Secure Telephone Identity Revisited
Internet-Draft
Intended status: Standards Track
Expires: 3 October 2026

C. Wendt
R. ナ嗟iwa
Somos Inc.

1 April 2026

VESPER Out-of-Band OOB
draft-wendt-stir-vesper-oob-02

Abstract

This document describes a mechanism for delivering authenticated telephone call identity information using the VESPER framework in environments where SIP signaling is unavailable or unsuitable. By supporting an out-of-band (OOB) transport model, this approach enables entities to publish and retrieve signed PASSporT assertions independent of end-to-end delivery within SIP-based VoIP networks. These PASSporTs are signed with delegate certificates that were authorized for issuance by corresponding authority tokens, which represent the trust and validation of telephone number control and related claim information. Transparency features ensure that these authorizations are publicly auditable and cryptographically provable, supporting a higher standard of trust. This document also introduces support for Connected Identity to the STIR OOB model, enabling the called party to respond with a signed PASSporT asserting its identity, thereby binding the identities of both parties to the transaction and enhancing end-to-end accountability. The OOB mechanism serves as an alternative delivery path for PASSporTs in cases where end-to-end in-band SIP delivery is not possible, enabling verifiers to confirm the association between the originating telephone number and the identity asserting authority as part of the broader VESPER trust framework.

About This Document

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

The latest revision of this draft can be found at https://github.com/appliedbits/draft-wendt-stir-vesper-oob. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-wendt-stir-vesper-oob/.

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

Source for this draft and an issue tracker can be found at https://github.com/appliedbits/draft-wendt-stir-vesper-oob.

Status of This Memo

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

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

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

This Internet-Draft will expire on 3 October 2026.

Copyright Notice

Table of Contents

1. Introduction

The STIR framework enables the signing and verification of telephone calls using PASSporT [RFC8225] objects carried in SIP [RFC3261] in Identity Header Fields defined in [RFC8224]. However, there are scenarios where SIP-based in-band transmission is not feasible or the Identity Header Field may not be supported, such as legacy TDM interconnects or where intermediary network elements strip SIP Identity headers. STIR Out-of-Band (OOB) [RFC8816] addresses this generally for STIR by defining an OOB delivery model.

The VESPER framework [I-D.wendt-stir-vesper] extends the STIR architecture by introducing delegate certificates issued under authority tokens and recorded in certificate transparency logs, strengthening the association between telephone number assignments and the entities authorized to use them in signed PASSporTs.

This document defines how the VESPER framework extends to an out-of-band delivery mechanism corresponding to the model described in [RFC8816]. It enables authorized delegate certificate holders to

deliver PASSporTs over a non-SIP-based path for retrieval and validation by a STIR Verification Service, maintaining continuity of trust across heterogeneous networks.

The VESPER OOB delivery model is based on a publish-and-retrieve interface using an open discovery model for Call Placement Services (CPS). This document extends the concepts in [RFC8816] to specifically define an HTTPS-based interface for publishing and retrieving VESPER PASSporTs. It utilizes the following:

* A mechanism for announcing the associated OOB Call Placement Services (CPSs) using the CPS URI extension defined in [I-D.sliwa-stir-cert-cps-ext].

* A discovery mechanism for OOB endpoints using STI certificate transparency log monitoring, where CPS URIs embedded in delegate certificates become publicly discoverable when those certificates are logged.

It also optionally supports Connected Identity [I-D.ietf-stir-rfc4916-update], enabling both parties to authenticate their telephone numbers and establish end-to-end identity assurance, as also adopted by VESPER [I-D.wendt-stir-vesper].

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.

VESPER: Verifiable STI Presentation and Evidence for RTU [I-D.wendt-stir-vesper].

PASSporT: Personal Assertion Token as defined in [RFC8225].

Delegate Certificate: A certificate issued to an entity asserting right-to-use for a telephone number, based on an authority token, as defined in [RFC9060] and profiled in [I-D.wendt-stir-vesper].

Authority Token: A signed assertion that authorizes the issuance of a delegate certificate and represents the authorization of a subject’s right-to-use a telephone number and any associated claims, defined in [RFC9447].

CPS URI: Call Placement Service (CPS) URI extension in X.509 certs [I-D.sliwa-stir-cert-cps-ext].

