contains a token-bundle meeting the definition in Section 3.3. The challenge payload contains at least one hash algorithm identifier acceptable to the client.

Any of the failures above SHALL cause the challenge bundle to be deleted and otherwise ignored by the BP agent.

### 3.4. ACME Node ID Validation Response Bundles

Each ACME Node ID Validation Response Bundle SHALL be structured and encoded in accordance with [RFC9171].

Each Response Bundle has parameters as listed here:

**Bundle Processing Control Flags:** The primary block flags SHALL indicate that the payload is an administrative record. The primary block flags SHALL NOT indicate that user application acknowledgement is requested; this flag distinguishes the Response Bundle from the Challenge Bundle.

**Destination EID:** The Destination EID SHALL be identical to the Source Node ID of the Challenge Bundle to which this response corresponds.

**Source Node ID:** The Source Node ID SHALL be identical to the Destination EID of the Challenge Bundle to which this response corresponds.

**Creation Timestamp and Lifetime:** The Creation Timestamp SHALL be set to the time at which the response was generated. The response Lifetime SHALL indicate the response interval remaining if the Challenge Bundle indicated a limited Lifetime.

**Administrative Record Type Code:** Set to the ACME Node ID Validation type code defined in Section 8.3.

**Administrative Record Content:** The Response Bundle administrative record content SHALL consist of a CBOR map containing three pairs:

- \* One pair SHALL consist of key 1 with value of ACME challenge id-chal, copied from the Request Bundle, represented as a CBOR byte string.
- \* One pair SHALL consist of key 2 with value of ACME challenge token-bundle, copied from the Request Bundle, represented as a CBOR byte string.
- \* One pair SHALL consist of key 3 with value of a two-element array containing the pair of a hash algorithm identifier and the hash byte string. The algorithm identifier SHALL correspond to the Value column (integer or text string) of the algorithm registered in the "COSE Algorithms" registry of [IANA-COSE].

This structure is part of the extension CDDL in Appendix A. An example full Response Bundle is included in Appendix B.2

If the BP agent responding to a Challenge Bundle does not have a well-synchronized clock, it SHALL use any information about last-hop delays and (if present) Bundle Age extension data to infer the age of the Challenge Bundle and lifetime of the Response Bundle.

Response Bundles SHALL include a BIB in accordance with Section 4. Response Bundles SHALL NOT be directly encrypted by BCB or any other method (see Section 7.1 for explanation).

#### 3.4.1. Response Bundle Checks

A proper Response Bundle meets all of the following criteria:

- \* The Response Bundle was received within the time interval allowed for the challenge. The allowed interval starts at the Creation Timestamp and extends for the Lifetime of the associated Challenge Bundle. The interval of the Challenge Bundle is used here because the interval of the Response Bundle could be made to disagree with the Challenge Bundle.
- \* The Response Bundle Source Node ID is identical to the Node ID being validated. The comparison of Node IDs SHALL use the comparison logic of the NODE-ID from Section 4.4.1 of [RFC9174].
- \* The Response Bundle contains a BIB which covers at least the primary block and payload. That BIB has a security source which is trusted and passes security-context-specific validation.

- \* The response payload contains the same id-chal and token-bundle as sent in the Challenge Bundle (this is also how the two bundles are correlated). The response payload contains a hash algorithm identifier acceptable to the server (as indicated in the challenge bundle). The response payload contains the expected Key Authorization digest computed by the ACME server.

Any of the failures above SHALL cause that single-perspective validation to fail. Any of the failures above SHOULD be distinguished as subproblems to the ACME client. The lack of a response within the expected response interval, as defined in Section 3.2, SHALL also be treated as a validation failure.

### 3.5. Multi-Perspective Validation

To avoid possible on-path attacks in certain networks, an ACME server can perform a single validation using multiple challenge bundle sources or via multiple routing paths. This technique is called multi-perspective validation as recommended in Section 10.2 of [RFC8555] and an implementation used by Let's Encrypt is described in [LE-multi-perspective].

When required by policy, an ACME server SHALL send multiple challenge bundles from different sources in the DTN network. When multiple Challenge Bundles are sent for a single validation, it is a matter of ACME server policy to determine whether or not the validation as a whole is successful. The result of each single-source validation is defined as success or failure in Section 3.4.1.

A RECOMMENDED validation policy is to succeed if the challenge from a primary bundle source is successful and if there are no more than one failure from a secondary source. The determination of which perspectives are considered primary or secondary is an implementation matter.

Regardless of whether a validation is single- or multi-perspective, a validation failure SHALL be indicated by an ACME error type of "incorrectResponse". Each specific perspective failure SHOULD be indicated to the ACME client as a validation subproblem.

## 4. Bundle Integrity Gateway

This section defines a BIB use which closely resembles the function of DKIM email signing [RFC6376]. In this mechanism a routing node in a DTN sub-network attests to the origination of a bundle by adding a BIB before forwarding it. The bundle receiver then need not trust the source of the bundle, but only trust this security source node. The receiver needs policy configuration to know which security

sources are permitted to attest for which bundle sources.

An integrity gateway SHALL validate the Source Node ID of a bundle, using local-network-specific means, before adding a BIB to the bundle. The exact means by which an integrity gateway validates a bundle's source is network-specific, but could use physical-layer, network-layer or BP-convergence-layer authentication. The bundle source could also add its own BIB with a local-network-specific security context or local-network-specific key material (i.e. a group key shared within the local network).

When an integrity gateway adds a BIB it SHALL be in accordance with [RFC9172]. The BIB targets SHALL cover both the payload block and the primary block (either directly as a target or as additional authenticated data for the payload block MAC/signature). The Security Source of this BIB SHALL be either the bundle source Node ID itself or a routing node trusted by the destination node (see Section 7.2).

## 5. Certificate Request Profile

The ultimate purpose of this ACME validation is to allow a CA to issue certificates following the profiles of Section 4.4.2 of [RFC9174], [I-D.sipos-dtn-udpcl], and [I-D.bsipos-dtn-bpsec-cose]. These purposes are referred to here as bundle security certificates.

One defining aspect of bundle security certificates is the Extended Key Usage key purpose `id-kp-bundleSecurity` of [IANA-SMI]. When requesting a certificate which includes a Node ID SAN, the CSR SHOULD include an Extended Key Usage of `id-kp-bundleSecurity`. When a bundle security certificate is issued based on a validated Node ID SAN, the certificate SHALL include an Extended Key Usage of `id-kp-bundleSecurity`.

### 5.1. Multiple Identity Claims

A single bundle security CSR MAY contain a mixed set of SAN claims, including combinations of "ip", "dns", and "bundleEID" claims. There is no restriction on how a certificate combines these claims, but each claim SHALL be validated by an ACME server to issue such a certificate as part of an associated ACME order. This is no different than the existing behavior of [RFC8555] but is mentioned here to make sure that CA policy handles such situations; especially related to validation failure of an identifier in the presence of multiple identifiers. The initial "ip" validations are defined in [RFC8738] and initial "dns" validations are defined in [RFC8555] and [RFC8737]. The specific use case of [RFC9174] allows, and for some network policies requires, that a certificate authenticate both the



DNS name of an entity as well as the Node ID of the entity.

## 5.2. Generating Encryption-only or Signing-only Bundle Security Certificates

ACME extensions specified in this document can be used to request encryption-only or signing-only bundle security certificates.

In order to request signing only bundle security certificate, the CSR SHALL include the key usage extension with `digitalSignature` and/or `nonRepudiation` bits set and no other bits set.

In order to request encryption only bundle security certificate, the CSR SHALL 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 key usage extension in the CSR, signals to ACME server to issue a bundle security certificate suitable for both signing and encryption.

## 6. Implementation Status

This section is to be removed before publishing as an RFC.

[NOTE to the RFC Editor: please remove this section before publication, as well as the reference to [RFC7942] and [github-dtn-demo-agent] and [github-dtn-wireshark].]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations can exist.

An example implementation of the this draft of ACME Node ID Validation has been created as a GitHub project [github-dtn-demo-agent] and is intended to use as a proof-of-concept and as a possible source of interoperability testing.

A Wireshark dissector for of the this draft of ACME Node ID Validation has been created as a GitHub project [github-dtn-wireshark] and is intended to be used to inspect and troubleshoot implementations.

## 7. Security Considerations

This section separates security considerations into threat categories based on guidance of BCP 72 [RFC3552].

### 7.1. Threat: Passive Leak of Validation Data

Because this challenge mechanism is used to bootstrap security between DTN Nodes, the challenge and its response are likely to be transferred in plaintext. The only ACME data present on-the-wire is a random token and a cryptographic digest, so there is no sensitive data to be leaked within the Node ID Validation bundle exchange. Because each challenge uses a separate token pair, there is no value in an on-path attacker seeing the tokens from past challenges and/or responses.

It is possible for intermediate BP nodes to encapsulate-and-encrypt Challenge and/or Response Bundles while they traverse untrusted networks, but that is a DTN configuration matter outside of the scope of this document.

### 7.2. Threat: BP Node Impersonation

As described in Section 8.1 of [RFC8555], it is possible for an active attacker to alter data on both ACME client channel and the DTN validation channel.

The primary mitigation is to delegate bundle integrity sourcing to a trusted routing node near, in the sense of bundle routing topology, to the bundle source node as defined in Section 4. This is functionally similar to DKIM signing of [RFC6376] and provides some level of bundle origination.

Another way to mitigate single-path on-path attacks is to attempt validation of the same Node ID from multiple sources or via multiple bundle routing paths, as defined in Section 3.5. It is not a trivial task to guarantee bundle routing though, so more advanced techniques such as onion routing (using bundle-in-bundle encapsulation [I-D.ietf-dtn-bibect]) could be employed.

### 7.3. Threat: Bundle Replay

It is possible for an on-path attacker to replay both Challenge Bundles or Response Bundles. Even in a properly-configured DTN it is possible that intermediate bundle routers to use multicast forwarding of a unicast-destination bundle.

Ultimately, the point of the ACME bundle exchange is to derive a Key Authorization and its cryptographic digest and communicate it back to the ACME server for validation, so the uniqueness of the Key Authorization directly determines the scope of replay validity. The uniqueness of each token-bundle to each challenge bundle ensures that the Key Authorization is unique to the challenge bundle. The uniqueness of each token-chal to the ACME challenge ensures that the Key Authorization is unique to that ACME challenge and not based solely on values visible to on-path eavesdroppers.

Having each bundle's primary block and payload block covered by a BIB from a trusted security source (see Section 4) ensures that a replayed bundle cannot be altered in the blocks used by ACME. All together, these properties mean that there is no degraded security caused by replay of either a Challenge Bundle or a Response Bundle even in the case where the primary or payload block is not covered by a BIB. The worst that can come of bundle replay is the waste of network resources as described in Section 7.4.

### 7.4. Threat: Denial of Service

The behaviors described in this section all amount to a potential denial-of-service to a BP agent.

A malicious entity can continually send Challenge Bundles to a BP agent. The victim BP agent can ignore Challenge Bundles which do not conform to the specific time interval and challenge token for which the ACME client has informed the BP agent that challenges are expected. The victim BP agent can require all Challenge Bundles to be BIB-signed to ensure authenticity of the challenge.

A malicious entity can continually send Response Bundles to a BP agent. The victim BP agent can ignore Response Bundles which do not conform to the specific time interval or Source Node ID or challenge token for an active Node ID validation.

Similar to other validation methods, an ACME server validating a DTN Node ID could be used as a denial of service amplifier. For this reason any ACME server can rate-limit validation activities for individual clients and individual certificate requests.

### 7.5. Inherited Security Considerations

Because this protocol relies on ACME for part of its operation, the security considerations of [RFC8555] apply to all ACME client--server exchanges during Node ID validation.

Because this protocol relies on BPv7 for part of its operation, the security considerations of [RFC9171] and [RFC9172] apply to all BP messaging during Node ID validation.

### 7.6. Out-of-Scope BP Agent Communication

Although messaging between an ACME client or ACME server and its associated BP agent are out-of-scope for this document, both of those channels are critical to Node ID validation security. Either channel can potentially leak data or provide attack vectors if not properly secured. These channels need to protect against spoofing of messaging in both directions to avoid interruption of normal validation sequencing and to prevent false validations from succeeding.

The ACME server and its BP agent exchange the outgoing id-chal, token-bundle, and Key Authorization digest but these values do not need to be confidential (they are also in plaintext over the BP channel).

