HTTPAPI
Internet-Draft
Intended status: Standards Track
Expires: 6 November 2026

J. McGraw
TaskHawk

5 May 2026

Protocol 427: An HTTP Budget-Required Status Code with Post-Quantum-Signed Budget Attestations

draft-mcgraw-httpapi-agent-budget-00

Abstract

Internet-deployed software agents are increasingly authorized to spend money, consume metered services, or commit other resources on behalf of human or organizational principals. Existing HTTP authentication and payment patterns conflate two orthogonal concerns: whether the requester holds a credential at all (the "401" axis), and whether the requester has been authorized to spend a specific amount through a specific settlement rail (the "budget" axis). This document defines the 427 (Budget Required) HTTP status code, the "Budget" HTTP authentication scheme, a CBOR-encoded Budget-Attestation envelope signed with a post-quantum digital signature algorithm, and a version-negotiation mechanism using the existing 426 (Upgrade Required) status code. The mandatory primary signature uses ML-DSA-87 (FIPS 204). An optional "rail-keyed" signature, computed with a hash-based stateless signature algorithm, provides cryptographic diversification for settlement-rail submissions.

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-mcgraw-httpapi-agent-budget/.

Discussion of this document takes place on the HTTPAPI Working Group mailing list (mailto:httpapi@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/httpapi/. Subscribe at https://www.ietf.org/mailman/listinfo/httpapi/.

Status of This Memo

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

This Internet-Draft will expire on 6 November 2026.

Copyright Notice

Table of Contents

1. Introduction

Software agents acting on behalf of human or organizational principals increasingly need to make HTTP requests that may incur cost, consume metered services, or trigger settlement on a payment rail. The HTTP authentication framework [RFC9110] as historically deployed addresses "who are you" questions through schemes such as Basic, Bearer, and DPoP [RFC9449], and the HTTP 402 (Payment Required) status code has been appropriated by the L402 protocol [L402] and the x402 protocol [X402] to convey "you must pay before proceeding" semantics.

Neither pattern cleanly captures the question this specification addresses: "do you hold a recently-issued, cryptographically-attested authorization to spend up to a stated amount, valid for this

request?" That question is orthogonal to the bearer-credential axis; an agent may hold a perfectly valid bearer token and still lack any spending authority, or hold a spending authority and need to present it for any of several requests across one or more origins.

This document defines:

* The 427 (Budget Required) HTTP status code (Section 3), used by an origin server or gateway to indicate that a request will not be processed until the requester presents a valid Budget-Attestation.

* The "Budget" HTTP authentication scheme (Section 4), used in the WWW-Authenticate response header field of a 427 response and in the Authorization request header field of subsequent requests.

* The Budget-Attestation envelope (Section 5), a CBOR-encoded [RFC8949], COSE-signed [RFC9052] structure that carries semantic claims about the issuer, the agent, the bound request, the permitted settlement rails, and the spending amount. The envelope's primary signature uses ML-DSA-87 [FIPS204]; an optional rail-keyed signature (Section 5.4) uses any IETF-registered SLH-DSA [FIPS205] parameter set.

* A version-negotiation mechanism (Section 6) using a Protocol-427-Version response field [RFC9651] and the existing 426 (Upgrade Required) status code [RFC9110] for cases in which the requester presents a Protocol-427 version the origin does not support.

1.1. Motivation and Use Cases

A representative scenario is an autonomous research agent that has been issued, by its operating organization, a cryptographically-attested allowance of "USD 10 over the next 60 seconds, spendable through any of {x402, L402, mpp}". The agent traverses several origins, some of which require payment. At each origin requiring payment, the agent presents the same Budget-Attestation; each origin verifies the attestation and either processes the request directly or initiates a rail-specific settlement that consumes from the attested allowance.

The benefits of attesting authority separately from invoking a settlement rail are: (1) operators retain control over how much an agent can spend without participating in every transaction; (2) origins can verify authorization without coupling to any specific payment rail's protocol; (3) cryptographic agility, including post-quantum agility, can be applied uniformly to the authorization layer without modifying any settlement-rail protocol.

1.2. Conventions and Definitions

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

The following terms are used throughout this document:

Operator: An entity that issues Budget-Attestations on behalf of a principal.

Agent: An entity that holds and presents Budget-Attestations issued by an Operator in order to make HTTP requests.

Verifier: An origin server, gateway, or intermediary that receives a Budget-Attestation, validates its signatures, and decides whether to process the bearing request.

Budget-Attestation: The CBOR-encoded, signed envelope defined in Section 5.