CPS Discovery: The process of identifying the CPS endpoint responsible for a given TN or SPC by monitoring STI-CT logs for delegate certificates containing the CPS URI extension defined in [I-D.sliwa-stir-cert-cps-ext].

3. VESPER OOB Architectural Overview

The VESPER OOB architecture enables the out-of-band signing, publishing, discovery, and verification of PASSporTs using a trust framework based on delegate certificates and transparency mechanisms. These components interact across SIP and HTTPS protocols to support parallel in-band and out-of-band delivery of telephone number authentication information. Figure 1 illustrates the flow of identity data between the authentication service, the out-of-band Call Placement Service (CPS), and the verification service.

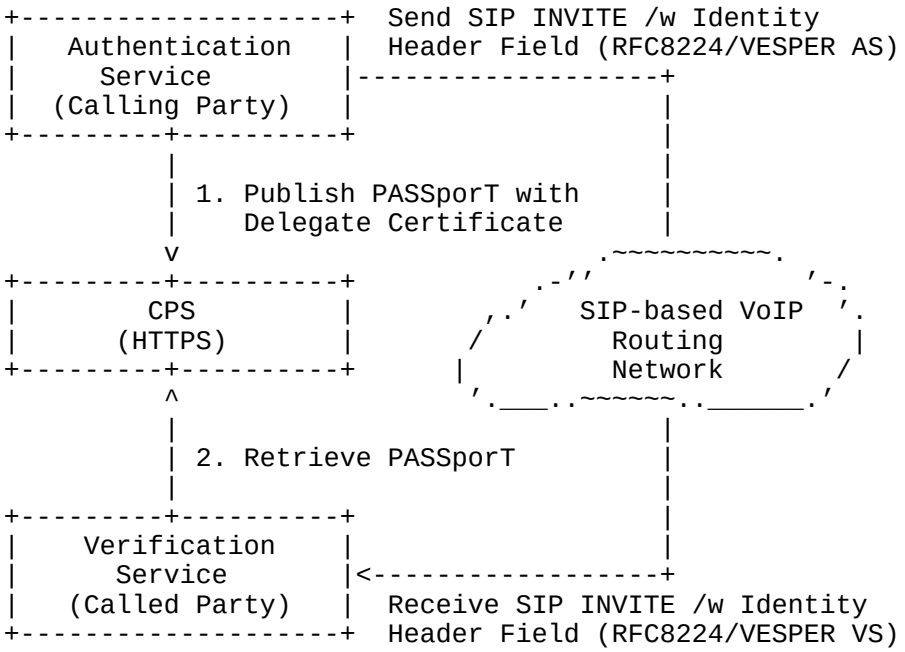

Figure 1 - Architecture showing both in-band and out-of-band PASSporT delivery

4. HTTPS Interface Specification

The interface design is conceptually aligned with the interface model described in [ATIS-1000105] Section 7. It supports two categories of HTTPS methods:

General Operations:

These required endpoints enable basic VESPER-OOB publish and retrieval functions:

* GET /health - check service availability

* POST /passports/{DEST}/{ORIG} - publish one or more signed PASSporTs, optionally with a ’response_uuid’ for Connected Identity

* GET /passports/{DEST}/{ORIG} - retrieve published PASSporTs and optionally discover an associated ’response_uuid’

Connected Identity Extensions:

These optional endpoints are used if a response_uuid was included in the publish operation and the recipient supports Connected Identity:

* POST /respond/{UUID} - the called party submits a ’rsp’ PASSporT

* GET /passports/response/{UUID} - the caller polls for the response

* GET /passports/response/stream/{UUID} - Server-Sent Events (SSE) push interface (optional)

* wss://.../stream/respond/{UUID} - WebSocket push delivery (optional)

All endpoints MUST be served over HTTPS. All endpoints that expose PASSporTs, Connected Identity UUIDs, or response PASSporTs MUST require authentication via Access JWT as defined in Section 4.1. The GET /health endpoint does not require authentication. CPS operators SHOULD additionally enforce rate-limits and access-control policies.

Server certificates SHOULD be validated using standard PKIX procedures. HTTP Strict Transport Security (HSTS) MAY be used by CPS operators to enforce HTTPS usage.

4.1. Common Access JWT

All CPS interfaces that require authorization MUST support Access JWTs signed using the ES256 algorithm and validated against trusted VESPER delegate certificates. These tokens establish caller or responder identity and intent.