Depending on implementation details, the ACME client might transmit the client account key thumbprint to its BP agent to allow computing the Key Authorization digest on the BP agent. If an ACME client does transmit its client account key thumbprint to a BP agent, it is important that this data is kept confidential because it provides the binding of the client account key to the Node ID validation (as well as for all other types of ACME validation). Avoiding this transmission would require a full round-trip between BP agent and ACME client, which can be undesirable if the two are separated by a long-delay network.

## 8. IANA Considerations

This specification adds to the ACME registry and BP registry for this behavior.

### 8.1. ACME Identifier Types

Within the "Automated Certificate Management Environment (ACME) Protocol" registry [IANA-ACME], the following entry has been added to the "ACME Identifier Types" sub-registry.

Label	Reference
bundleEID	This specification and [RFC3986]

Table 1

## 8.2. ACME Validation Methods

Within the "Automated Certificate Management Environment (ACME) Protocol" registry [IANA-ACME], the following entry has been added to the "ACME Validation Methods" sub-registry.

Label	Identifier Type	ACME	Reference
dtm-nodeid-01	bundleEID	Y	This specification

Table 2

## 8.3. Bundle Administrative Record Types

Within the "Bundle Protocol" registry [IANA-BP], the following entries have been added to the "Bundle Administrative Record Types" sub-registry.

[NOTE to the RFC Editor: For [RFC5050] compatibility the AR-TBD value needs to be no larger than 15, but such compatibility is not needed. For BPbis the AR-TBD value needs to be no larger than 65535 as defined by [I-D.sipos-dtn-bpv7-admin-iana].]

Bundle Protocol Version	Value	Description	Reference
7	AR-TBD	ACME Node ID Validation	This specification

Table 3

## 9. References

### 9.1. Normative References

- [IANA-ACME] IANA, "Automated Certificate Management Environment (ACME) Protocol", <<https://www.iana.org/assignments/acme/>>.
- [IANA-BP] IANA, "Bundle Protocol", <<https://www.iana.org/assignments/bundle/>>.
- [IANA-COSE] IANA, "CBOR Object Signing and Encryption (COSE)", <<https://www.iana.org/assignments/cose/>>.
- [IANA-SMI] IANA, "Structure of Management Information (SMI) Numbers", <<https://www.iana.org/assignments/smi-numbers/>>.
- [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>>.
- [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>>.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <<https://www.rfc-editor.org/info/rfc4086>>.
- [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>>.
- [RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, DOI 10.17487/RFC4838, April 2007, <<https://www.rfc-editor.org/info/rfc4838>>.

- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [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>>.
- [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>>.
- [RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <<https://www.rfc-editor.org/info/rfc8610>>.
- [RFC9171] Burleigh, S., Fall, K., Birrane, E., and , "Bundle Protocol Version 7", RFC 9171, DOI 10.17487/RFC9171, January 2022, <<https://www.rfc-editor.org/info/rfc9171>>.
- [RFC9172] Birrane, E., and K. McKeever, "Bundle Protocol Security (BPsec)", RFC 9172, DOI 10.17487/RFC9172, January 2022, <<https://www.rfc-editor.org/info/rfc9172>>.
- [I-D.ietf-cose-hash-algs]  
Schaad, J., "CBOR Object Signing and Encryption (COSE): Hash Algorithms", Work in Progress, Internet-Draft, draft-ietf-cose-hash-algs-09, 14 September 2020, <<https://datatracker.ietf.org/doc/html/draft-ietf-cose-hash-algs-09>>.

## 9.2. Informative References

- [RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, DOI 10.17487/RFC5050, November 2007, <<https://www.rfc-editor.org/info/rfc5050>>.

- [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>>.
- [RFC8737] Shoemaker, R.B., "Automated Certificate Management Environment (ACME) TLS Application-Layer Protocol Negotiation (ALPN) Challenge Extension", RFC 8737, DOI 10.17487/RFC8737, February 2020, <<https://www.rfc-editor.org/info/rfc8737>>.
- [RFC8738] Shoemaker, R.B., "Automated Certificate Management Environment (ACME) IP Identifier Validation Extension", RFC 8738, DOI 10.17487/RFC8738, February 2020, <<https://www.rfc-editor.org/info/rfc8738>>.
- [RFC8823] Melnikov, A., "Extensions to Automatic Certificate Management Environment for End-User S/MIME Certificates", RFC 8823, DOI 10.17487/RFC8823, April 2021, <<https://www.rfc-editor.org/info/rfc8823>>.
- [RFC9174] Sipos, B., Demmer, M., Ott, J., and S. Perreault, "Delay-Tolerant Networking TCP Convergence-Layer Protocol Version 4", RFC 9174, DOI 10.17487/RFC9174, January 2022, <<https://www.rfc-editor.org/info/rfc9174>>.
- [I-D.ietf-dtn-bibect]  
Burleigh, S., "Bundle-in-Bundle Encapsulation", Work in Progress, Internet-Draft, draft-ietf-dtn-bibect-03, 18 February 2020, <<https://datatracker.ietf.org/doc/html/draft-ietf-dtn-bibect-03>>.
- [I-D.sipos-dtn-udpcl]  
Sipos, B., "Delay-Tolerant Networking UDP Convergence Layer Protocol", Work in Progress, Internet-Draft, draft-sipos-dtn-udpcl-01, 26 March 2021, <<https://datatracker.ietf.org/doc/html/draft-sipos-dtn-udpcl-01>>.
- [I-D.sipos-dtn-bpv7-admin-iana]  
Sipos, B., "Bundle Protocol Version 7 Administrative Record Types Registry", Work in Progress, Internet-Draft, draft-sipos-dtn-bpv7-admin-iana-01, 2 March 2022, <<https://datatracker.ietf.org/doc/html/draft-sipos-dtn-bpv7-admin-iana-01>>.



[I-D.bsipos-dtn-bpsec-cose]  
 Sipos, B., "DTN Bundle Protocol Security COSE Security Context", Work in Progress, Internet-Draft, draft-bsipos-dtn-bpsec-cose-06, 3 June 2021,  
 <<https://datatracker.ietf.org/doc/html/draft-bsipos-dtn-bpsec-cose-06>>.

[github-dtn-demo-agent]  
 Sipos, B., "Python implementation of basic BPv7-related protocols",  
 <<https://github.com/BrianSipos/dtn-demo-agent/>>.

[github-dtn-wireshark]  
 Sipos, B., "Wireshark Dissectors for BPv7-related Protocols",  
 <<https://github.com/BrianSipos/dtn-wireshark/>>.

[LE-multi-perspective]  
 Aas, J., McCarney, D., and R. Shoemaker, "Multi-Perspective Validation Improves Domain Validation Security", 19 February 2020,  
 <<https://letsencrypt.org/2020/02/19/multi-perspective-validation.html>>.

## Appendix A. Administrative Record Types CDDL

[NOTE to the RFC Editor: The "0xFFFF" in this CDDL is replaced by the "ACME Node ID Validation" administrative record type code.]

The CDDL extension of BP [RFC9171] for the ACME bundles is:

```
; All ACME records have the same structure
$admin-record /= [0xFFFF, acme-record]
acme-record = {
    id-chal,
    token-bundle,
    ? key-auth-digest ; present in Response Bundles
    ? alg-list ; present in Challenge Bundles
}
id-chal = (1 => bstr)
token-bundle = (2 => bstr)
key-auth-digest = (3 => [
    alg: alg-id,
    value: bstr
])
alg-list = (4 => [+ alg-id])
# From the IANA registry, only hash algorithms allowed
alg-id: tstr / int
```

## Appendix B. Example Authorization

[NOTE to the RFC Editor: The "0xFFFF" in these examples are replaced by the "ACME Node ID Validation" administrative record type code.]

This example is a bundle exchange for the ACME server with Node ID "dtn://acme-server/" performing a verification for ACME client Node ID "dtn://acme-client/". The example bundles use no block CRC or BPsec integrity, which is for simplicity and is not recommended for normal use. The provided figures are extended diagnostic notation [RFC8610].

For this example the ACME client key thumbprint has the base64url encoded value of:

"LPJNul-wow4m6DsqxnbhnsWHlwfp0JecwQzYpOLmCQ"

And the ACME-server generated token-chal has the base64url-encoded value of:

"tPUZNY4ONIk6LxErRFEjVw"

## B.1. Challenge Bundle

For the single challenge bundle in this example, the token-bundle (transported as byte string via BP) has the base64url-encoded value of:

"p3yRYFU4KxwQaHQjJ2RdiQ"

The minimal-but-valid Challenge Bundle is shown in Figure 2. This challenge requires that the ACME client respond within a 60 second time window.

```

[
  [
    7, / BP version /
    0x22, / flags: user-app-ack, payload-is-an-admin-record /
    0, / CRC type: none /
    [1, "//acme-client/"], / destination /
    [1, "//acme-server/"], / source /
    [1, 0], / report-to: none /
    [1000000, 0], / timestamp: 2000-01-01T00:16:40+00:00 /
    60000 / lifetime: 60s /
  ],
  [
    1, / block type code /
    1, / block number /
    0, / flags /
    0, / CRC type: none /
    <<[ / type-specific data /
      0xFFFF, / record-type-code /
      { / record-content /
        1: b64'dDtaviYTPUWFS3NK37YWfQ', / id-chal /
        2: b64'p3yRYFU4KxwQaHQjJ2RdiQ' / token-bundle /
      }
    ]>>
  ]
]

```

Figure 2: Example Challenge Bundle

## B.2. Response Bundle

When the tokens are combined with the key thumbprint, the full Key Authorization value (a single string split across lines for readability) is:

```

"p3yRYFU4KxwQaHQjJ2RdiQtPUZNY4ONIk6LxErRFEjVw."
"LPJNul-wow4m6DsrxbninhSWHlwfp0JecwQzYpOLmCQ"

```

The minimal-but-valid Response Bundle is shown in Figure 3. This response indicates that there is 30 seconds remaining in the response time window.

```

[
  [
    7, / BP version /
    0x02, / flags: payload-is-an-admin-record /
    0, / CRC type: none /
    [1, "//acme-server/"], / destination /
    [1, "//acme-client/"], / source /
    [1, 0], / report-to: none /
    [1030000, 0], / timestamp: 2000-01-01T00:17:10+00:00 /
    30000 / lifetime: 30s /
  ],
  [
    1, / block type code /
    1, / block number /
    0, / flags /
    0, / CRC type: none /
    <<[ / block-type-specific data /
      0xFFFF, / record-type-code /
      { / record-content /
        1: b64'dDtaviYTPUWFS3NK37YWfQ', / id-chal /
        2: b64'p3yRYFU4KxwQaHQjJ2RdiQ', / token-bundle /
        3: b64'mVIOJEQZie8XpYM6MMVSQUiNPH64URnhM9niJ5XHrew'
        / key auth. digest /
      }
    ]>>
  ]
]

```

Figure 3: Example Response Bundle

#### Acknowledgments

This specification is based on DTN use cases related to PKIX certificate issuance.

The workflow and terminology of this validation method was originally copied from the work of Alexey Melnikov in [RFC8823].

#### Author's Address

Brian Sipos  
 RKF Engineering Solutions, LLC  
 7500 Old Georgetown Road  
 Suite 1275  
 Bethesda, MD 20814-6198  
 United States of America  
 Email: brian.sipos+ietf@gmail.com

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: June 20, 2022

O. Friel  
R. Barnes  
Cisco  
R. Shekh-Yusef  
Auth0  
M. Richardson  
Sandelman Software Works  
December 17, 2021

ACME Integrations  
draft-ietf-acme-integrations-06

Abstract

This document outlines multiple advanced use cases and integrations that ACME facilitates without any modifications or enhancements required to the base ACME specification. The use cases include ACME integration with EST, BRSKI and TEAP.

Status of This Memo

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

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

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

This Internet-Draft will expire on June 20, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. ACME Integration with EST . . . . .	4
4. ACME Integration with BRSKI . . . . .	7
5. ACME Integration with BRSKI Default Cloud Registrar . . . . .	9
6. ACME Integration with TEAP . . . . .	11
7. ACME Integration Considerations . . . . .	15
7.1. Service Operators . . . . .	15
7.2. CSR Attributes . . . . .	15
7.3. Certificate Chains and Trust Anchors . . . . .	15
7.3.1. EST /cacerts . . . . .	16
7.3.2. TEAP PKCS#7 TLV . . . . .	16
7.4. id-kp-cmcRA . . . . .	16
7.5. Error Handling . . . . .	17
8. IANA Considerations . . . . .	17
9. Security Considerations . . . . .	17
9.1. Denial of Service against ACME infrastructure . . . . .	18
10. Informative References . . . . .	19
Authors' Addresses . . . . .	21

## 1. Introduction

ACME [RFC8555] defines a protocol that a certification authority (CA) and an applicant can use to automate the process of domain name ownership validation and X.509 (PKIX) certificate issuance. The protocol is rich and flexible and enables multiple use cases that are not immediately obvious from reading the specification. This document explicitly outlines multiple advanced ACME use cases including:

- o ACME integration with EST [RFC7030]
- o ACME integration with BRSKI [RFC8995]
- o ACME integration with BRSKI Default Cloud Registrar [I-D.ietf-anima-brski-cloud]
- o ACME integration with TEAP [RFC7170] and TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski]

The integrations with EST, BRSKI (which is based upon EST), and TEAP enable automated certificate enrollment for devices.