Settlement Rail: An out-of-band protocol used to effect actual transfer of value associated with a Budget-Attestation. Examples include x402, L402, and Lightning multi-path payments ("mpp"). Rail names are carried as text strings in the Budget-Attestation; this document does not define how settlements are performed.

Rail-Keyed Signature: An optional second signature on the Budget-Attestation envelope, computed with a stateless hash-based signature algorithm (SLH-DSA), used to provide cryptographic diversification when the attestation is presented in connection with a settlement-rail submission.

1.3. Relationship to HTTP Message Signatures

[RFC9421] defines a generic mechanism for digitally signing components of individual HTTP messages. Although both that mechanism and Budget-Attestation involve cryptographic signatures that travel with HTTP traffic, they address different concerns.

RFC 9421 signatures bind a signature to a single HTTP message; their subject is "the message". A Budget-Attestation, by contrast, is a portable spending authority issued by an Operator and presented by an Agent across one or more requests; its subject is "the authority", and it carries semantic claims (issuer, agent, expiry, permitted settlement rails, amounts) that are not part of any HTTP message component.

[RFC9421] references signing keys by keyid and, as of the publication of this document, registers no post-quantum signature algorithm in the "HTTP Signature Algorithms" registry. This document specifies CBOR encoding [RFC8949] and COSE signing structures [RFC9052] for the Budget-Attestation envelope so that multi-kilobyte post-quantum signature sizes are accommodated outside HTTP header fields.

The two mechanisms are complementary. An Agent MAY carry a Budget-Attestation in a request that is itself signed per [RFC9421]; in that case, the RFC 9421 signature covers the request as transmitted, and the embedded Budget-Attestation covers the spending authority irrespective of transport.

1.4. Venue and Coordination

This document is submitted to the HTTPAPI Working Group. It defines protocol elements that, per the HTTPAPI charter and Section 4.6 of [RFC9205], require coordination with the HTTP Working Group. In particular, registration of the 427 status code in the "HTTP Status Codes" registry is subject to IETF Review under Section 16.2.1 of [RFC9110], and the author expects that registration to be reviewed by the HTTP Working Group prior to IESG approval.

This document follows the best practices of [RFC9205] for applications that use HTTP. Where it deviates -- specifically, by defining a new HTTP status code -- it does so consistent with the guidance of Section 4.6 of [RFC9205].

1.5. Open Issues

This subsection is to be removed before Working Group adoption or publication.

* The CBOR tag value for the tagged form of the Budget-Attestation envelope is marked TBD pending IANA assignment.

* The default reason-code registry policy (Section 8.7) is set to Specification Required; alternative policies may be considered during Working Group review.

* The author seeks feedback on whether the rail-keyed signature (Section 5.4) belongs in this document or in a separate companion document focused on settlement-rail bindings.

2. Overview of Operation

A typical Protocol 427 exchange proceeds as follows.

```
GET /research/papers/12345 HTTP/1.1
Host: api.example
```

The origin determines that processing the request will incur cost and that no Budget-Attestation accompanies the request. It returns:

```
HTTP/1.1 427 Budget Required
Date: Tue, 05 May 2026 14:00:00 GMT
Cache-Control: no-store
Content-Type: application/problem+json
Content-Length: 213
Protocol-427-Version: 1
WWW-Authenticate: Budget realm="api.example",
                  alg="ML-DSA-87",
                  rails="x402 l402 mpp",
                  nonce="QMjVqg5Xb6yV0bO_t9X8gQ",
                  max-age=900

{
  "type": "https://taskhawktech.com/problems/budget-required",
  "title": "Budget attestation required",
  "status": 427,
  "detail": "A valid Budget-Attestation is required.",
  "min-budget": {"USD": 250},
  "accepted-rails": ["x402","l402","mpp"],
  "max-age": 900,
  "protocol-version": 1
}
```

The Agent obtains a Budget-Attestation from its Operator (out of band; issuance is not specified by this document) and retries:

```
POST /research/papers/12345 HTTP/1.1
Host: api.example
Authorization: Budget attestation=":2BhA...base64url-CBOR...kQ=="
Content-Length: 0
```

If the attestation validates against the Verifier's policy, the Verifier processes the request normally. If validation fails, the Verifier returns 427 again with a reason extension member in the Problem Details body indicating the failure (Section 7).

