Stream: Internet Engineering Task Force (IETF)
RFC: 9891
Category: Experimental
Published: November 2025
ISSN: 2070-1721
Author: B. Sipos
RKF Engineering

RFC 9891
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 that allows an ACME server to validate the Delay-Tolerant Networking (DTN) Node ID for an ACME client. A DTN Node ID is an identifier used in the Bundle Protocol (BP) to name a "singleton endpoint": an endpoint that is registered on a single BP Node. The DTN Node ID is encoded both as a certificate Subject Alternative Name (SAN) identity of type otherName with an Other Name form of `BundleEID` and as an ACME Identifier type "bundleEID".

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9891.

Copyright Notice

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) Certification Authorities (CAs) to validate network domain names of clients, the same mechanism can be used to validate any of the subject identity claims supported by the PKIX profile [RFC5280].

In this specification, the claim being validated is a Subject Alternative Name (SAN) identity of type `otherName` with an Other Name form of `BundleEID`, which is used to represent a Bundle Protocol (BP) Endpoint ID (EID) in a Delay-Tolerant Networking (DTN) overlay network. A DTN Node ID is any EID that can uniquely identify a BP Node, as defined in Section 4.2.5.2 of [RFC9171], which is equivalent to the EID being usable as a singleton endpoint. One common EID used as a Node ID is the Administrative EID, which is guaranteed to exist on any BP Node. At the time of writing, the URI schemes "dtn" and "ipn" as defined in Section 4.2.5.1 of [RFC9171] are

valid for a singleton endpoint and, thus, a Node ID. Because the `BundleEID` claim is new to ACME, a new ACME Identifier type "bundleEID" is needed to manage this claim within ACME messaging.

Once an ACME server validates a Node ID, either as a pre-authorization of an identifier type "bundleEID" as one of the authorizations of an order containing an identifier type "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 BP Agent with the new certificate is an implementation matter.

1.1. Scope

This document describes the ACME message contents [RFC8555], Bundle Protocol version 7 (BPv7) payloads [RFC9171], and Bundle Protocol Security (BPSec) operations [RFC9172] needed to validate claims of Node ID ownership.

This document does not address:

- Mechanisms for communication between an ACME client or ACME server and their associated BP Agent(s), depicted as steps 1, 2, 5, and 6 of Figure 1. This document only describes exchanges between ACME client-server pairs and between their respective associated BP Agents.
- Specific BP extension blocks or BPSec contexts necessary to fulfill the security requirements of this protocol. The exact security context needed, and its parameters, is 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 a Node ID to a 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 signing 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 EID is used as the destination of a Bundle and can indicate both a singleton or a group 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 the "Delay-Tolerant Networking Architecture" [RFC4838].

When an ACME client requests a pre-authorization or an order with an identifier type "bundleEID" (Section 2), the ACME server offers a "bp-nodeid-00" challenge type (Section 3) 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 Figure 1.