ACME for subdomains [I-D.ietf-acme-subdomains] outlines how ACME can be used by a client to obtain a certificate for a subdomain identifier from an ACME server where the client has fulfilled a challenge against a parent domain, but does not need to fulfil a challenge against the explicit subdomain. This is a useful optimization when ACME is used to issue certificates for large numbers of devices as it reduces the domain ownership proof traffic (DNS or HTTP) and ACME traffic overhead, but is not a necessary requirement.

## 2. Terminology

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.

The following terms are defined in DNS Terminology [RFC8499] and are reproduced here:

- o Label: An ordered list of zero or more octets that makes up a portion of a domain name. Using graph theory, a label identifies one node in a portion of the graph of all possible domain names.
- o Domain Name: An ordered list of one or more labels.
- o Subdomain: "A domain is a subdomain of another domain if it is contained within that domain. This relationship can be tested by seeing if the subdomain's name ends with the containing domain's name." (Quoted from [RFC1034], Section 3.1) For example, in the host name "nnn.mmm.example.com", both "mmm.example.com" and "nnn.mmm.example.com" are subdomains of "example.com". Note that the comparisons here are done on whole labels; that is, "ooo.example.com" is not a subdomain of "oo.example.com".
- o Fully-Qualified Domain Name (FQDN): This is often just a clear way of saying the same thing as "domain name of a node", as outlined above. However, the term is ambiguous. Strictly speaking, a fully-qualified domain name would include every label, including the zero-length label of the root: such a name would be written "www.example.net." (note the terminating dot). But, because every name eventually shares the common root, names are often written relative to the root (such as "www.example.net") and are still called "fully qualified". This term first appeared in [RFC0819]. In this document, names are often written relative to the root.

The following terms are used in this document:

- o BRSKI: Bootstrapping Remote Secure Key Infrastructures [RFC8995]
- o Certification Authority (CA): An organization that is responsible for the creation, issuance, revocation, and management of Certificates. The term applies equally to both Roots CAs and Subordinate CAs
- o CMS: Cryptographic Message Syntax [RFC5652]
- o CMC: Certificate Management over CMS [RFC5272]
- o CSR: Certificate Signing Request
- o EST: Enrollment over Secure Transport [RFC7030]
- o MASA: Manufacturer Authorized Signing Authority as defined in [RFC8995]
- o RA: PKI Registration Authority
- o TEAP: Tunnelled Extensible Authentication Protocol [RFC7170]
- o TLV: Type-Length-Value format defined in TEAP

### 3. ACME Integration with EST

EST [RFC7030] defines a mechanism for clients to enroll with a PKI Registration Authority by sending CMC messages over HTTP. EST section 1 states:

"Architecturally, the EST service is located between a Certification Authority (CA) and a client. It performs several functions traditionally allocated to the Registration Authority (RA) role in a PKI."

EST section 1.1 states that:

"For certificate issuing services, the EST CA is reached through the EST server; the CA could be logically "behind" the EST server or embedded within it."

When the CA is logically "behind" the EST RA, EST does not specify how the RA communicates with the CA. EST section 1 states:

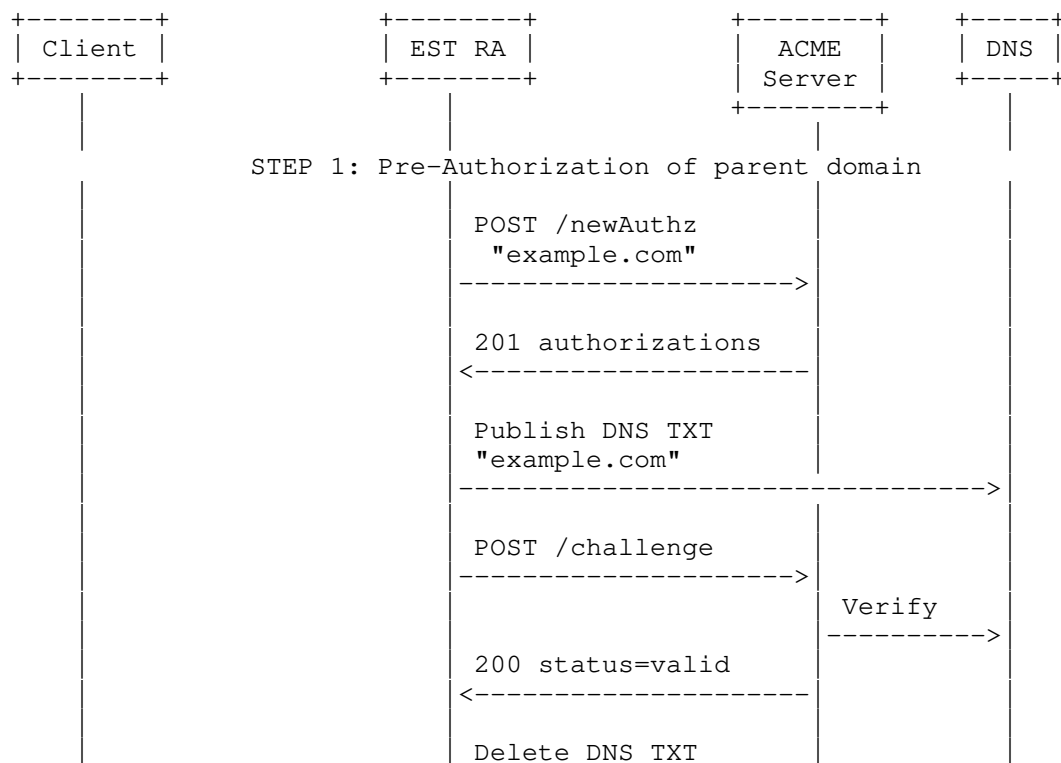
"The nature of communication between an EST server and a CA is not described in this document."

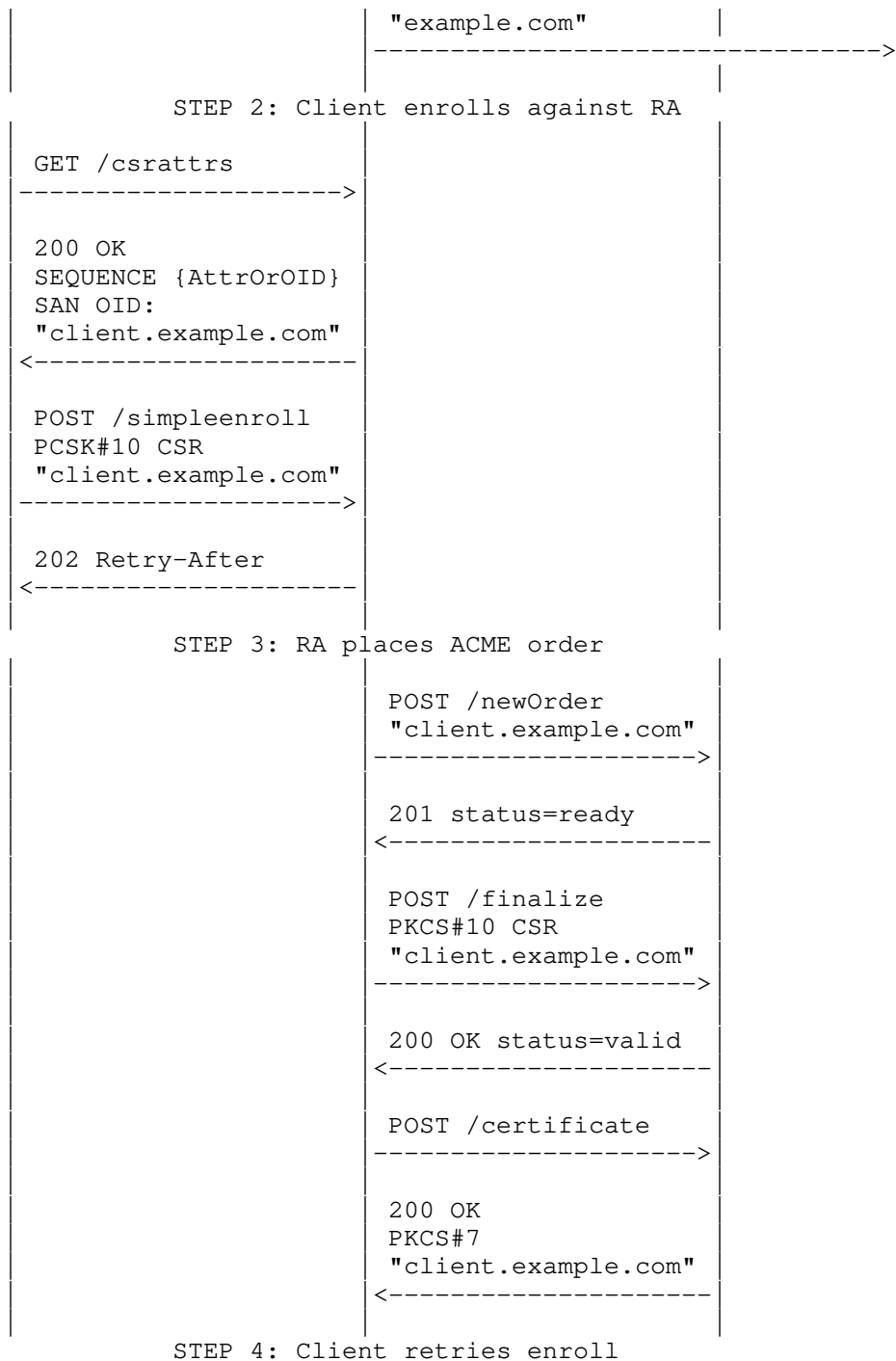


This section outlines how ACME could be used for communication between the EST RA and the CA. The example call flow leverages [I-D.ietf-acme-subdomains] and shows the RA proving ownership of a parent domain, with individual client certificates being subdomains under that parent domain. This is an optimization that reduces DNS and ACME traffic overhead. The RA could of course prove ownership of every explicit client certificate identifier. The example also illustrates using the ACME DNS challenge type, but this integration is not limited to DNS challenges.

The call flow illustrates the client calling the EST /csrattrs API before calling the EST /simpleenroll API. This enables the server to indicate what fields the client should include in the CSR that the client sends in the /simpleenroll API. CSR Attributes handling are discussed in Section 7.2.

The call flow illustrates the EST RA returning a 202 Retry-After response to the client's simpleenroll request. This is an optional step and may be necessary if the interactions between the RA and the ACME server take some time to complete. The exact details of when the RA returns a 202 Retry-After are implementation specific.