3. The 427 (Budget Required) Status Code

The 427 (Budget Required) status code indicates that the requester must present a valid Budget-Attestation as defined in this document before the request can be processed. The 427 status code differs from 401 (Unauthorized) in that the requester may hold a fully-valid authentication credential (for example, a Bearer token) and still receive 427 because the request requires a budget attestation in addition to, or instead of, that credential. It differs from 402 (Payment Required) in that 427 is a request for evidence of pre-issued spending authority, not a request to immediately initiate payment.

A 427 response MUST include a Protocol-427-Version response header field (Section 6.1) and a WWW-Authenticate header field specifying the "Budget" authentication scheme (Section 4).

A 427 response SHOULD include a response body in the "application/problem+json" [RFC9457] or "application/problem+cbor" [RFC9457] media type carrying diagnostic information. When present, the body SHOULD use the "budget-required" problem type defined in Section 8.6.

3.1. Cacheability

Responses with the 427 status code are not cacheable by default; per Section 15.1 of [RFC9110], status codes not enumerated as cacheable are not cacheable absent explicit cache directives. A server SHOULD send Cache-Control: no-store (Section 5.2.2.5 of [RFC9111]) with a 427 response, and a cache MUST NOT store a 427 response unless explicitly permitted by Cache-Control directives in the response. This rule mirrors the treatment of 428, 429, and 431 in [RFC6585].

3.2. Relationship to 401, 402, 426

* 401 (Unauthorized) is returned when no, or insufficient, authentication credential is supplied. A request MAY receive 401 for credential reasons even if a Budget-Attestation is present.

* 402 (Payment Required) is "reserved for future use" by Section 15.5.2 of [RFC9110] but has been adopted by L402 and x402 to mean "initiate a payment-rail interaction". A request MAY receive 402 from a downstream rail-handling endpoint after a 427 has been satisfied, depending on deployment.

* 426 (Upgrade Required) is used by this document to negotiate Protocol-427 versions; see Section 6.

4. The "Budget" Authentication Scheme

The "Budget" authentication scheme is registered in Section 8.2. It is used in the WWW-Authenticate response header field of a 427 response and in the Authorization request header field of a subsequent request.

4.1. Challenge Syntax

A WWW-Authenticate challenge using the "Budget" scheme has the following syntax, expressed in ABNF [RFC9110]:

```
budget-challenge   = "Budget" 1*SP budget-params
budget-params      = budget-param *( OWS "," OWS budget-param )
budget-param       = ( "realm" "=" quoted-string )
                   / ( "alg" "=" quoted-string )
                   / ( "rails" "=" quoted-string )
                   / ( "nonce" "=" quoted-string )
                   / ( "max-age" "=" 1*DIGIT )
                   / ( "attestation" "=" quoted-string )
                   / auth-param
```

Parameter semantics:

realm: A protection-space identifier as defined in Section 11.5 of [RFC9110]. REQUIRED.

alg: A space-separated list of acceptable signature algorithm names for the Budget-Attestation primary signature. Tokens correspond to COSE algorithm names ([I-D.ietf-cose-dilithium]). REQUIRED.

rails: A space-separated list of settlement-rail tokens acceptable to the Verifier. REQUIRED if the Verifier expects a rail-keyed signature (Section 5.4); OPTIONAL otherwise.

nonce: A server-supplied opaque value, base64url-encoded, that the Agent SHOULD echo in the nonce claim of any presented Budget-Attestation bound to this challenge. REQUIRED.

max-age: An advisory, non-negative integer indicating, in seconds, the maximum recommended lifetime of any Budget-Attestation presented in response to this challenge. OPTIONAL.

attestation: Reserved for future use; servers MUST NOT include it in challenges emitted under this version of Protocol 427.

4.2. Credentials Syntax

An Authorization request header field using the "Budget" scheme has the following syntax:

```
budget-credentials = "Budget" 1*SP "attestation" "=" quoted-string
```

The value of the attestation parameter is the base64url Section 5 of [RFC8949] encoding of the CBOR Budget-Attestation envelope (Section 5).

When the encoded envelope would exceed an HTTP-implementation header size limit (a typical limit is 8 KiB; SLH-DSA signatures (Section 5.4) can exceed this), the Agent MUST instead submit the envelope in a request body of media type "application/budget-attestation+cose" (Section 8.5) and convey only the scheme name and a content-binding parameter in the Authorization header field, as follows:

```
budget-credentials-body = "Budget" 1*SP "binding" "=" quoted-string
```

The binding value is the base64url-encoded SHA-256 hash of the CBOR-encoded envelope contained in the request body.

5. The Budget-Attestation Envelope

The Budget-Attestation envelope is a COSE_Sign structure [RFC9052] carrying a Budget-Claims set as its payload. It MAY be transported as a tagged CBOR value (CBOR tag TBD) or as an untagged value when delivered as the body of an "application/budget-attestation+cose" HTTP message.

