Workgroup:
Automated Certificate Management Environment
Internet-Draft: draft-ietf-acme-dtnnodeid-01
Published: 7 March 2021
Intended Status: Experimental
Expires: 8 September 2021
Authors: B. Sipos
RKF Engineering

# Automated Certificate Management Environment (ACME) Delay-Tolerant Networking (DTN) Node ID Validation Extension

## 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 Uniform Resource Identifier (URI) and ACME Identifier type "uri".

## 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 8 September 2021.

**Table of Contents**



## 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 Uniform Resource Identifier (URI) used to represent the Node ID of a Delay-Tolerant Networking (DTN) Node. A DTN Node ID is a URI with a specific set of allowed schemes, and determines how bundles are routed within a DTN. Currently the schemes "dtn" and "ipn" as defined in [I-D.ietf-dtn-bpbis] are valid for a Node ID.

Once an ACME server validates a Node ID, either as a pre-authorization of the "uri" or as one of the authorizations of an order containing a "uri", the client can finalize the order using an associated certificate signing request. 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 4.1. Once a certificate is issued for a Node ID, how the ACME client configures the 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 [I-D.ietf-acme-email-smime]. For that reason some token splitting terminology in this document is taken from the email specification.

### 1.1. 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 a ACME client requests a pre-authorization or an order with a "uri" which could be used as a DTN Node ID, the ACME server offers a 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 signature, and sends an ACME Node ID Validation Response Bundle with the signature. Finally, the ACME server receives the Response Bundle and checks that the signature came from the client account key associated with the original request.

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

### 1.2. 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.

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 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.

### 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
```

## 2. URI ACME Identifier

This specification is the first to make use of a URI to identify a service for a certificate request in ACME. The URI-type identifier is general purpose, and validating ownership of a URI requires a specific purpose related to its "scheme" component. In this document, the only purpose for which a URI ACME identifier is validated is as a DTN Node ID (see Section 3), but other specifications can define challenge types for other URI uses.

Identifiers of type "uri" MUST appear in an extensionRequest attribute [RFC2985] requesting a subjectAltName extension of type uniformResourceIdentifier having a value consistent with the requirements of [RFC3986].

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

If an ACME server wishes to request proof that a user controls a URI, it SHALL create an authorization with the identifier type "uri". The value field of the identifier SHALL contain the textual form of the URI as defined in Section 3 of [RFC3986]. The ACME server SHALL NOT decode or attempt to dereference the URI value on its own. It is the responsibility of a validation method to ensure the URI ownership via scheme-specific means authorized by the ACME client.

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

```
{"type": "uri", "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 URI 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 [I-D.ietf-acme-email-smime], this challenge splits the token into two parts. Each part reaches the client through a different channel: one via the ACME channel in the challenge object, the other via the DTN channel within the Challenge Bundle. The Key Authorization

result requires that the ACME client have access to the results of each channel to get both parts of the token.

The DTN Node ID Challenge SHALL only be allowed for URIs usable as a DTN Node ID, which are currently the schemes "dtn" and "ipn" as defined in [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 "uri" for the desired Node ID. From either of these entry points an authorization for the "uri" 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-part2> from the challenge object in accordance with Section 3.1.

3. The ACME client indicates to the BP agent the source and challenge <token-part2> which is authorized for use.

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 ACME client waits for indication from the BP agent that a Challenge Bundle has been received, including its <token-part1> payload.

6. The ACME client concatenates <token-part1> with <token-part2> (as text strings) and computes the Key Authorization in accordance with Section 8.1 of [RFC8555] using the full token and client account key digest.

7. The ACME client indicates to the BP agent the SHA-256 digest of the Key Authorization result, which results in a Response Bundle being sent 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-part2> is no longer usable.

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 "uri", 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" and a <token-part2>.

3. The ACME server sends one or more Challenge Bundles in accordance with Section 3.3. Each challenge bundle SHALL contain a distinct <token-part1> to be able to correlate with a response bundle. Computing an expected Key Authorization digest is not necessary until a response is received.

4. The ACME server waits for Response Bundle(s) for a limited interval of time. A default response interval, used when the challenge does not contain an RTT, SHOULD be a configurable parameter of the ACME server. If the ACME client indicated an RTT value in the challenge object, the response interval SHOULD be twice the RTT (with limiting logic applied as described below). The lower limit on response waiting time is network-specific, but SHOULD NOT be shorter than one second. The upper limit on response waiting time is network-specific, but SHOULD NOT be longer than one minute (60 seconds) for a terrestrial-only DTN. Responses are encoded in accordance with Section 3.4.

5. Once received and decoded, the ACME server checks the contents of each Response Bundle in accordance with Section 3.5. After all Challenge Bundles have either been responded to or timed-out, the validation procedure is successful only if all responses are successful.