<pre> POST /simpleenroll PCSK#10 CSR "client.example.com" -----&gt;  200 OK PKCS#7 "client.example.com" &lt;----- </pre>			
--------------------------------------------------------------------------------------------------------------------------	--	--	--

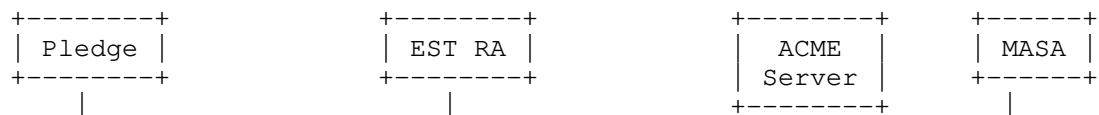
#### 4. ACME Integration with BRSKI

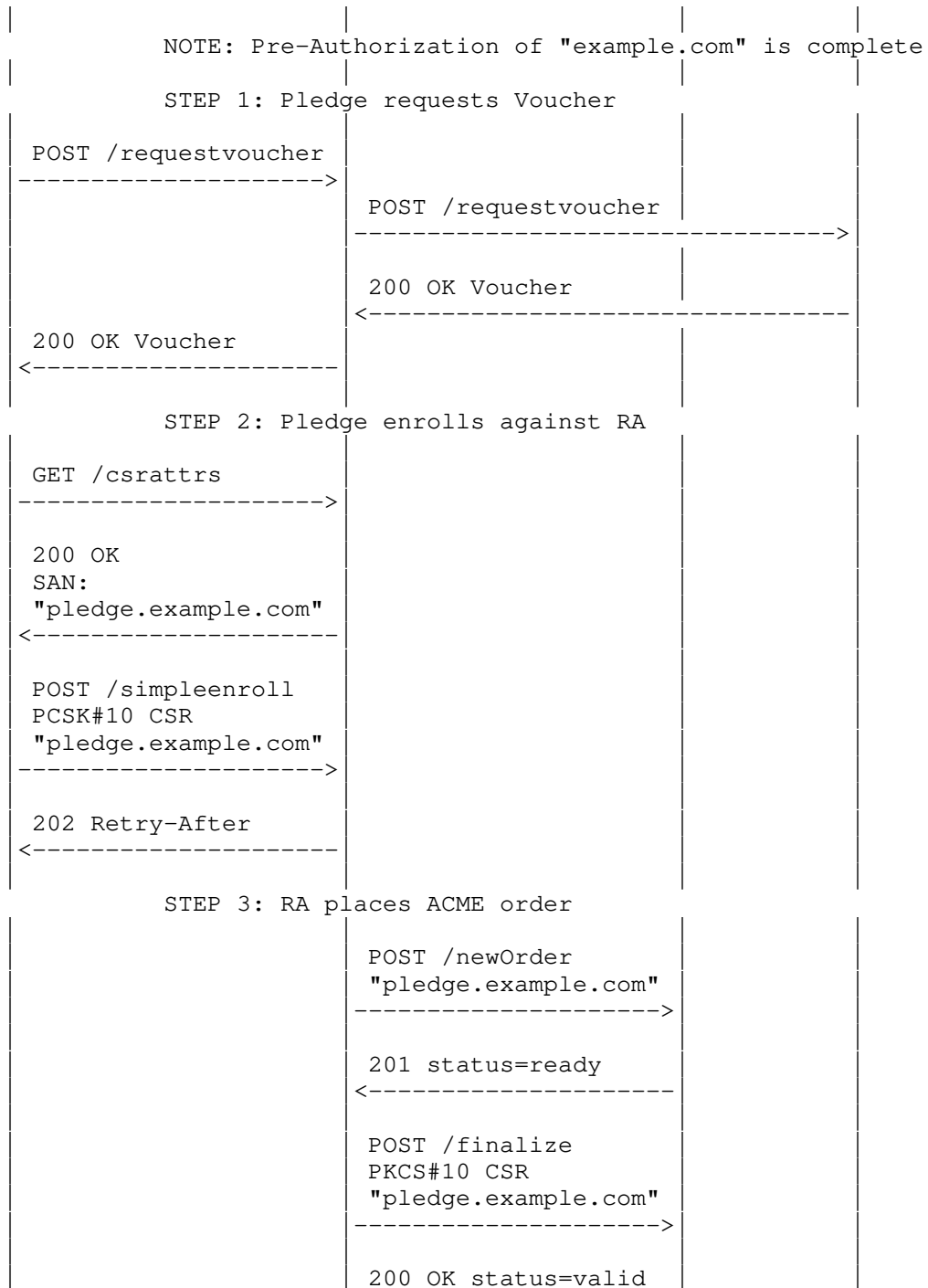
BRSKI [RFC8995] is based upon EST [RFC7030] and defines how to autonomically bootstrap PKI trust anchors into devices via means of signed vouchers. EST certificate enrollment may then optionally take place after trust has been established. BRSKI voucher exchange and trust establishment are based on EST extensions and the certificate enrollment part of BRSKI is fully based on EST. Similar to EST, BRSKI does not define how the EST RA communicates with the CA. Therefore, the mechanisms outlined in the previous section for using ACME as the communications protocol between the EST RA and the CA are equally applicable to BRSKI.

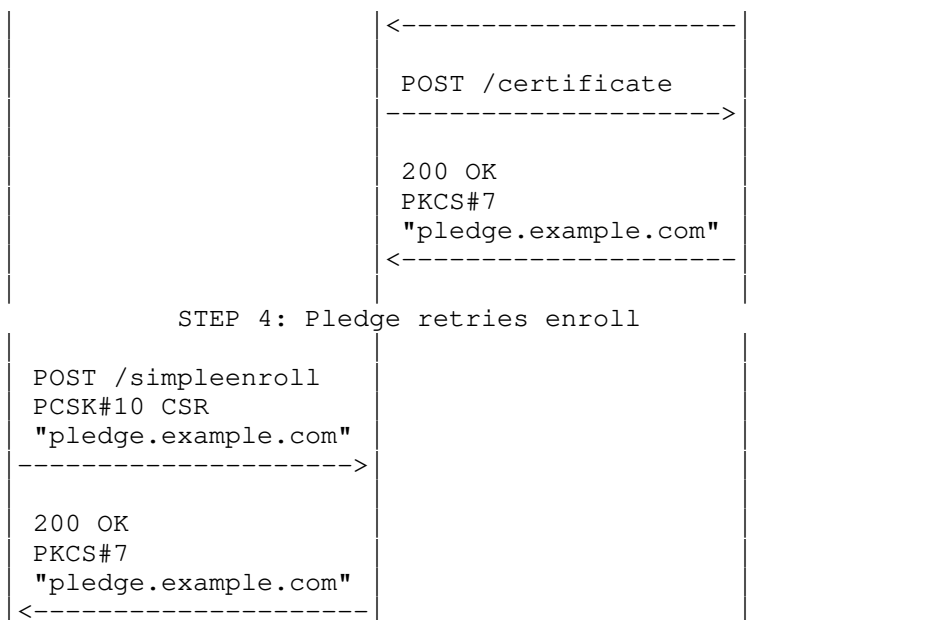
The following call flow shows how ACME may be integrated into a full BRSKI voucher plus EST enrollment workflow. For brevity, it assumes that the EST RA has previously proven ownership of a parent domain and that pledge certificate identifiers are a subdomain of that parent domain. The domain ownership exchanges between the RA, ACME and DNS are not shown. Similarly, not all BRSKI interactions are shown and only the key protocol flows involving voucher exchange and EST enrollment are shown.

Similar to the EST section above, the client calls EST /csrattrs API before calling the EST /simpleenroll API. This enables the server to indicate what fields the pledge should include in the CSR that the client sends in the /simpleenroll API. Refer to section {csr-attributes} for more details.

The call flow illustrates the RA returning a 202 Retry-After response to the initial EST /simpleenroll API. This may be appropriate if processing of the /simpleenroll request and ACME interactions takes some time to complete.





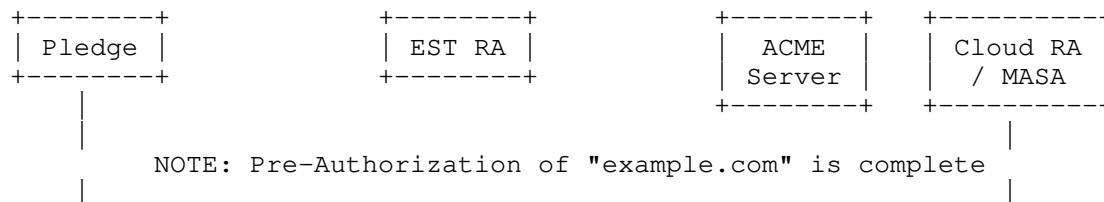


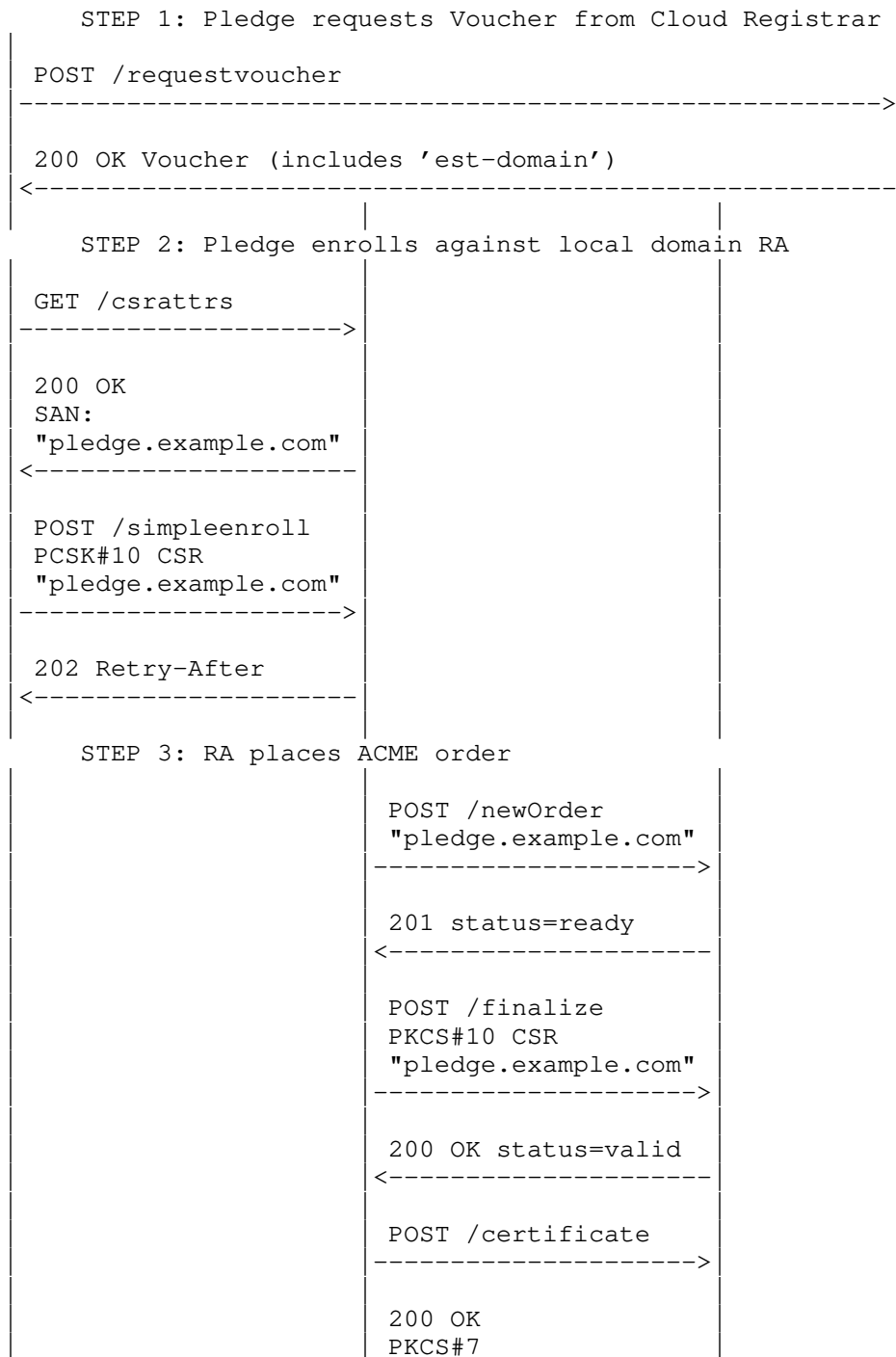
## 5. ACME Integration with BRSKI Default Cloud Registrar

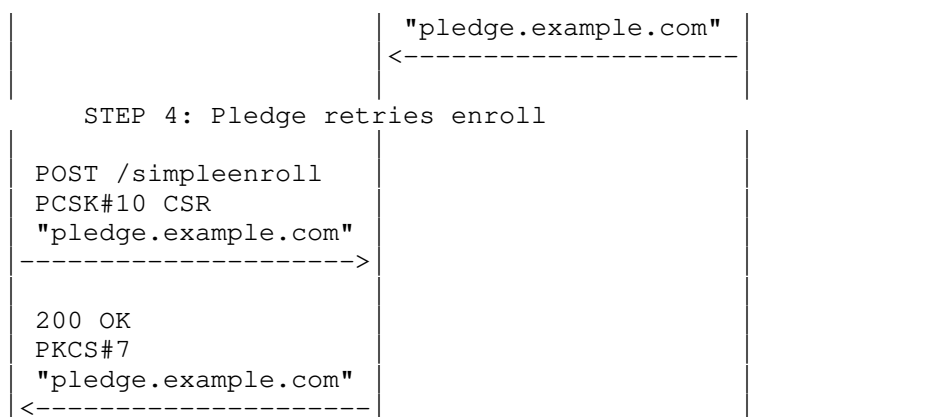
BRSKI Cloud Registrar [I-D.ietf-anima-brski-cloud] specifies the behaviour of a BRSKI Cloud Registrar, and how a pledge can interact with a BRSKI Cloud Registrar when bootstrapping. Similar to the local domain registrar BRSKI flow, ACME can be easily integrated with a cloud registrar bootstrap flow.