5.1. CDDL Definition

The schema below is given in CDDL [RFC8610].

```
; Budget-Attestation envelope (Protocol 427)
; The tagged form uses CBOR tag TBD; the untagged form is used in
; HTTP message bodies of media type
; application/budget-attestation+cose.

Budget-Attestation = #6.TBD(Budget-Attestation-Untagged) /
                     Budget-Attestation-Untagged

Budget-Attestation-Untagged = [
  protected   : bstr .cbor Protected-Header,
  unprotected : Unprotected-Header,
  claims      : bstr .cbor Budget-Claims,
```

```
  signatures  : [+ Signature]
]

Protected-Header = {
  1 => int,           ; alg of the primary signature (ML-DSA-87)
  ? 4 => bstr         ; kid (operator key identifier)
}

Unprotected-Header = {
  * (int / tstr) => any
}

Budget-Claims = {
  "version"  => uint,                ; protocol version
  "iss"      => tstr,                ; issuer (operator) identifier
  "agent"    => tstr,                ; agent identifier
  "iat"      => uint,                ; issued-at, seconds since epoch
  "exp"      => uint,                ; expiry, seconds since epoch
  "nonce"    => bstr .size (16..64), ; replay-protection nonce
  "rb"       => Request-Binding,
  "rails"    => [+ tstr],            ; permitted settlement rails
  ? "amt"    => Rail-Amount-Map      ; amount cap (minor units)
}

Request-Binding = {
  "method"  => tstr,                ; e.g., "POST"
  "uri-h"   => bstr .size 32,       ; SHA-256 of canonical target URI
  ? "body-h" => bstr .size 32       ; SHA-256 of body, if bound
}

Rail-Amount-Map = {
  + tstr => uint                    ; minor units per currency code
}

Signature = [
  protected   : bstr .cbor Sig-Protected-Header,
  unprotected : Unprotected-Header,
  signature   : bstr
]

Sig-Protected-Header = {
  1 => int,                         ; alg id (COSE)
  ? 4 => bstr,                      ; kid
  ? "role" => "operator" / "rail"   ; primary vs rail-keyed
}
```

5.2. Canonical Encoding

Budget-Attestations MUST be encoded using the Core Deterministic Encoding Requirements of Section 4.2.1 of [RFC8949]. Verifiers MUST reject envelopes whose CBOR encoding does not satisfy those requirements.

5.3. Primary Signature

The Budget-Attestation envelope MUST include exactly one primary signature. The primary signature’s Sig-Protected-Header MUST set alg to the COSE identifier for ML-DSA-87 as defined by [I-D.ietf-cose-dilithium]. Future revisions of this document MAY add additional MUST-implement primary algorithms; the rules of algorithm agility (Section 9.3) apply.

5.4. Rail-Keyed Signature

A Budget-Attestation envelope MAY include a second signature, denoted the "rail-keyed" signature, distinguished by role = "rail" in its Sig-Protected-Header. The rail-keyed signature provides cryptographic diversification: even if the primary lattice-based signature key is compromised, settlement-rail interactions can require an additional, hash-based signature over the same envelope.

The rail-keyed signature, when present, MUST use a stateless hash-based signature algorithm registered in the COSE Algorithms Registry under [I-D.ietf-cose-sphincs-plus]. Implementations of this document MUST support SLH-DSA-SHA2-128f as the rail-keyed algorithm ([FIPS205]); implementations MAY additionally support any other SLH-DSA parameter set with a registered COSE algorithm identifier.

A Verifier that requires a rail-keyed signature for settlement-rail acceptance MUST advertise this expectation through deployment-specific means; this document does not extend the WWW-Authenticate challenge to advertise per-rail signature requirements.

Because SLH-DSA signatures are large (FIPS 205 reports 17,088 bytes for SLH-DSA-SHA2-128f and may be tens of kilobytes for higher parameter sets), envelopes carrying a rail-keyed signature MUST be transported in an HTTP message body of media type "application/budget-attestation+cose"; they MUST NOT be carried in the Authorization header field’s attestation parameter.

5.5. Verification

A Verifier processing a Budget-Attestation MUST, in order:

1. Verify that the encoded CBOR satisfies the Core Deterministic Encoding Requirements of Section 4.2.1 of [RFC8949].

2. Verify that the version claim is supported (Section 6).

3. Verify the primary signature against the operator key identified by the kid parameter or otherwise resolved by deployment policy.