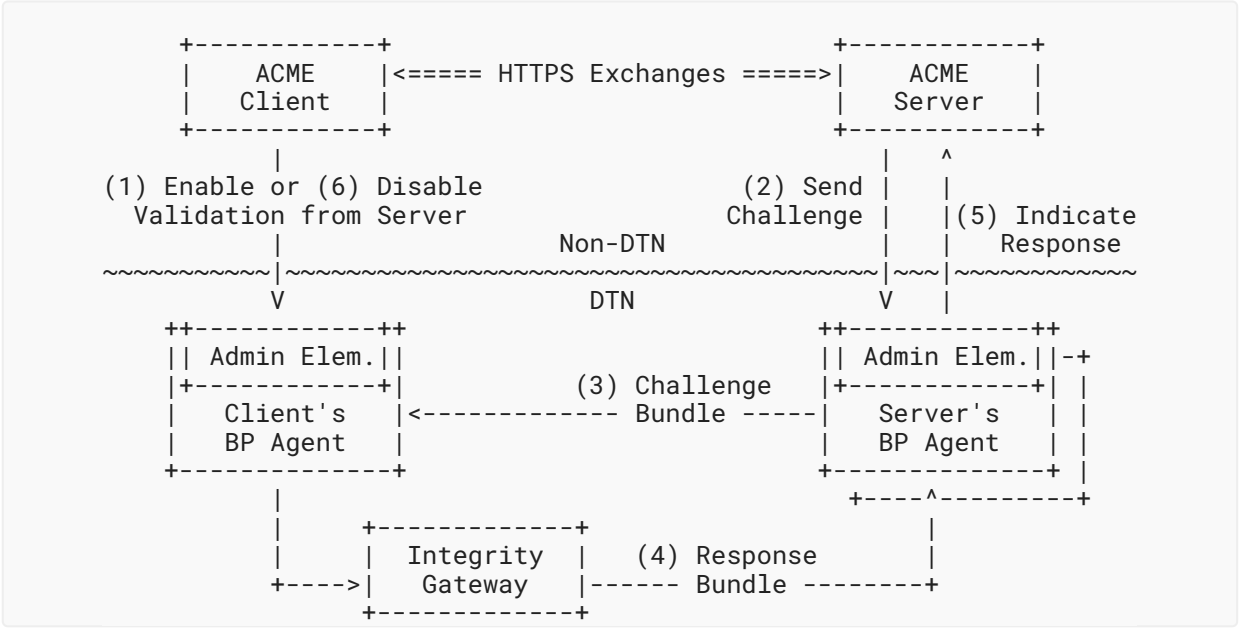

Figure 1: The Relationships and Flows Between Node ID Validation Entities

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 keep the Node ID unchanged while disambiguating from other service data Bundles.

There is nothing in this protocol that requires network-topological co-location of either the ACME client or ACME server with their respective associated BP Agent. While ACME requires a low-enough latency network to perform HTTPS exchanges between the 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 BP 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 Concise Binary Object Representation (CBOR) structure using the Concise Data Definition Language (CDDL) [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 brevity. These terms are as follows:

Challenge Object: This is a shortened form of the full "DTN Node ID Challenge Object". It is a JSON object created by the ACME server for challenge type "bp-nodeid-00" as defined in Section 3.1.

Response Object: This is a shortened form of the full "DTN Node ID Response Object". It is a JSON object created by the ACME client to authorize a challenge type "bp-nodeid-00" as defined in Section 3.2.

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 as defined in Section 3.3.

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 as defined in Section 3.4.

Because this is a document produced by the ACME WG, the following DTN BP terms are defined here to clarify how they are used by this ACME identifier type and validation mechanism.

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

Administrative EID: An Administrative EID is unique for a node within a specific URI scheme. Although any Node ID can be a valid Bundle Source and Destination, the Administrative EID 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; for the "ipn" scheme, this is the service number zero. These are formally defined under Section 4.2.5.1 of [RFC9171].

1.5. Experiment Scope

The emergent properties of DTN naming and BP security are still being developed and explored, especially between different organizational and administrative domains. Thus, the Experimental status of this document is related more to the practical utility of this kind of Node ID validation than to the validation method itself. The original use case is in large or cross-organizational networks where a BP Node can be trusted to be allocated and added to a network, but the method of certificate validation and issuance is desired to be in-band on the network rather than configured solely through a side channel using bespoke or manual protocols. Because this mechanism is so similar to other validation methods, specifically email address validation [RFC8823], it is expected to have few implementation difficulties or interoperability issues.

Part of the experimental nature of the validation method defined in Section 3, and BP more generally, is understanding its vulnerability to different kinds of on-path attacks. Some attacks could be based on the topology of the BP overlay network, while others could be based on the underlying (IP) network topology. Because not all of the attack surfaces of this validation method

are known or fully understood, the usefulness of the multi-perspective technique described in Section 3.5 is also not assured. The point of those multi-perspective requirements is that both the ACME client and server have consistent logic for handling the technique.

The usefulness of the integrity gateway defined in Section 4 to this validation method is experimental: the way that naming authority in a DTN is either allocated, delegated, or enforced is not a settled matter. How the organization running the CA (and its ACME server) can delegate some level of trust to a different organization running a connected DTN with a security gateway is also not defined. The organization running the integrity gateway would need to apply some minimal amount of policy to nodes running behind it, such as patterns to their Node IDs, which would behave conceptually similar to delegation of subdomains in the Domain Name System (DNS), but without the online interaction of DNS.

A successful experiment of this validation method would involve using the ACME protocol along with this Node ID validation to allow issuing of identity certificates across administrative domains. One possible scenario for this would be an issuing CA and an ACME server on the edge of a BP transit network operated by some agency. This transit network is used by edge nodes and accessed via edge routers of a peer network operated by a second agency. The nodes of the peer network are trusted by the second agency but not the first. The transit network can refuse to route traffic from the peer network that is not traceable to a valid identity certificate, which the edge nodes can obtain via the ACME server.