BRSKI cloud registrar is flexible and allows for multiple different local domain discovery and redirect scenarios. In the example illustrated here, the extension to [RFC8366] Vouchers which is defined in [I-D.ietf-anima-brski-cloud], and allows the specification of a bootstrap EST domain, is leveraged. This extension allows the cloud registrar to specify the local domain RA that the pledge should connect to for the purposes of EST enrollment.

Similar to the sections above, the client calls EST /csrattrs API before calling the EST /simpleenroll API.







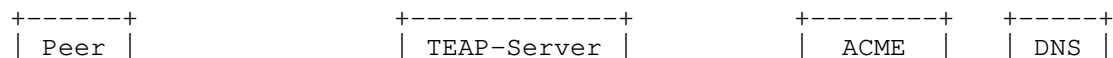
## 6. ACME Integration with TEAP

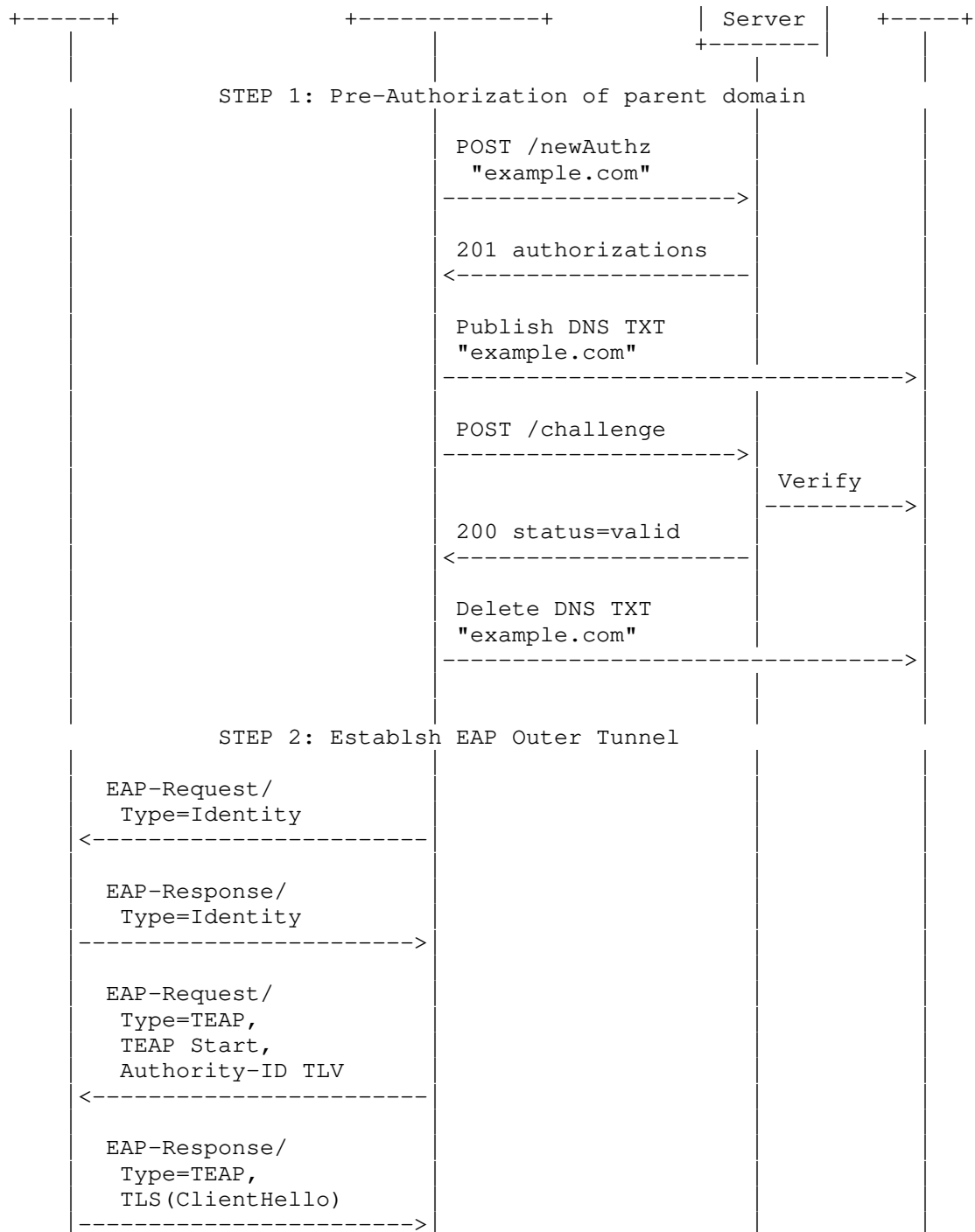
TEAP [RFC7170] defines a tunnel-based EAP method that enables secure communication between a peer and a server by using TLS to establish a mutually authenticated tunnel. TEAP enables certificate provisioning within the tunnel. TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski] defines extensions to TEAP that includes additional Type-Length-Value (TLV) elements for certificate enrollment and BRSKI handling inside the TEAP tunnel. Neither TEAP [RFC7170] or TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski] define how the TEAP server communicates with the CA.

This section outlines how ACME could be used for communication between the TEAP server and the CA. The example call flow leverages [I-D.ietf-acme-subdomains] and shows the TEAP server proving ownership of a parent domain, with individual client certificates being subdomains under that parent domain.

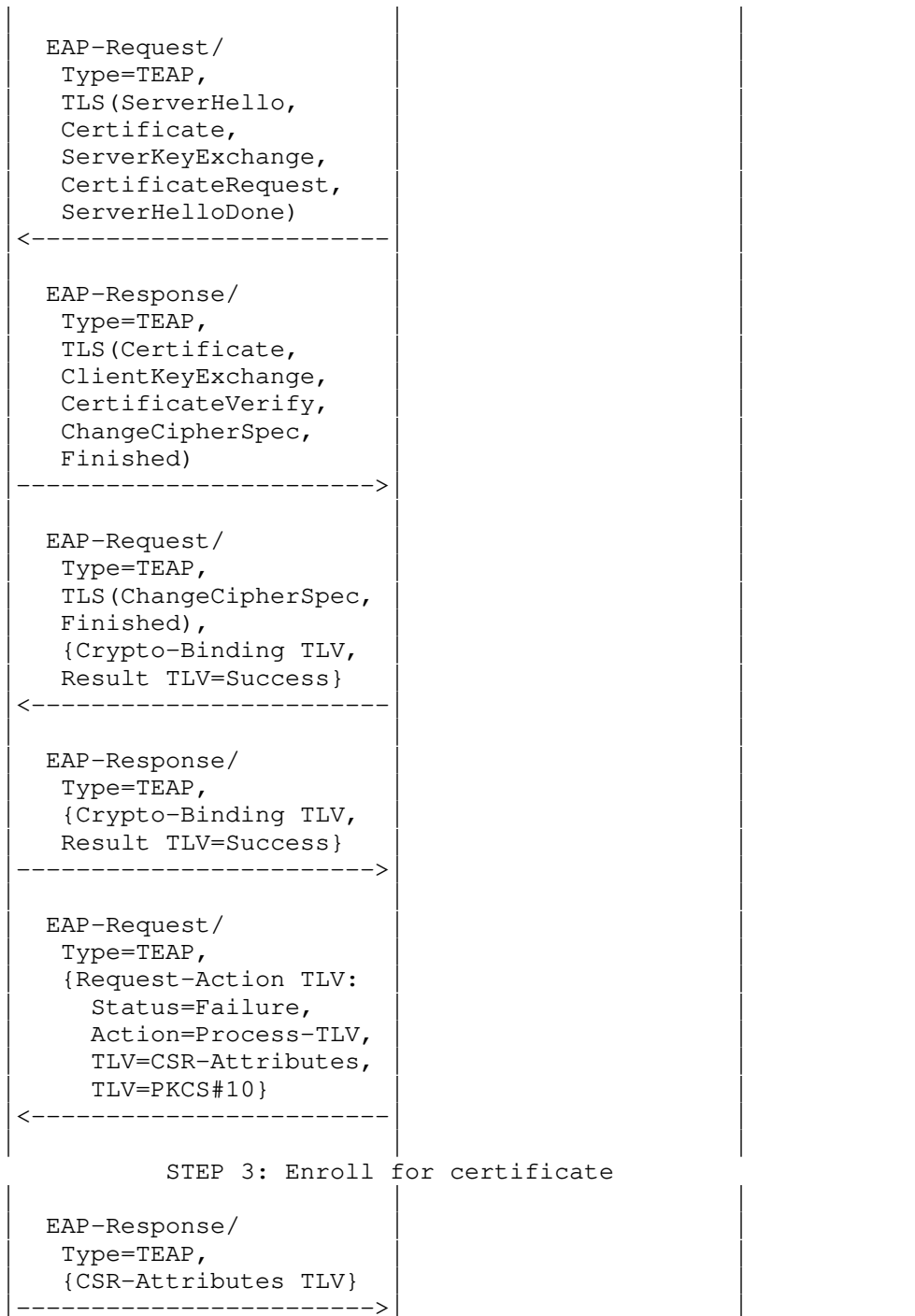
The example illustrates the TEAP server sending a Request-Action TLV including a CSR-Attributes TLV instructing the peer to send a CSR-Attributes TLV to the server. This enables the server to indicate what fields the peer should include in the CSR that the peer sends in the PKCS#10 TLV.

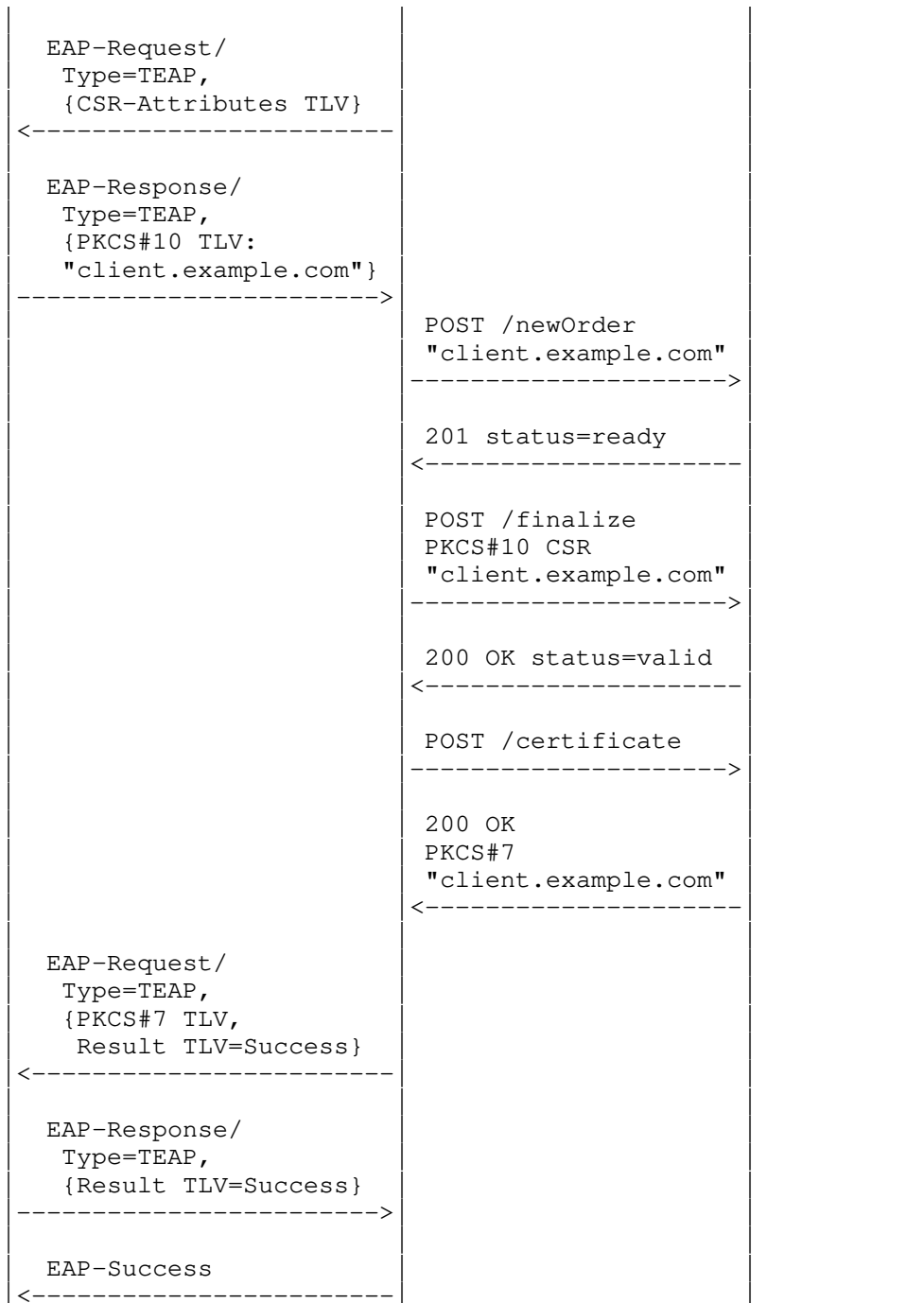
Although not explicitly illustrated in this call flow, the Peer and TEAP Server could exchange BRSKI TLVs, and a BRSKI integration and voucher exchange with a MASA server could take place over TEAP. Whether a BRSKI TLV exchange takes place or not does not impact the ACME specific message exchanges.