An ACME server MAY send multiple challenges from different origins in the DTN network to avoid possible on-path attacks, as recommended in Section 10.2 of [RFC8555]. If responses are received from multiple challenges, any response failure SHALL cause a failure of the overall validation. Each response failure MAY be indicated to the ACME client as a validation subproblem.

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 validation token indicated by the ACME client. A BP agent SHALL respond to multiple challenges

with the same parameters. 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, string):** The source Node ID of bundles originating at the ACME server as a text URI.

**token-part2 (required, string):** A random value that uniquely identifies the challenge. This value MUST have at least 128 bits of entropy. It MUST NOT 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://example-acme-server/",
  "token-part2": "tPUZNY4ONIk6LxErRFEjVw"
}
```

The only over-the-wire data required by ACME for a Challenge Bundle is a nonce token, split into two parts, but the response data needs a client account key to generate the Key Authorization and its digest. The client account key is kept within the ACME client, the BP agent needs only the derived key thumbprint for its Response Bundle.

### 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.

### 3.3. ACME Node ID Validation Challenge Bundles

Each ACME Node ID Validation 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 identical to the Node ID being validated. The ACME server SHOULD NOT perform URI normalization on the Node ID given by the ACME client.

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

**Report-to Node ID:** The Report-to Node ID SHALL indicate the Node ID of the ACME server performing the challenge if status reports are requested.

**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 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 7.3.

**Administrative Record Content:** The Challenge Bundle administrative record content SHALL consist of a CBOR map containing one pair. The pair SHALL consist of key 1 with value of ACME challenge token-part1, represented as a CBOR byte string. The token-part1 is a random value that uniquely identifies the challenge. This value MUST have at least 128 bits of entropy. See [RFC4086] for additional information on randomness requirements.

An example full Challenge Bundle is included in Appendix B.1

Challenge Bundles SHOULD be BIB-signed in accordance with [I-D.ietf-dtn-bpsec] if the ACME server is capable of signing bundles. BP agents SHALL refuse to respond to a Challenge Bundle which is signed by a known ACME server but has an invalid signature. Challenge Bundles SHOULD NOT be directly encrypted (by BCB or any other method).

### 3.4. ACME Node ID Validation Response Bundles

Each ACME Node ID Validation 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 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 7.3.

**Administrative Record Content:** The Response 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-part1, copied from the Request Bundle, represented as a CBOR byte string. One pair SHALL consist of key 2 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.

An example full Response Bundle is included in Appendix B.2

Response Bundles MAY be BIB-signed in accordance with [I-D.ietf-dtn-bpsec] if the BP agent is capable of signing bundles. A BIB on the bundle gives no more security than the Key Authorization itself. Response Bundles SHOULD NOT be directly encrypted (by BCB or any other method).

### 3.5. 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 Response Bundle Source Node ID is identical to the Node ID being validated. The comparison of Node IDs SHALL use the comparison logic of [RFC3986] and scheme-based normalization of those schemes specified in [I-D.ietf-dtn-bpbis].

* The response payload contains the <token-part1> as sent in the Challenge Bundle. The response payload contains the expected Key Authorization digest computed by the ACME server. Because multiple Challenge Bundles can be sent to validate the same Node ID, the <token-part1> in the response is needed to correlate with the expected Key Authorization digest.

Any of the failures above SHALL cause the validation to fail. Any of the failures above SHOULD be indicated as subproblems to the ACME client.

## 4. 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] and [I-D.bsipos-dtn-bpsec-cose]. These purposes are referred to here as bundle security certificates.

One common behavior of bundle security certificates are the use of the Extended Key Usage key purpose "id-kp-bundleSecurity". Any CA

implementing the validation method defined in this document SHOULD also support issuing certificates with the bundle security Extended Key Usage.

### 4.1. Multiple Identity Claims

A single bundle security certificate request MAY contain a mixed set of SAN claims, including combinations of "ip", "dns", and "uri" 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.

### 4.2. 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 S/MIME 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 S/MIME 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 an S/MIME certificate suitable for both signing and encryption.

## 5. Implementation Status

[NOTE to the RFC Editor: please remove this section before publication, as well as the reference to [RFC7942] and [github-acme-dtnnodeid].]

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 has been created as a GitHub project [github-acme-dtnnodeid] and is intended to use as a proof-of-concept and as a possible source of interoperability testing. This example implementation only constructs encoded bundles and does not attempt to provide a full BP Agent interface.

## 6. Security Considerations

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

### 6.1. 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 ACME data itself is a random token (nonce) and a cryptographic signature, so there is no sensitive data to be leaked within the Node ID Validation bundle exchange.

Under certain circumstances, when BPSEC key material is available to the BP agent managed by the ACME client, the use of a BCB for the Request Bundle and/or Response Bundle can give additional confidentiality to the bundle metadata. This is not expected to be a general use case, as the whole point of ACME is to validate identifiers of untrusted client services.

### 6.2. 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.

One way to mitigate single-path on-path attacks is to attempt validation of the same Node ID via multiple bundle routing paths, as recommended in Section 3. 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.

Under certain circumstances, when BPSEC key material is available to the BP agent managed by the ACME client, the use of a BIB signature on the Response Bundle can give additional assurance that the response is coming from a valid BP agent.

### 6.3. 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 ACME Node ID challenges to a BP agent. The victim BP agent can ignore ACME challenges 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.

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. IANA Considerations

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

### 7.1. ACME Identifier Type

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

### 7.2. ACME Validation Method

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 |
|---|---|---|---|
| dtn-nodeid-01 | uri | Y | This specification |

Table 2

### 7.3. BP Bundle Administrative Record Types

Within the "Bundle Protocol" registry [IANA-BP], the following entry has been added to the "Bundle Administrative Record Types" sub-registry. [NOTE to the RFC Editor: For RFC5050 compatibility this value needs to be no larger than 15, but such compatibility is not needed. BPbis has no upper limit on this code point value.]

| Value | Description | Reference |
|---|---|---|
| TBD | ACME Node ID Validation | This specification |

Table 3

## 8. Acknowledgments

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

## 9. References

### 9.1. 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/>.

**[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. Birrane, "Bundle Protocol Version 7", Work in Progress, Internet-Draft, draft-ietf-dtn-bpbis-31, 25 January 2021, <https://tools.ietf.org/html/draft-ietf-dtn-bpbis-31>.

**[I-D.ietf-dtn-bpsec]** Birrane, E. and K. McKeever, "Bundle Protocol Security Specification", Work in Progress, Internet-Draft, draft-ietf-dtn-bpsec-26, 8 January 2021, <https://tools.ietf.org/html/draft-ietf-dtn-bpsec-26>.

### 9.2. Informative References

**[I-D.ietf-acme-email-smime]**

Melnikov, A., "Extensions to Automatic Certificate Management Environment for end-user S/MIME certificates", Work in Progress, Internet-Draft, draft-ietf-acme-email-smime-13, 20 November 2020, <https://tools.ietf.org/html/draft-ietf-acme-email-smime-13>.

**[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://tools.ietf.org/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-tcpclv4-24, 7 December 2020, <https://tools.ietf.org/html/draft-ietf-dtn-tcpclv4-24>.

**[I-D.bsipos-dtn-bpsec-cose]**
Sipos, B., "DTN Bundle Protocol Security COSE Security Contexts", Work in Progress, Internet-Draft, draft-bsipos-dtn-bpsec-cose-04, 22 December 2020, <https://tools.ietf.org/html/draft-bsipos-dtn-bpsec-cose-04>.

**[github-acme-dtnnodeid]** Sipos, B., "ACME Node ID Example Implementation", <https://github.com/BSipos-RKF/acme-dtnnodeid/>.

## 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-part1,
  ? key-auth-digest ; present for Response Bundles
}
token-part1 = (1 => bstr)
key-auth-digest = (2 => bstr)
```