4.1.1. Access JWT Header

```
{
  "alg": "ES256",
  "x5c": [
    "MIIB3TCCAYOgAwIBAgIUUjF7Jq9kYfU12nJkBA==",
    "IUUjF7Jq9kYfU12nJkBAMIIB3TCCAYOgAwIBAg=="
  ]
}
```

* ’alg’: MUST be "ES256" as required by STIR PASSporT and VESPER.

* ’x5c’: An array of base64-encoded certificates representing the end-entity delegate certificate and any intermediate certificates with an optionally included root certificate. These MUST be validated against the trust anchors defined in the certificate policy defined in [RFC8226].

4.1.2. Access JWT Claims

The Access JWT payload MUST contain the following claims:

Claim	Description
'iat'	Issued-at timestamp (Unix time). MUST be recent (< 5 min skew).
'exp'	Expiration timestamp for the token.
'jti'	Unique token ID for replay prevention and audit.
'action'	Operation intent: "publish", "retrieve", or "respond".
'aud'	CPS hostname. MUST match the target server.
'iss'	SPC or TN of the signer. MUST match TNAuthList in cert.
'sub'	SPC or TN of the subscriber on whose behalf the action is taken.
'orig'	Object with TN/URI of the originating party.
'dest'	Object with TN/URI of the destination party.
'passports'	OPTIONAL. See below for digest definition.
'rsp_passport'	OPTIONAL. See below for digest definition.

Table 1: Access JWT Claims

The 'passports' claim, when present for a "publish" action, MUST contain the base64url-encoded SHA-256 hash of the JCS [RFC8785] canonicalization of the complete JSON request body object (i.e., the object containing the passports array). The 'rsp_passport' claim, when present for a "respond" action, MUST contain the base64url-encoded SHA-256 hash of the JCS [RFC8785] canonicalization of the complete JSON request body object (i.e., the object containing the rsp_passport field). Hash values MUST use base64url encoding without padding as defined in RFC 4648 Section 5.

Note: The TNAuthList in a VESPER delegate certificate may contain TN entries, SPC entries, or both. In the common case where an entity signs for its own telephone numbers, iss and sub will be the same value and correspond to a TN or SPC in the TNAuthList, as the signing entity and the telephone number holder are the same party. In platform or delegated deployments, iss may identify an SPC-authorized signing entity while sub identifies the subscriber's TN, both of which are covered by the certificate's TNAuthList.

4.1.2.1. Examples

Publish Token (Calling Party):

```
{
  "iat": 1693590000,
  "exp": 1608048425,
  "jti": "550e8400-e29b-41d4-a716-446655440000",
  "action": "publish",
  "aud": "cps.example.net",
  "iss": "12013776051",
  "sub": "12013776051",
  "orig": { "tn": "12013776051" },
  "dest": { "tn": ["19032469103"] },
  "passports": "sha256-XyZabc123..."
}
```

Retrieve Token (Verifying Called Party):

```
{
  "iat": 1693590100,
  "exp": 1693590400,
  "jti": "550e8400-e29b-41d4-a716-426655440002",
  "action": "retrieve",
  "aud": "cps.example.net",
  "iss": "19032469103",
  "sub": "19032469103",
  "orig": { "tn": "12013776051" },
  "dest": { "tn": ["19032469103"] }
}
```

Respond Token (Called Party responding with Connected Identity):

```
{
  "iat": 1693590050,
  "exp": 1693590400,
  "jti": "550e8400-e29b-41d4-a716-426655440001",
  "action": "respond",
  "aud": "cps.example.net",
  "iss": "19032469103",
  "sub": "19032469103",
  "orig": { "tn": "12013776051" },
  "dest": { "tn": ["19032469103"] },
  "rsp_passport": "sha256-AbCdEf123..."
}
```

4.1.3. Validation Rules

The CPS MUST validate the Access JWT as follows:

* Signature: Must be signed with ES256 using a VESPER delegate certificate that chains to a trusted STI root.

* Certificate: The certificate in 'x5c' MUST match the 'iss'/'sub' TN and contain valid TNAuthList entries.

* Time Validity: 'iat' MUST be recent (within an allowed freshness window, e.g., 5 minutes).

* Audience: 'aud' MUST match the target CPS domain.