4. Verify that iat and exp define a non-empty interval that includes the current time, allowing for clock skew of at most 60 seconds.

5. Verify that the nonce was issued by the Verifier as part of an outstanding 427 challenge, and has not previously been observed for this Verifier.

6. Verify that rb.method matches the request method, that rb.uri-h matches the SHA-256 of the canonical target URI, and (if body-h is present) that body-h matches the SHA-256 of the request body.

7. Verify that any rail-keyed signature, if present, validates under a configured rail key.

If all checks pass, the Verifier processes the request normally. If any check fails, the Verifier MUST respond with 427 and SHOULD include a reason extension member in the Problem Details body indicating the failure.

6. Versioning

This document defines version 1 of Protocol 427. Future revisions MAY define additional versions.

6.1. The Protocol-427-Version Field

The Protocol-427-Version response header field is a Structured Field [RFC9651] of type Item containing an Integer. Its value is the Protocol-427 version under which the response was constructed.

A 427 response MUST include exactly one Protocol-427-Version header field. Other responses MAY include the field to advertise the origin’s Protocol-427 capability.

6.2. Version Negotiation via 426

If a request bearing an Authorization: Budget credential carries a version claim that the Verifier does not support, the Verifier MUST respond with 426 (Upgrade Required) per Section 15.5.22 of [RFC9110] and MUST include an Upgrade header field listing one or more Protocol-427/N tokens for versions the Verifier supports. The token "Protocol-427" is registered in Section 8.4.

Example:

```
HTTP/1.1 426 Upgrade Required
Upgrade: Protocol-427/1
Connection: Upgrade
Protocol-427-Version: 1
```

7. Reason Codes

A Verifier returning 427 SHOULD include a reason extension member in the Problem Details body indicating the cause of failure. This document defines the following initial reason codes:

Reason	Meaning
ok	Used in audit and diagnostic contexts only; never appears on a 427.
expired	The presented attestation’s exp is in the past.
signature_mismatch	A signature on the envelope did not validate.
replay	The presented nonce has previously been observed.
unknown_operator	The iss is not recognized.
malformed_cbor	The envelope failed CBOR canonicalization checks.
version_unsupported	The version claim is not supported.
rail_not_authorized	A rail in rails is not permitted by policy.
revoked	The operator key has been revoked.

Table 1

A registry for additional reason codes is established in Section 8.7.

8. IANA Considerations

This document creates registrations in several IANA registries.

8.1. Status Code 427

IANA is requested to register the following entry in the "HTTP Status Codes" registry [IANA.HTTP.StatusCodes] established by Section 16.2.1 of [RFC9110]:

Value	Description	Reference
427	Budget Required	Section 3 of this document

Table 2

8.2. "Budget" Authentication Scheme

IANA is requested to register the following scheme in the "Hypertext Transfer Protocol (HTTP) Authentication Scheme Registry" [IANA.HTTP.AuthSchemes] established by Section 16.4.1 of [RFC9110]:

Authentication Scheme Name	Reference	Notes
Budget	Section 4 of this document	Conveys a CBOR-encoded, post-quantum signed Budget-Attestation; see Section 5.

Table 3

8.3. "Protocol-427-Version" Field

IANA is requested to register the following entry in the "HTTP Field Name Registry" [IANA.HTTP.Fields] established by Section 18.4 of [RFC9110]:

Field Name	Status	Reference	Comments
Protocol-427-Version	permanent	Section 6.1 of this document	Structured Field, Item of type Integer; see [RFC9651].

Table 4

8.4. "Protocol-427" Upgrade Token

IANA is requested to register the following entry in the "HTTP Upgrade Token Registry" [IANA.HTTP.UpgradeTokens] established by Section 16.7 of [RFC9110]:

Value	Description	Expected Version Tokens	Reference
Protocol-427	Agent Budget Negotiation Protocol	A non-negative integer denoting the Protocol-427 version	Section 6 of this document

Table 5

8.5. "application/budget-attestation+cose" Media Type

IANA is requested to register the following media type [RFC6838] in the "Media Types" registry [IANA.MediaTypes]:

Type name: application

Subtype name: budget-attestation+cose

Required parameters: N/A

Optional parameters: cose-type (when present, MUST be identical to its usage for application/cose, per Section 2 of [RFC9052])

Encoding considerations: binary; uses Concise Binary Object Representation (CBOR) [RFC8949].

Security considerations: See Section 9 of this document.

Interoperability considerations: Implementations MUST follow the CBOR encoding rules of [RFC8949] and the CDDL definition of Section 5 of this document.