## 7. ACME Integration Considerations

### 7.1. Service Operators

The goal of these integrations is enabling issuance of certificates with identifiers in a given domain by an ACME server to a client. It is expected that the EST RA or TEAP servers that the client sends certificate enrollment requests to are operated by the organization that controls the domains. The ACME server is not necessarily operated by the organization that controls the domain.

### 7.2. CSR Attributes

In all integrations, the client MUST send a CSR Attributes request to the EST or TEAP server prior to sending a certificate enrollment request. This enables the server to indicate to the client what attributes, and what attribute values, it expects the client to include in the subsequent CSR request. For example, the server could instruct the peer what Subject Alternative Name entries to include in its CSR.

EST [RFC7030] is not clear on how the CSR Attributes response should be structured, and in particular is not clear on how a server can instruct a client to include specific attribute values in its CSR. [I-D.richardson-lamps-rfc7030-csrattrs] clarifies how a server can use CSR Attributes response to specify specific values for attributes that the client should include in its CSR.

Servers MUST use this mechanism to tell the client what identifiers to include in CSR request. ACME [RFC8555] allows the identifier to be included in either CSR Subject or Subject Alternative Name fields, however [I-D.ietf-uta-use-san] states that Subject Alternative Name field MUST be used. This document aligns with [I-D.ietf-uta-use-san] and Subject Alternate Name field MUST be used. The identifier must be a subdomain of a domain that the server has control over and can fulfill ACME challenges against. The leftmost part of the identifier MAY be a field that the client presented to the server in an IEEE 802.1AR [IDevID].

Servers MAY use this field to instruct the client to include other attributes such as specific policy OIDs. Refer to EST [RFC7030] section 2.6 for further details.

### 7.3. Certificate Chains and Trust Anchors

ACME [RFC8555] section 9.1 states that ACME servers may return a certificate chain to an ACME client where an end entity certificate is followed by certificates that certify it. The trust anchor

certificate MAY be omitted from the chain as it is assumed that the trust anchor is already known by the ACME client i.e. the EST or TEAP server.

#### 7.3.1. EST /cacerts

EST [RFC7030] section 4.2.3 states that the /simpleenroll response contains "only the certificate that was issued". EST [RFC7030] section 4.1.3 states that the /cacerts response "MUST include any additional certificates the client would need to build a chain from an EST CA-issued certificate to the current EST CA TA".

Therefore, the EST server MUST return only the ACME end entity certificate in the /simpleenroll response. The EST server MUST return the remainder of the chain returned by the ACME server to the EST server in the /cacerts response to the client, appending the trust anchor root CA if necessary.

#### 7.3.2. TEAP PKCS#7 TLV

TEAP [RFC7170] section 4.2.16 allows for download of a PKCS#7 certificate chain in response to a TEAP PKCS#10 TLV request. TEAP also allows for download of multiple PKCS#7 certificates in response to a TEAP Trusted-Server-Root TLV request.

The TEAP server MUST return the full ACME client certificate chain in the PKCS#7 response to the PKCS#10 TLV request. The TEAP server MUST return the ACME server trust anchor in a PKCS#7 response to a Trusted-Server-Root TLV request. As outlined in Section 7.4, the TEAP server SHOULD also return the trust anchor that was used for issuing its own identity certificate, if different from the ACME server trust anchor.

#### 7.4. id-kp-cmcRA

BRSKI [RFC8995] mandates that the id-kp-cmcRA extended key usage bit is set in the Registrar (or EST RA) end entity certificate that the Registrar uses when signing voucher request messages sent to the MASA. Public ACME servers may not be willing to issue end entity certificates that have the id-kp-cmcRA extended key usage bit set. In these scenarios, the EST RA may be used by the pledge to get issued certificates by a public ACME server, but the EST RA itself will need an end entity certificate that has been issued by a different CA (e.g. an operator deployed private CA) and that has the id-kp-cmcRA bit set.

## 7.5. Error Handling

ACME [RFC8555] section 6.7 defines multiple errors that may be returned by an ACME server to an ACME client. TEAP [RFC7170] section 4.2.6 defines multiple errors that may be returned by a TEAP server to a client in an Error TLV. EST [RFC7030] section 4.2.3 defines how an EST server may return an error encoded in a CMC response, or may return a human readable error in the response body.

The following mapping from ACME errors to CMC [RFC5272] section 6.1.4 CMCFailInfo and TEAP [RFC7170] section 4.2.6 error codes is RECOMMENDED.

ACME	CMCFailInfo	TEAP Error Code
badCSR	badRequest	1025 Bad CSR
caa	badRequest	1025 Bad CSR
rejectedIdentifier	badIdentity	1024 Bad Identity In CSR
all other errors	internalCAError	1026 Internal CA Error

## 8. IANA Considerations

This document does not make any requests to IANA.

## 9. Security Considerations

This draft is informational and makes no changes to the referenced specifications. All security considerations from these referenced documents are applicable here:

- o EST [RFC7030]
- o BRSKI [RFC8995]
- o BRSKI Default Cloud Registrar [I-D.ietf-anima-brski-cloud]
- o TEAP [RFC7170] and TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski]

Additionally, all Security Considerations in ACME in the following areas are equally applicable to ACME Integrations.

It is expected that the integration mechanisms proposed here will primarily use the DNS-01 challenge documented in [RFC8555] section 8.4. The security considerations in RFC8555 says:

The DNS is a common point of vulnerability for all of these challenges. An entity that can provision false DNS records for a domain can attack the DNS challenge directly and can provision false A/AAAA records to direct the ACME server to send its HTTP validation query to a remote server of the attacker's choosing.

It is expected that the TEAP-EAP server/EST Registrar will perform DNS dynamic updates to a DNS primary server using [RFC3007] Dynamic updates, secured with either SIG(0), or TSIG keys.

A major source of vulnerability is the disclosure of these DNS key records. An attacker that has access to them, can provision their own certificates into the the name space of the entity.

For many uses, this may allow the attacker to get access to some enterprise resource. When used to provision, for instance, a (SIP) phone system this would permit an attacker to impersonate a legitimate phone. Not only does this allow for redirection of phone calls, but possibly also toll fraud.

Operators should consider restricting the integration server such that it can only update the DNS records for a specific zone or zones where ACME is required for client certificate enrollment automation. For example, if all IoT devices in an organisation enroll using EST against an EST RA, and all IoT devices will be issued certificates in a subdomain under `iot.example.com`, then the integration server could be issued a credential that only allows updating of DNS records in a zone that includes domains in the `iot.example.com` namespace, but does not allow updating of DNS records under any other `example.com` DNS namespace.

When performing challenge fulfilment via writing files to HTTP web servers, write access should only be granted to a specific set of servers, and only to a specific set of directories for storage of challenge files.

#### 9.1. Denial of Service against ACME infrastructure

The intermediate node (the TEAP-EAP server, or the EST Registrar) should cache the resulting certificates such that if the communication with the pledge is lost, subsequent attempts to enroll will result in the cache certificate being returned.

As many ACME servers have per-day, per-IP and per-subjectAltName limits, it is prudent not to request identical certificates too often. This could be due to operator or installer error, with multiple configuration resets occurring within a short period of time.

The cache should be indexed by the complete contents of the Certificate Signing Request, and should not persist beyond the notAfter date in the certificate.

This means that if the private/public keypair changes on the pledge, then a new certificate will be issued. If the requested SubjectAltName changes, then a new certificate will be requested.

In a case where a device is simply factory reset, and enrolls again, then the same certificate can be returned.

## 10. Informative References

- [CAB] CA/Browser Forum, "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates", n.d., <<https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.7.1.pdf>>.
- [I-D.ietf-acme-subdomains] Friel, O., Barnes, R., Hollebeek, T., and M. Richardson, "ACME for Subdomains", draft-ietf-acme-subdomains-00 (work in progress), October 2021.
- [I-D.ietf-anima-brski-cloud] Friel, O., Shekh-Yusef, R., and M. Richardson, "BRSKI Cloud Registrar", draft-ietf-anima-brski-cloud-02 (work in progress), October 2021.
- [I-D.ietf-uta-use-san] Salz, R., "Update to Verifying TLS Server Identities with X.509 Certificates", draft-ietf-uta-use-san-00 (work in progress), April 2021.
- [I-D.lear-eap-teap-brski] Lear, E., Friel, O., Cam-Winget, N., and D. Harkins, "TEAP Update and Extensions for Bootstrapping", draft-lear-eap-teap-brski-06 (work in progress), August 2021.
- [I-D.richardson-lamps-rfc7030-csrattrs] Richardson, M., Harkins, D., Oheimb, D. D. V., and O. Friel, "Clarification of RFC7030 CSR Attributes definition", draft-richardson-lamps-rfc7030-csrattrs-01 (work in progress), October 2021.
- [IDevID] IEEE, "IEEE Standard for Local and metropolitan area networks - Secure Device Identity", n.d., <<https://1.ieee802.org/security/802-lar>>.

- [RFC0819] Su, Z. and J. Postel, "The Domain Naming Convention for Internet User Applications", RFC 819, DOI 10.17487/RFC0819, August 1982, <<https://www.rfc-editor.org/info/rfc819>>.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [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>>.
- [RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, DOI 10.17487/RFC3007, November 2000, <<https://www.rfc-editor.org/info/rfc3007>>.
- [RFC5272] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC)", RFC 5272, DOI 10.17487/RFC5272, June 2008, <<https://www.rfc-editor.org/info/rfc5272>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.
- [RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <<https://www.rfc-editor.org/info/rfc7030>>.
- [RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <<https://www.rfc-editor.org/info/rfc7170>>.
- [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>>.
- [RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <<https://www.rfc-editor.org/info/rfc8366>>.
- [RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <<https://www.rfc-editor.org/info/rfc8499>>.



- [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>>.
- [RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <<https://www.rfc-editor.org/info/rfc8995>>.

## Authors' Addresses

Owen Friel  
Cisco

Email: [ofriel@cisco.com](mailto:ofriel@cisco.com)

Richard Barnes  
Cisco

Email: [rlb@ipv.sx](mailto:rlb@ipv.sx)

Rifaat Shekh-Yusef  
Auth0

Email: [rifaat.s.ietf@gmail.com](mailto:rifaat.s.ietf@gmail.com)

Michael Richardson  
Sandelman Software Works

Email: [mcr+ietf@sandelman.ca](mailto:mcr+ietf@sandelman.ca)

ACME Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 3 September 2022

O. Friel  
R. Barnes  
Cisco  
T. Hollebeek  
DigiCert  
M. Richardson  
Sandelman Software Works  
2 March 2022

ACME for Subdomains  
draft-ietf-acme-subdomains-02

Abstract

This document outlines how ACME can be used by a client to obtain a certificate for a subdomain identifier from a certification authority. The client has fulfilled a challenge against a parent domain but does not need to fulfill a challenge against the explicit subdomain as certification authority policy allows issuance of the subdomain certificate without explicit subdomain ownership proof.