A valuable result from any experiment, even an unsuccessful one, would be feedback about this method to improve later versions. That feedback could include improvements to object or message structure, random versus deterministic token values, client or server procedures, or naming more generally.

2. Bundle EID ACME Identifier

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

Every ACME identifier type "bundleEID" **SHALL** have a value containing a text URI consistent with the requirements of Section 4.2.5.1 of [RFC9171] using one of the schemes available from the "Bundle Protocol URI Scheme Types" registry [IANA-BP]. Any identifier type "bundleEID" value that fails to properly percent-decode **SHALL** be rejected with an ACME error type of "malformed".

An ACME server supporting identifier type "bundleEID" **SHALL** decode and normalize (based on scheme-specific syntax) all such received identifier values. Any identifier type "bundleEID" value for which the scheme-specific part does not match the scheme-specific syntax **SHALL** be rejected with an ACME error type of "malformed". Any identifier type "bundleEID" value that uses a scheme not handled by the ACME server **SHALL** be rejected with an ACME error type of "rejectedIdentifier".

When an ACME server needs to request proof that a client controls a Bundle EID, it **SHALL** create an authorization with the identifier type "bundleEID". An ACME server **SHALL NOT** attempt to dereference a Bundle EID value on its own. It is the responsibility of an ACME validation method to ensure the EID ownership using a method 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 Bundle EID

A PKIX certificate SAN identity containing an Other Name form of `BundleEID` can hold any EID value (as a text URI). But the certificate profile used by the TCP Convergence Layer [RFC9174] and supported by the ACME validation method in Section 3 requires that the value hold a Node ID (Section 1.4).

In addition to the narrowing of that certificate profile, this validation method requires that the client's BP Agent respond to administrative records sent to the Node ID being validated. Typically, this is limited to an Administrative EID (Section 1.4) destination. However, the administrative element of a BP Node is not prohibited from receiving administrative records for, or 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 EIDs, 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 email-address validation [RFC8823], this challenge splits the token into several parts, each part 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 similar to the challenge "from" used by email validation in Section 3 of [RFC8823].

The token parts are as follows:

`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; thus, 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 having 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 Object **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** provide 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 type "bundleEID" for the desired Node ID. From either of these entry points, an authorization for the identifier type "bundleEID" 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 (Section 3.1) contained in an authorization object associated with the order in accordance with Section 7.1.4 of [RFC8555].
3. The ACME client indicates to the administrative element of its BP Agent the `id-chal` that is authorized for use along with the associated `token-chal` and client account key thumbprint. The ACME client waits, if necessary, until the Agent is configured before proceeding to the next step.
4. The ACME client POSTs a response object (Section 3.2) to the challenge URL on the ACME server in 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 type "bundleEID", where the URI value is a Node ID.
2. The ACME server generates an authorization for the Node ID with challenge type "bp-nodeid-00" 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 contains a distinct, random `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 a Response Bundle(s) for a limited interval of time (based on the 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 that 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 Object

The DTN Node ID Challenge Object is included by the ACME server as defined in Section 7.5 of [RFC8555] when it supports validating Node IDs.

The DTN Node ID Challenge Object has the following content:

type (required, string): The string "bp-nodeid-00".