Published specification: This document.

Applications that use this media type: Agents, Operators, and Verifiers implementing Protocol 427.

Fragment identifier considerations: As specified for "+cose" in the corresponding structured-syntax suffix registration.

Additional information:

* Deprecated alias names for this type: N/A

* Magic number(s): (none, generic CBOR)

* File extension(s): .cbor

* Macintosh file type code(s): N/A

Person & email address to contact for further information: John Paul McGraw, Jr. <j.mcgraw@taskhawktech.com>

Intended usage: COMMON

Restrictions on usage: N/A

Author: John Paul McGraw, Jr.

Change controller: IETF

Provisional registration?: No.

8.6. "budget-required" Problem Type

IANA is requested to register the following entry in the "HTTP Problem Types" registry [IANA.HTTPProblemTypes] established by Section 6 of [RFC9457]:

Field	Value
Type URI	https://taskhawktech.com/problems/budget-required
Title	Budget attestation required
Recommended HTTP status code	427
Reference	Section 8.6 of this document

Table 6

8.7. Protocol-427 Reason Codes Registry

IANA is requested to establish a new registry titled "Protocol-427 Reason Codes" with registration policy "Specification Required" Section 4.6 of [RFC8126]. Each registration consists of a Reason token (printable ASCII, no whitespace), a Description, and a Reference.

The initial registry contents are the entries listed in Section 7.

8.8. CBOR Tag for Budget-Attestation

IANA is requested to assign one tag from the "First Come First Served" range of the "CBOR Tags" registry to identify the tagged form of the Budget-Attestation envelope (Section 5).

9. Security Considerations

This section follows the guidance of [RFC9205] and adapts the threat-class analysis of [RFC9421] to the Budget-Attestation envelope.

9.1. Replay Protection

Each Budget-Attestation carries a server-issued nonce (echoed from the WWW-Authenticate challenge), an iat and exp, and a request_binding. Verifiers MUST reject attestations whose nonce has previously been observed for the same Verifier and MUST reject attestations whose iat-exp interval does not include the current time.

Implementations are reminded of the early-data replay considerations of [RFC8470]: Verifiers SHOULD NOT accept Budget-Attestations carried in TLS 1.3 0-RTT data, or MUST employ the Early-Data header field mechanism of [RFC8470].

9.2. Signature Substitution and Multiple-Signature Confusion

The Budget-Attestation envelope can carry exactly one primary signature and at most one rail-keyed signature. Verifiers MUST require the primary signature for any acceptance decision; the rail-keyed signature is additive authority and MUST NOT substitute for the primary. Verifiers SHOULD reject envelopes containing multiple primary signatures.

9.3. Algorithm Downgrade

The COSE alg parameter in each Sig-Protected-Header is part of the signed input. Verifiers MUST compare the alg in the signed protected header against a configured policy and MUST NOT permit clients to "negotiate down" to a weaker algorithm via the WWW-Authenticate alg parameter.

9.4. Operator Key Compromise and Revocation

Operator keys SHOULD be rotated periodically. When a key is compromised or retired, Verifiers MUST be able to learn of the revocation through deployment-specific means (for example, an operator JWKS endpoint or signed revocation list); the revoked reason code is provided for diagnostic responses to the bearer of an attestation under a revoked key.

9.5. Post-Quantum Cryptographic Considerations

Per [I-D.ietf-cose-dilithium], the ML-DSA seed and the expanded private key require equal protection. Implementations SHOULD use constant-time, side-channel-resistant ML-DSA implementations. ML-DSA is non-deterministic; implementations SHOULD prefer the hedged randomization mode defined in [FIPS204], Section 3.

SLH-DSA is hash-based and offers an independent cryptographic foundation; this is the rationale for offering a rail-keyed signature option (Section 5.4). Implementations MUST use the pure mode of SLH-DSA per [I-D.ietf-cose-sphincs-plus]; HashSLH-DSA is not supported.

9.6. Bearer-Token Hygiene

Until validated, a Budget-Attestation has bearer-token characteristics (possession implies authority). TLS MUST be used for any HTTP exchange involving Authorization: Budget. Servers SHOULD scrub Authorization header values from request logs.

9.7. Verifier Responsibilities

Failure to verify is the dominant failure mode for any signature-based protocol. Verifiers MUST validate every check listed in Section 5 and MUST NOT short-circuit verification under any condition.

10. Privacy Considerations

This section follows the guidance of [RFC6973] and Section 6.1 of [RFC9205].