Status of This Memo

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

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

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

This Internet-Draft will expire on 3 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights

and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	2
3. ACME Workflow and Identifier Requirements . . . . .	3
4. ACME Issuance of Subdomain Certificates . . . . .	5
4.1. ACME Challenge Type . . . . .	5
4.2. Authorization Object . . . . .	5
4.3. Pre-Authorization . . . . .	6
4.4. New Orders . . . . .	7
4.5. Directory Object Metadata . . . . .	9
5. Illustrative Call Flow . . . . .	9
6. IANA Considerations . . . . .	15
6.1. Authorization Object Fields Registry . . . . .	15
6.2. Directory Object Metadata Fields Registry . . . . .	15
7. Security Considerations . . . . .	16
7.1. ACME Server Policy Considerations . . . . .	17
8. References . . . . .	17
8.1. Normative References . . . . .	17
8.2. Informative References . . . . .	17
Authors' Addresses . . . . .	18

## 1. Introduction

ACME [RFC8555] defines a protocol that a certification authority (CA) and an applicant can use to automate the process of domain name ownership validation and X.509v3 (PKIX) [RFC5280] certificate issuance. This document outlines how ACME can be used to issue subdomain certificates, without requiring the ACME client to explicitly fulfill an ownership challenge against the subdomain identifiers - the ACME client need only fulfill an ownership challenge against a parent domain identifier.

## 2. Terminology

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.

The following terms are defined in DNS Terminology [RFC8499] and are reproduced here:

- \* **Label:** An ordered list of zero or more octets that makes up a portion of a domain name. Using graph theory, a label identifies one node in a portion of the graph of all possible domain names.
- \* **Domain Name:** An ordered list of one or more labels.
- \* **Subdomain:** "A domain is a subdomain of another domain if it is contained within that domain. This relationship can be tested by seeing if the subdomain's name ends with the containing domain's name." (Quoted from [RFC1034], Section 3.1) For example, in the host name "nnn.mmm.example.com", both "mmm.example.com" and "nnn.mmm.example.com" are subdomains of "example.com". Note that the comparisons here are done on whole labels; that is, "ooo.example.com" is not a subdomain of "oo.example.com".
- \* **Fully-Qualified Domain Name (FQDN):** This is often just a clear way of saying the same thing as "domain name of a node", as outlined above. However, the term is ambiguous. Strictly speaking, a fully-qualified domain name would include every label, including the zero-length label of the root: such a name would be written "www.example.net." (note the terminating dot). But, because every name eventually shares the common root, names are often written relative to the root (such as "www.example.net") and are still called "fully qualified". This term first appeared in [RFC0819]. In this document, names are often written relative to the root.

The following additional terms are used in this document:

- \* **Certification Authority (CA):** An organization that is responsible for the creation, issuance, revocation, and management of Certificates. The term applies equally to both Roots CAs and Subordinate CAs
- \* **CSR:** Certificate Signing Request
- \* **Parent Domain:** a domain is a parent domain of a subdomain if it contains that subdomain, as per the [RFC8499] definition of subdomain. For example, for the host name "nnn.mmm.example.com", both "mmm.example.com" and "example.com" are parent domains of "nnn.mmm.example.com".

### 3. ACME Workflow and Identifier Requirements

A typical ACME workflow for issuance of certificates is as follows:

1. client POSTs a newOrder request that contains a set of "identifiers"

2. server replies with a set of "authorizations" and a "finalize" URI
3. client sends POST-as-GET requests to retrieve the "authorizations", with the downloaded "authorization" object(s) containing the "identifier" that the client must prove that they control, and a set of associated "challenges", one of which the client must fulfil
4. client proves control over the "identifier" in the "authorization" object by completing one of the specified challenges, for example, by publishing a DNS TXT record
5. client POSTs a CSR to the "finalize" API
6. server replies with an updated order object that includes a "certificate" URI
7. client sends POST-as-GET request to the "certificate" URI to download the certificate

ACME places the following restrictions on "identifiers":

- \* [RFC8555] section 7.1.3: The authorizations required are dictated by server policy; there may not be a 1:1 relationship between the order identifiers and the authorizations required.
- \* [RFC8555] section 7.1.4: the only type of "identifier" defined by the ACME specification is an FQDN: "The only type of identifier defined by this specification is a fully qualified domain name (type: "dns"). The domain name MUST be encoded in the form in which it would appear in a certificate."
- \* [RFC8555] section 7.4: the "identifier" in the CSR request must match the "identifier" in the newOrder request: "The CSR MUST indicate the exact same set of requested identifiers as the initial newOrder request."
- \* [RFC8555] section 8.3: the "identifier", or FQDN, in the "authorization" object must be used when fulfilling challenges via HTTP: "Construct a URL by populating the URL template ... where the domain field is set to the domain name being verified"
- \* [RFC8555] section 8.4: the "identifier", or FQDN, in the "authorization" object must be used when fulfilling challenges via DNS: "The client constructs the validation domain name by prepending the label "\_acme-challenge" to the domain name being validated."

ACME does not mandate that the "identifier" in a newOrder request matches the "identifier" in "authorization" objects.

#### 4. ACME Issuance of Subdomain Certificates

As noted in the previous section, ACME does not mandate that the "identifier" in a newOrder request matches the "identifier" in "authorization" objects. This means that the ACME specification does not preclude an ACME server processing newOrder requests and issuing certificates for a subdomain without requiring a challenge to be fulfilled against that explicit subdomain.

ACME server policy could allow issuance of certificates for a subdomain to a client where the client only has to fulfill an authorization challenge for a parent domain of that subdomain. This allows a flow where a client proves ownership of, for example, "example.org" and then successfully obtains a certificate for "sub.example.org".

ACME server policy is out of scope of this document, however some commentary is provided in Section 7.1.

Clients need a mechanism to instruct the ACME server that they are requesting authorization for all subdomains subordinate to the specified domain, as opposed to just requesting authorization for an explicit domain identifier. Clients need a mechanism to do this in both newAuthz and newOrder requests. ACME servers need a mechanism to indicate to clients that authorization objects are valid for all subdomains under the specified domain. These are described in this section.

##### 4.1. ACME Challenge Type

ACME for subdomains is restricted for use with "dns-01" challenges. If a server policy allows a client to fulfill a challenge against a parent domain of a requested certificate FQDN identifier, then the server MUST issue a "dns-01" challenge against that parent domain.

##### 4.2. Authorization Object

ACME [RFC8555] section 7.1.4 defines the authorization object. When ACME server policy allows authorization for subdomains subordinate to an domain, the server indicates this by including the "subdomains" flag in the authorization object for that domain identifier:

subdomains (optional, boolean): This field MUST be present and true for authorizations where ACME server policy allows certificates to be issued for any subdomain subordinate to the domain specified in the 'identifier' field of the authorization object.

The following example shows an authorization object for the domain example.org where the authorization covers the subdomains subordinate to example.org.

```
{
  "status": "valid",
  "expires": "2015-03-01T14:09:07.99Z",

  "identifier": {
    "type": "dns",
    "value": "example.org"
  },

  "challenges": [
    {
      "url": "https://example.com/acme/chall/prV_B7yEyA4",
      "type": "http-01",
      "status": "valid",
      "token": "DGyRejmCefe7v4NfDGDkFA",
      "validated": "2014-12-01T12:05:58.16Z"
    }
  ],

  "subdomains": true
}
```

If the "subdomains" field is not included, then the assumed default value is false.

#### 4.3. Pre-Authorization

The standard ACME workflow has authorization objects created reactively in response to a certificate order. ACME also allows for pre-authorization, where clients obtain authorization for an identifier proactively, outside of the context of a specific issuance. With the ACME pre-authorization flow, a client can pre-authorize for a domain once, and then issue multiple newOrder requests for certificates with identifiers in the subdomains subordinate to that domain.

ACME [RFC8555] section 7.4.1 defines the "identifier" object for newAuthz requests. One additional field for the "identifier" object is defined:

subdomains (optional, boolean): An ACME client sets this flag to indicate to the server that it is requesting an authorization for the subdomains subordinate to the specified domain identifier value

Clients include the flag in the "identifier" object of newAuthz requests to indicate that they are requesting a subdomain authorization. In the following example newAuthz payload, the client is requesting pre-authorization for the subdomains subordinate to example.org.

```
"payload": base64url({
  "identifier": {
    "type": "dns",
    "value": "example.org",
    "subdomains": true
  }
})
```

If the server is willing to allow a single authorization for the subdomains, and there is not an existing authorization object for the identifier, then it will create an authorization object and include the "subdomains" flag with value of true. If the server policy does not allow creation of subdomain authorizations subordinate to that domain, the server can create an authorization object for the indicated identifier, and include the "subdomains" flag with value of false. In both scenarios, handling of the pre-authorization follows the process documented in ACME section 7.4.1.

#### 4.4. New Orders

Clients need a mechanism to optionally indicate to servers whether or not they are authorized to fulfill challenges against parent domains for a given identifier FQDN. For example, if a client places an order for an identifier foo.bar.example.org, and is authorized to update DNS TXT records against the parent domains bar.example.org or example.org, then the client needs a mechanism to indicate control over the parent domains to the ACME server.

This can be achieved by adding an optional field "parentDomain" to the "identifiers" field in the order object:



parentDomain (optional, string): This is a parent domain of the requested identifier. The client MUST have DNS control over the parent domain.

This field specifies a parent domain of the identifier that the client has DNS control over, and is capable of fulfilling challenges against. Based on server policy, the server can choose to issue a challenge against any parent domain of the identifier up to and including the specified "parentDomain", and create a corresponding authorization object against the chosen identifier.

In the following example newOrder payload, the client requests a certificate for identifier foo.bar.example.org and indicates that it can fulfill a challenge against the parent domain bar.example.org. The server can then choose to issue a challenge against either foo.bar.example.org or bar.example.org identifiers.

```
"payload": base64url({
  "identifiers": [
    { "type": "dns",
      "value": "foo.bar.example.org",
      "parentDomain": "bar.example.org" }
  ],
  "notBefore": "2016-01-01T00:04:00+04:00",
  "notAfter": "2016-01-08T00:04:00+04:00"
})
```

In the following example newOrder payload, the client requests a certificate for identifier foo.bar.example.org and indicates that it can fulfill a challenge against the parent domain example.org. The server can then choose to issue a challenge against any one of foo.bar.example.org, bar.example.org or example.org identifiers.

```
"payload": base64url({
  "identifiers": [
    { "type": "dns",
      "value": "foo.bar.example.org",
      "parentDomain": "example.org" }
  ],
  "notBefore": "2016-01-01T00:04:00+04:00",
  "notAfter": "2016-01-08T00:04:00+04:00"
})
```

If the client is unable to fulfill authorizations against parent domain, the client should not include the "parentDomain" field.

Server newOrder handling generally follows the process documented ACME section 7.4. If the server is willing to allow subdomain authorizations for the domain specified in "parentDomain", then it creates an authorization object against that parent domain and includes the "subdomains" flag with a value of true. If the server policy does not allow creation of subdomain authorizations against that parent domain, then it can create an authorization object for the indicated identifier value, and includes the "subdomains" flag with value of false.