id-chal (required, string): This is a random value associated with a challenge that allows a client to filter valid Challenge Bundles (Section 3.3). 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 BCP 106 [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 BCP 106 [RFC4086] for additional information on randomness requirements.

```
{
  "type": "bp-nodeid-00",
  "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 Response Object

The DTN Node ID response object is sent by the ACME client when it authorizes validation of a Node ID as defined in Section 7.5.1 of [RFC8555]. Because a DTN has the potential for significantly longer (but roughly predictable) 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 non-DTN network delays (on the order of seconds).

The DTN Node ID 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 response object **SHALL NOT** be sent until the BP Agent has been configured to properly respond to the challenge. 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) regarding 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 the response interval is network specific, but it **SHOULD NOT** be shorter than one second. The upper limit on response interval is network specific, but it **SHOULD NOT** be longer than one minute (60 seconds) for a terrestrial-only DTN.

3.3. ACME Node ID Validation Challenge Bundle

Each ACME Node ID Validation Challenge Bundle **SHALL** be structured and encoded in accordance with BPv7 [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 the ACME server will accept responses to this challenge.

Administrative Record Type Code: This is 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 the following three pairs:

- One pair **SHALL** consist of key 1 with a 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 a 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 BCP 106 [RFC4086] for additional information on randomness requirements.

- One pair **SHALL** consist of key 4 with a 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 [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 that use this validation method **SHALL** support the hash algorithm "SHA-256" (value -16) [RFC9054]. This requirement allows for different implementations to be configured to use an interoperable algorithm, but does not preclude the use of other algorithms with either higher or lower priority.

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 in accordance with Section 4.4.2 of [RFC9171].

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 the Block Confidentiality Block (BCB) method or any other method (see Section 6.1).

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 that covers at least the primary block and payload. That BIB has a Security Source that is trusted and passes security-context-specific validation (i.e., Message Authentication Code (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` matching the definition in Section 3.3.
- The challenge payload contains at least one hash algorithm identifier acceptable to the client.

Failure to match any of the above **SHALL** cause the Challenge Bundle to be otherwise ignored by the BP Agent. It is an implementation matter of how to react to such failures, which could include logging the event, incrementing counters, or raising alarms.

3.4. ACME Node ID Validation Response Bundles

Each ACME Node ID Validation Response Bundle **SHALL** be structured and encoded in accordance with BPv7 [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 7.3.

Administrative Record Content: The Response Bundle administrative record content **SHALL** consist of a CBOR map containing the following 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 [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 6.1 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 that covers at least the primary block and payload. That BIB has a Security Source that 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 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 is used by the Let's Encrypt service in its operational deployment [LE-multi-perspective]. The utility of a multi-perspective validation is part of the experimental nature (see Section 1.5) of this specification.

When required by policy, an ACME server **SHALL** send multiple Challenge Bundles from different sources in the BP 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 is 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 that closely resembles the function of DKIM email signing [RFC6376]. In this mechanism, a routing node in a DTN subnetwork 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, it only needs to trust this Security Source node. The receiver needs policy configuration to know which Security Sources are permitted to attest for which Bundle sources. The utility of an integrity gateway is part of the experimental nature (Section 1.5) of this specification.

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 it 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 BPSec [RFC9172] requirements. 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 6.2).

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] and Section 4 of [BPSEC-COSE]. These purposes are referred to here as "Bundle security certificates".

The ACME identifier type "bundleEID" correlates to the PKIX certificate and certificate request "NODE-ID" as defined in Section 4.4.1 of [RFC9174]. This NODE-ID is present in certificate requests with an `extensionRequest` attribute (see [RFC2985]) containing a SAN extension with identities of type `otherName` having an Other Name form of `BundleEID`, identified by `id-on-bundleEID` from the "SMI Security for PKIX Other Name Forms" registry [IANA-SMI]. Because the `BundleEID` Other Name form is encoded as an IA5String, it **SHALL** be treated as being in the percent-encoded form of Section 2.1 of [RFC3986].

One defining aspect of Bundle security certificates is the Extended Key Usage key purpose `id-kp-bundleSecurity` [IANA-SMI], as defined in Section 4.4.2.1 of [RFC9174]. When requesting a certificate that includes a NODE-ID, 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, 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 identifiers, including combinations of IP-ID [RFC9525], DNS-ID [RFC9525], and NODE-ID [RFC9174] types. These correspond with ACME identifier types "ip", "dns", and "bundleEID", respectively.

There is no restriction on how a certificate combines these claims, but each identifier **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 ACME [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. Existing validation mechanisms are defined for identifier types "ip" [RFC8738] and "dns" [RFC8555] among others [IANA-ACME].

The specific use case of TLS-based security for BPv7 CL transport in Section 4.4 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. These authentications apply to each identifier type, used for different network layers, at different points during secure session establishment.

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. The validity of a request for either a restricted-use or unrestricted-use certificate is dependent on both CA policy to issue such certificates as well as constraints based on the requested key and algorithm types.

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

In order to request an encryption-only Bundle security certificate, the CSR **SHALL** include the key usage extension with the 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 the ACME server to issue a Bundle security certificate suitable for both signing and encryption.

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 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 Bundles and/or Response Bundles while they traverse untrusted networks, but that is a DTN configuration matter outside of the scope of this document.

6.2. Threat: BP Node Impersonation

As described in Section 10.1 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, the Bundle source node as defined in Section 4. This is functionally similar to the DKIM signing [RFC6376] and provides some level of secure Bundle origination.

Another way to mitigate on-path attacks is to attempt validation of the same Node ID from multiple sources, hopefully 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) could be employed.

6.3. 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 would use multi-path forwarding of a singleton-destination Bundle.

Ultimately, the point of the ACME Bundle exchange is to derive a Key Authorization value and its cryptographic digest, and to communicate that digest 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 6.4.

6.4. 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 that 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 that 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.

6.5. Inherited Security Considerations

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

6.6. 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.

7. IANA Considerations

This specification adds to the "Automated Certificate Management Environment (ACME) Protocol" registry group and the "Bundle Protocol" registry group.

7.1. ACME Identifier Types

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

Label	Reference
bundleEID	RFC 9891

Table 1

7.2. ACME Validation Methods

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

Label	Identifier Type	ACME	Reference
bp-nodeid-00	bundleEID	Y	RFC 9891

Table 2

7.3. Bundle Administrative Record Types

Within the "Bundle Protocol" registry group [IANA-BP], the following entry has been added to the "Bundle Administrative Record Types" registry.

Bundle Protocol Version	Value	Description	Reference
7	255	ACME Node ID Validation	RFC 9891

Table 3

8. References

8.1. 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 (MIB Module Registrations)", <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>.

[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>.

[RFC9054] Schaad, J., "CBOR Object Signing and Encryption (COSE): Hash Algorithms", RFC 9054, DOI 10.17487/RFC9054, August 2022, <https://www.rfc-editor.org/info/rfc9054>.

[RFC9171] Burleigh, S., Fall, K., and E. Birrane, III, "Bundle Protocol Version 7", RFC 9171, DOI 10.17487/RFC9171, January 2022, <https://www.rfc-editor.org/info/rfc9171>.

[RFC9172] Birrane, III, E. and K. McKeever, "Bundle Protocol Security (BPSec)", RFC 9172, DOI 10.17487/RFC9172, January 2022, <https://www.rfc-editor.org/info/rfc9172>.

[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>.

[RFC9525] Saint-Andre, P. and R. Salz, "Service Identity in TLS", RFC 9525, DOI 10.17487/RFC9525, November 2023, <https://www.rfc-editor.org/info/rfc9525>.

8.2. Informative References

[BPSEC-COSE] Sipos, B., "Bundle Protocol Security (BPSec) COSE Context", Work in Progress, Internet-Draft, draft-ietf-dtn-bpsec-cose-10, 15 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-dtn-bpsec-cose-10>.

[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>.

[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>.

[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>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

[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>.

Appendix A. Administrative Record Types CDDL

The extension of BPv7 from Appendix B of [RFC9171] for the ACME Bundles in Sections 3.3 and 3.4 is the following CDDL:

```
; All ACME records have the same structure
$admin-record /= [0xFF, acme-record]
acme-record = acme-challenge-record / acme-response-record
acme-challenge-record = {
  id-chal,
  token-bundle,
  alg-list
}
acme-response-record = {
  id-chal,
  token-bundle,
  key-auth-digest
}

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 COSE registry, only hash algorithms allowed
alg-id = tstr / int
```

Appendix B. Example Authorization

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-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 /
      0xFF, / record-type-code /
      { / record-content /
        1: b64'dDtaviYTPUWFS3NK37YWfQ', / id-chal /
        2: b64'p3yRYFU4KxwQaHQjJ2RdiQ', / token-bundle /
        4: [-16] / alg-list: SHA-256 /
      }
    ]>>
  ]
]
```

Figure 2: Example Challenge Bundle

B.2. Response Bundle

When the tokens are combined with the key thumbprint, the full Key Authorization value is the following, folded across lines for readability using the "single backslash" strategy of Section 7 of [RFC8792].

```
/ NOTE: '\' line wrapping per RFC 8792 /

"p3yRYFU4KxwQaHQjJ2RdiQtPUZNY4ONIk6LxErRFEjVw.\
LPJNul-wow4m6DsqxbninhsWHlwfp0JecwQzYpOLmCQ"
```

The minimal-but-valid Response Bundle is shown in Figure 3. This response indicates that there are 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 /
      0xFF, / record-type-code /
      { / record-content /
        1: b64'dDtaviYTPUWFS3NK37YWfQ', / id-chal /
        2: b64'p3yRYFU4KxwQaHQjJ2RdiQ', / token-bundle /
        3: [-16, b64'mVIOJEQZie8XpYM6MMVSQUiNPH64URnhM9niJ5XHrew']
        / SHA-256 key auth. digest /
      }
    ]>>
  ]
]
```

Figure 3: Example Response Bundle

Acknowledgements

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

The workflow and terminology of this validation method were originally copied from the work of Alexey Melnikov for email validation [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