Budget-Attestations carry identifiers (iss, agent) and authority metadata (rails, amt) that, if leaked or correlated across Verifiers, can enable tracking of an Agent’s commercial activity. Mitigations include:

* short attestation lifetimes (RECOMMENDED exp - iat <= 900 seconds);

* TLS confidentiality for any HTTP message bearing an attestation or a 427 challenge;

* uniformly random nonce values, at least 128 bits long, that MUST NOT encode timestamps or sequence numbers;

* Operator-managed rotation of the agent identifier;

* restraint in Problem Details detail content (it MUST NOT contain the agent identifier or signature material);

* the option to omit the rail-keyed signature when not required by policy.

The nonce issued by a Verifier in a 427 challenge can itself be a tracking vector across multiple Agents observed by an on-path attacker. Verifiers SHOULD generate independent random nonces per challenge.

This document does not require Verifiers to retain any per-Agent state beyond what is necessary to detect nonce replay. Implementations SHOULD apply data minimization Section 6.1 of [RFC6973] when constructing the agent identifier and when logging verification outcomes.

11. References

12. References

12.1. Normative References

[FIPS204] National Institute of Standards and Technology (NIST), "Module-Lattice-Based Digital Signature Standard", FIPS PUB 204, DOI 10.6028/NIST.FIPS.204, August 2024, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf>.

[FIPS205] National Institute of Standards and Technology (NIST), "Stateless Hash-Based Digital Signature Standard", FIPS PUB 205, DOI 10.6028/NIST.FIPS.205, August 2024, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf>.

[I-D.ietf-cose-dilithium]
Prorock, M. and O. Steele, "ML-DSA for JOSE and COSE", Work in Progress, Internet-Draft, draft-ietf-cose-dilithium-11, 15 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-dilithium-11>.

[I-D.ietf-cose-sphincs-plus]
Prorock, M., Steele, O., and H. Tschofenig, "SLH-DSA for JOSE and COSE", Work in Progress, Internet-Draft, draft-ietf-cose-sphincs-plus-07, 15 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-sphincs-plus-07>.

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

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/rfc/rfc6838>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

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

[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/rfc/rfc8610>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/rfc/rfc9052>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

[RFC9111] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Caching", STD 98, RFC 9111, DOI 10.17487/RFC9111, June 2022, <https://www.rfc-editor.org/rfc/rfc9111>.

[RFC9205] Nottingham, M., "Building Protocols with HTTP", BCP 56, RFC 9205, DOI 10.17487/RFC9205, June 2022, <https://www.rfc-editor.org/rfc/rfc9205>.

[RFC9457] Nottingham, M., Wilde, E., and S. Dalal, "Problem Details for HTTP APIs", RFC 9457, DOI 10.17487/RFC9457, July 2023, <https://www.rfc-editor.org/rfc/rfc9457>.

[RFC9651] Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", RFC 9651, DOI 10.17487/RFC9651, September 2024, <https://www.rfc-editor.org/rfc/rfc9651>.

12.2. Informative References

[L402] Lightning Labs, "L402 Protocol Specification", 2024, <https://github.com/lightninglabs/L402>.

[RFC6585] Nottingham, M. and R. Fielding, "Additional HTTP Status Codes", RFC 6585, DOI 10.17487/RFC6585, April 2012, <https://www.rfc-editor.org/rfc/rfc6585>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/rfc/rfc6973>.

[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/rfc/rfc7942>.

[RFC8470] Thomson, M., Nottingham, M., and W. Tarreau, "Using Early Data in HTTP", RFC 8470, DOI 10.17487/RFC8470, September 2018, <https://www.rfc-editor.org/rfc/rfc8470>.

[RFC9421] Backman, A., Ed., Richer, J., Ed., and M. Sporny, "HTTP Message Signatures", RFC 9421, DOI 10.17487/RFC9421, February 2024, <https://www.rfc-editor.org/rfc/rfc9421>.

[RFC9449] Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., and D. Waite, "OAuth 2.0 Demonstrating Proof of Possession (DPoP)", RFC 9449, DOI 10.17487/RFC9449, September 2023, <https://www.rfc-editor.org/rfc/rfc9449>.

[X402] Coinbase, Inc., "x402: An Open Standard for Internet-Native Payments", 2025, <https://www.x402.org/x402-whitepaper.pdf>.