#### 4.5. Directory Object Metadata

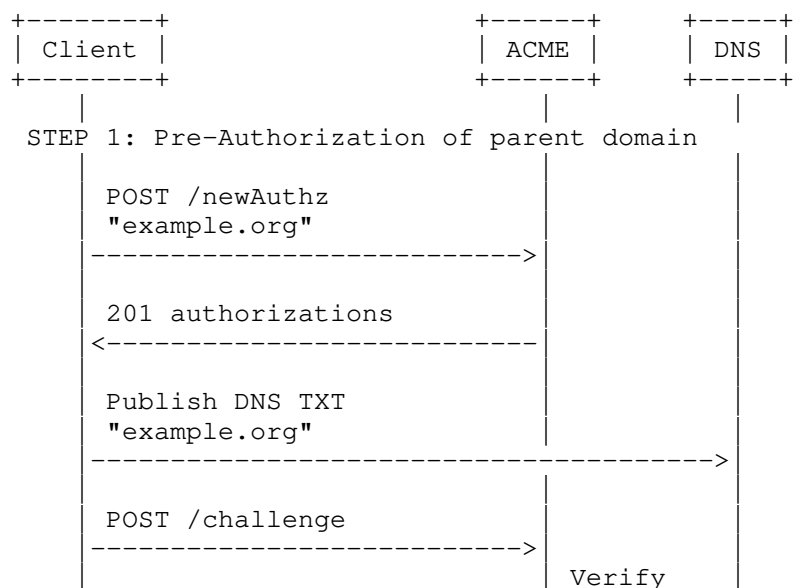
An ACME server can advertise support for authorization of subdomains by including the following boolean flag in its "ACME Directory Metadata Fields" registry:

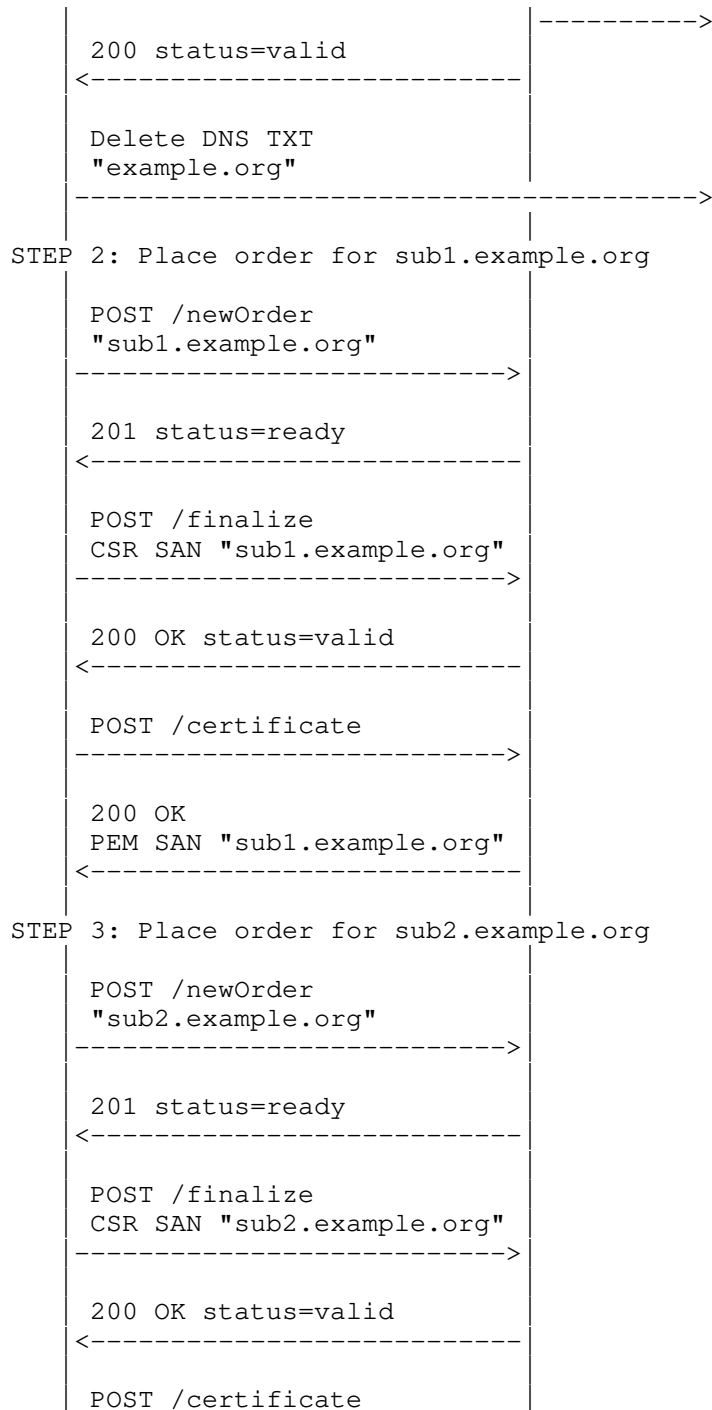
subdomains (optional, bool): Indicates if an ACME server supports authorization of subdomains.

If not specified, then no default value is assumed. If an ACME server supports authorization of subdomains, it can indicate this by including this field with a value of "true".

#### 5. Illustrative Call Flow

The call flow illustrated here uses the ACME pre-authorization flow using DNS-based proof of ownership.





```

----->
200 OK
PEM SAN "sub2.example.org"
<-----

```

\* STEP 1: Pre-authorization of parent domain

The client sends a newAuthz request for the parent domain including the "subdomains" flag in the identifier object.

```

POST /acme/new-authz HTTP/1.1
Host: example.com
Content-Type: application/jose+json

```

```

{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/evOfKhNU60wg",
    "nonce": "uQpSjlRb4vQVCjVYAyyUWg",
    "url": "https://example.com/acme/new-authz"
  }),
  "payload": base64url({
    "identifier": {
      "type": "dns",
      "value": "example.org",
      "subdomains": true
    }
  }),
  "signature": "nuSDISbWG8mMgE7H...QyVUL68yzf3Zawps"
}

```

The server creates and returns an authorization object for the identifier including the "subdomains" flag. The object is initially in "pending" state.

```
{
  "status": "pending",
  "expires": "2015-03-01T14:09:07.99Z",

  "identifier": {
    "type": "dns",
    "value": "example.org"
  },

  "challenges": [
    {
      "url": "https://example.com/acme/chall/prV_B7yEyA4",
      "type": "http-01",
      "status": "pending",
      "token": "DGyRejmCefe7v4NfDGDkFA",
      "validated": "2014-12-01T12:05:58.16Z"
    }
  ],

  "subdomains": true
}
```

Once the client completes the challenge, the server will transition the authorization object and associated challenge object status to "valid". The flow above illustrates the ACME server replying to the client's challenge with status of "valid" after the ACME server has validated the DNS challenge. However, the validation flow may take some time, so the client may need to poll the authorization resource to see when it is finalized.

\* STEP 2: The client places a newOrder for `subl.example.org`

The client sends a newOrder request to the server and includes the subdomain identifier. Note that the identifier is a subdomain of the parent domain that has been pre-authorized in step 1. The client does not need to include the "subdomains" field in the "identifier" object as it has already pre-authorized the parent domain.

```
POST /acme/new-order HTTP/1.1
Host: example.com
Content-Type: application/jose+json

{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/evOfKhNU60wg",
    "nonce": "5XJ1L3lEkMG7tR6pA00clA",
    "url": "https://example.com/acme/new-order"
  }),
  "payload": base64url({
    "identifiers": [
      { "type": "dns", "value": "sub1.example.org" }
    ],
    "notBefore": "2016-01-01T00:04:00+04:00",
    "notAfter": "2016-01-08T00:04:00+04:00"
  }),
  "signature": "H6ZXtGjTZyUnPeKn...wEA4Tk1Bdh3e454g"
}
```

As an authorization object already exists for the parent domain, the server replies with an order object with a status of "ready" that includes a link to the existing "valid" authorization object.

```
HTTP/1.1 201 Created
Replay-Nonce: MYAuvOpaoIiywTezizk5vw
Link: <https://example.com/acme/directory>;rel="index"
Location: https://example.com/acme/order/TolocE8rfgo

{
  "status": "ready",
  "expires": "2016-01-05T14:09:07.99Z",

  "notBefore": "2016-01-01T00:00:00Z",
  "notAfter": "2016-01-08T00:00:00Z",

  "identifiers": [
    { "type": "dns", "value": "sub1.example.org" }
  ],

  "authorizations": [
    "https://example.com/acme/authz/PAniVnsZcis"
  ],

  "finalize": "https://example.com/acme/order/Tolocrfgo/finalize"
}
```

The client can proceed to finalize the order and download the certificate for `sub1.example.org`.

\* STEP 3: The client places a `newOrder` for `sub2.example.org`

The client sends a `newOrder` request to the server and includes the subdomain identifier. Note that the identifier is a subdomain of the parent domain that has been pre-authorized in step 1. The client does not need to include the `"subdomains"` field in the `"identifier"` object as it has already pre-authorized the parent domain.

```
POST /acme/new-order HTTP/1.1
Host: example.com
Content-Type: application/jose+json
```

```
{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/evOfKhNU60wg",
    "nonce": "5XJ1L3lEkMG7tR6pA00clA",
    "url": "https://example.com/acme/new-order"
  }),
  "payload": base64url({
    "identifiers": [
      { "type": "dns", "value": "sub2.example.org" }
    ],
    "notBefore": "2016-01-01T00:04:00+04:00",
    "notAfter": "2016-01-08T00:04:00+04:00"
  }),
  "signature": "H6ZXtGjTZyUnPeKn...wEA4Tk1Bdh3e454g"
}
```

As an authorization object already exists for the parent domain, the server replies with an order object with a status of `"ready"` that includes a link to the existing `"valid"` authorization object.

```

HTTP/1.1 201 Created
Replay-Nonce: MYAuvOpaoIiywTezizk5vw
Link: <https://example.com/acme/directory>;rel="index"
Location: https://example.com/acme/order/T0locE8rfgo

{
  "status": "ready",
  "expires": "2016-01-05T14:09:07.99Z",

  "notBefore": "2016-01-01T00:00:00Z",
  "notAfter": "2016-01-08T00:00:00Z",

  "identifiers": [
    { "type": "dns", "value": "sub1.example.org" }
  ],

  "authorizations": [
    "https://example.com/acme/authz/PAniVnsZcis"
  ],

  "finalize": "https://example.com/acme/order/R0ni7rdde/finalize"
}

```

The client can proceed to finalize the order and download the certificate for sub2.example.org.

## 6. IANA Considerations

### 6.1. Authorization Object Fields Registry

The following field is added to the "ACME Authorization Object Fields" registry defined in ACME [RFC8555].

Field Name	Field Type	Configurable	Reference
subdomains	boolean	false	RFC XXXX

### 6.2. Directory Object Metadata Fields Registry

The following field is added to the "ACME Directory Metadata Fields" registry defined in ACME [RFC8555].



Field Name	Field Type	Reference
subdomains	boolean	RFC XXXX

## 7. Security Considerations

This document documents enhancements to ACME [RFC8555] that optimize the protocol flows for issuance of certificates for subdomains. The underlying goal of ACME for Subdomains remains the same as that of ACME: managing certificates that attest to identifier/key bindings for these subdomains. Thus, ACME for Subdomains has the same two security goals as ACME:

1. Only an entity that controls an identifier can get an authorization for that identifier
2. Once authorized, an account key's authorizations cannot be improperly used by another account

ACME for Subdomains makes no changes to:

- \* account or account key management
- \* ACME channel establishment, security mechanisms or threat model
- \* Validation channel establishment, security mechanisms or threat model

Therefore, all Security Considerations in ACME in the following areas are equally applicable to ACME for Subdomains:

- \* Threat Model
- \* Integrity of Authorizations
- \* Denial-of-Service Considerations
- \* Server-Side Request Forgery
- \* CA Policy Considerations

Some additional comments on ACME server policy are given in the following section.

### 7.1. ACME Server Policy Considerations

The ACME for Subdomains and the ACME specifications do not mandate any specific ACME server or CA policies, or any specific use cases for issuance of certificates. For example, an ACME server could be used:

- \* to issue Web PKI certificates where the ACME server must comply with CA/Browser Forum [CAB] Baseline Requirements.
- \* as a Private CA for issuance of certificates within an organisation. The organisation could enforce whatever policies they desire on the ACME server.
- \* for issuance of IoT device certificates. There are currently no IoT device certificate policies that are generally enforced across the industry. Organizations issuing IoT device certificates can enforce whatever policies they desire on the ACME server.

ACME server policy could specify whether:

- \* issuance of subdomain certificates is allowed based on proof of ownership of a parent domain
- \* issuance of subdomain certificates is allowed, but only for a specific set of parent domains
- \* whether DNS based proof of ownership, or HTTP based proof of ownership, or both, are allowed

ACME server policy specification is explicitly out of scope of this document.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [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/rfc/rfc8174>>.

### 8.2. Informative References

- [CAB] CA/Browser Forum, "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates", n.d., <<https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.7.1.pdf>>.
- [RFC0819] Su, Z. and J. Postel, "The Domain Naming Convention for Internet User Applications", RFC 819, DOI 10.17487/RFC0819, August 1982, <<https://www.rfc-editor.org/rfc/rfc819>>.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/rfc/rfc1034>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/rfc/rfc5280>>.
- [RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <<https://www.rfc-editor.org/rfc/rfc8499>>.
- [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/rfc/rfc8555>>.

## Authors' Addresses

Owen Friel  
Cisco  
Email: [ofriel@cisco.com](mailto:ofriel@cisco.com)

Richard Barnes  
Cisco  
Email: [rlb@ipv.sx](mailto:rlb@ipv.sx)

Tim Hollebeek  
DigiCert  
Email: [tim.hollebeek@digicert.com](mailto:tim.hollebeek@digicert.com)

Michael Richardson  
Sandelman Software Works

Email: [mcr+ietf@sandelman.ca](mailto:mcr+ietf@sandelman.ca)