## Appendix B. Example Bundles

[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-wow4m6DsqxbninhsWHlwfp0JecwQzYpOLmCQ"

And the ACME-server generated token-part2 (transported to the ACME client via HTTPS) has the base64url-encoded value of:

"tPUZNY4ONIk6LxErRFEjVw"

### B.1. Challenge Bundle

For the single challenge bundle in this example, the token-part1 (transported as byte string via BP) has the base64url-encoded value of:

"p3yRYFU4KxwQaHQjJ2RdiQ"

The minimal-but-valid Challenge Bundle is shown in Figure 1. 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'p3yRYFU4KxwQaHQjJ2RdiQ' / token-part1 /
      }
    ]>>
  ]
]
```

Figure 1: Example Challenge Bundle

### B.2. Response Bundle

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

```
"p3yRYFU4KxwQaHQjJ2RdiQtPUZNY4ONIk6LxErRFEjVw.LPJNul-wow4m6DsqxbninhsWHlwfp0JecwQzYpOLmCQ"
```

The minimal-but-valid Response Bundle is shown in Figure 2. 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 /
    <<[ / type-specific data /
      0xFFFF, / record-type-code /
      { / record-content /
        1: b64'p3yRYFU4KxwQaHQjJ2RdiQ', / token-part1 /
        2: b64'mVIOJEQZie8XpYM6MMVSQUiNPH64URnhM9niJ5XHrew' / key auth. digest /
      }
    ]>>
  ]
]
```

Figure 2: 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: BSipos@rkf-eng.com