Appendix A. CDDL Schema (Collected)

The following CDDL block consolidates the schema of Section 5 for ease of implementation.

```
Budget-Attestation = #6.TBD(Budget-Attestation-Untagged) /
                     Budget-Attestation-Untagged

Budget-Attestation-Untagged = [
  protected   : bstr .cbor Protected-Header,
  unprotected : Unprotected-Header,
  claims      : bstr .cbor Budget-Claims,
  signatures  : [+ Signature]
]

Protected-Header = {
  1 => int,
  ? 4 => bstr
}

Unprotected-Header = {
  * (int / tstr) => any
}

Budget-Claims = {
  "version"  => uint,
  "iss"      => tstr,
  "agent"    => tstr,
  "iat"      => uint,
  "exp"      => uint,
  "nonce"    => bstr .size (16..64),
  "rb"       => Request-Binding,
  "rails"    => [+ tstr],
  ? "amt"    => Rail-Amount-Map
}
```

```
Request-Binding = {
  "method"  => tstr,
  "uri-h"   => bstr .size 32,
  ? "body-h" => bstr .size 32
}

Rail-Amount-Map = {
  + tstr => uint
}

Signature = [
  protected   : bstr .cbor Sig-Protected-Header,
  unprotected : Unprotected-Header,
  signature   : bstr
]

Sig-Protected-Header = {
  1 => int,
  ? 4 => bstr,
  ? "role" => "operator" / "rail"
}
```

Appendix B. Example

The following example shows a complete 427 challenge, the Agent’s follow-up request body containing a Budget-Attestation envelope (in CBOR diagnostic notation per Section 8 of [RFC8949]), and the Verifier’s response.

Initial response:

```
HTTP/1.1 427 Budget Required
Date: Tue, 05 May 2026 14:00:00 GMT
Cache-Control: no-store
Content-Type: application/problem+json
Protocol-427-Version: 1
WWW-Authenticate: Budget realm="api.example",
                  alg="ML-DSA-87",
                  rails="x402 l402 mpp",
                  nonce="QMjVqg5Xb6yV0bO_t9X8gQ",
                  max-age=900

{ "type": "https://taskhawktech.com/problems/budget-required",
  "title": "Budget attestation required",
  "status": 427,
  "min-budget": {"USD": 250},
  "accepted-rails": ["x402","l402","mpp"],
  "max-age": 900,
  "protocol-version": 1 }
```

Follow-up request (CBOR shown in diagnostic notation):

```
POST /research/papers/12345 HTTP/1.1
Host: api.example
Content-Type: application/budget-attestation+cose
Authorization: Budget binding="lQ7M...base64url-sha256..."

[ << { 1: -50 } >>,
  {},
  << { "version": 1,
       "iss":     "https://op.example/operators/42",
       "agent":   "agent-7c2e",
       "iat":     1746453600,
       "exp":     1746454500,
       "nonce":   h'40c8d5aa0e576fac95d1b3bfb7d5fc81',
       "rb":      { "method": "POST",
                    "uri-h":  h'a3f1...',
                    "body-h": h'0000...' },
       "rails":   ["x402","l402","mpp"],
       "amt":     { "USD": 250 } } >>,
  [ [ << { 1: -50, "role": "operator" } >>,
      {},
      h'<ML-DSA-87 signature, 4627 bytes>' ] ] ]
```

Successful response: as for the underlying request (200 OK, etc.).

Appendix C. Implementation Status

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

This section records known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, in accordance with the guidelines of [RFC7942].

The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Listing of any individual implementation does not imply endorsement by the IETF. No effort has been spent to verify the information presented here. Other implementations may exist.

C.1. Kevros (TaskHawk Systems)

Organization: TaskHawk Systems LLC

Name: Kevros

Description: An implementation of Protocol 427, including issuance and verification of Budget-Attestation envelopes signed with ML-DSA-87.

Coverage: Section 3 through Section 8.7 of this document; settlement-rail bindings for x402, L402, and Lightning multi-path payments.

Implementation experience: The host enforcement kernel within which Protocol 427 issuance and verification logic executes is formally specified in TLA+ and Lean 4. 12 safety invariants and 4 liveness properties of the kernel have been machine-checked: TLC bounded model checking explored 1.94 billion states with zero violations; Lean 4 interactive proof verifies 20 theorems with zero "sorry". Protocol 427 endpoints run inside this verified kernel and inherit its fail-closed and chain-integrity guarantees. A separate state-machine specification of Protocol 427 version negotiation and Budget-Attestation signature verification has not yet been written.

Contact: j.mcgraw@taskhawktech.com

Appendix D. Acknowledgments

The author thanks the Working Group for early feedback on this document.

Author’s Address

John Paul McGraw, Jr.
TaskHawk Systems LLC
Charlottesville, VA
United States of America
Email: j.mcgraw@taskhawktech.com