* Claims Match: The 'orig' and 'dest' claims MUST match the HTTP path parameters.

* Digest Integrity: If the 'passports' or 'rsp_passport' claim is present, its value MUST equal the base64url-encoded SHA-256 hash computed over the JCS [RFC8785] canonicalization of the complete JSON request body. Both parties MUST canonicalize the full request body object before hashing. The encoding of the hash value MUST use base64url without padding as defined in RFC 4648 Section 5.

4.1.4. Additional Security

* CPS SHOULD reject expired, reused, or improperly scoped JWTs.

* JWT replay prevention SHOULD be enforced using the jti field and short TTLs. The CPS MUST cache recent jti values and MUST reject re-use within the configured window.

* Tokens MUST be scoped per transaction; long-lived JWTs MUST NOT be used.

4.2. API Method Definitions

4.2.1. Method: 'GET /health'

4.2.1.1. Request Definition

Method: GET
Path: /health
Authentication: None required

4.2.1.2. Response Definition

200 OK - Service operational 503 Service Unavailable - Service not operational Body (optional):

```
{
  "status": 200,
  "message": "OK"
}
```

4.2.2. Publish Method: POST /passports/{DEST}/{ORIG}

This method allows the calling party to publish one or more signed PASSporTs associated with a specific ORIG and DEST pair. The CPS MAY optionally return a response_uuid for Connected Identity.

PASSporTs and Connected Identity response PASSporTs SHOULD be retained only for a short period of time unless longer retention is explicitly required by policy.

Note: [ATIS-1000105] defines a "re-publish" action for forwarding PASSporTs between CPSs. Because VESPER OOB uses a transparent discovery model based on STI-CT log monitoring rather than bilateral CPS-to-CPS communication, re-publishing is not required. CPS implementations conforming to this specification are not required to support the re-publish action or the associated "token" fields defined in [ATIS-1000105]. However, this specification is designed to be compatible with deployments that support both VESPER OOB and [ATIS-1000105].

4.2.2.1. Request Definition

Method: POST
Path: /passports/{DEST}/{ORIG}
Authentication: Access JWT with "action": "publish"

4.2.2.2. Request Headers

```
Content-Type: application/json
Authorization: Bearer <Access JWT>
```

The server SHOULD support an Idempotency-Key request header [I-D.ietf-httpapi-idempotency-key-header]. When present, repeated requests with the same key MUST return the original result without creating duplicate records.

4.2.2.3. Request Parameters

DEST: Canonicalized and percent-encoded destination telephone number or URI. ORIG: Canonicalized and percent-encoded originating telephone number or URI.

Canonicalization of TNs follows [RFC8224] and percent encoding of URIs follows [RFC3986].

Note: The path ordering places {DEST} before {ORIG} to align with the lookup pattern used by the called party, which typically knows its own number (DEST) and resolves PASSporTs based on the calling party (ORIG). The CPS MUST validate that the orig and dest claims in the Access JWT match the {ORIG} and {DEST} path parameters respectively; a mismatch MUST result in a 403 Forbidden response.

4.2.2.4. Request Body

The request body is a JSON object with the following field:

* passports: REQUIRED. An array of one or more PASSporT strings signed by the calling party. Multiple PASSporTs MAY be included when the authentication service issues PASSporTs with different ppt types (e.g., a base shaken PASSporT alongside a div or rcd PASSporT) for the same call. All PASSporTs in the array MUST share the same orig, dest, and iat values and MUST be signed by the same delegate certificate.

Authorization JWT Requirements:

The Access JWT for this method MUST include:

* "action": "publish"

All other validation requirements are defined in Common Access JWT.

4.2.2.5. Example Request

```
POST /passports/19032469103/12013776051 HTTP/1.1
Host: cps.example.com
Authorization: Bearer <Access JWT>
Content-Type: application/json

{
  "passports": [
    "eyJhbGciOiJFUzI1NiIsIn..."
  ]
}
```

4.2.2.6. Response Definition

Success Codes

201 - Created if the PASSporTs were successfully published.

Failure Codes

400 - Bad Request if required fields are missing or malformed
401 - Unauthorized if authentication fails
403 - Forbidden if certificate constraints are not met
429 - Too Many Requests if rate-limited
5xx errors (e.g., 503 Service Unavailable)

Responses MUST use status codes defined in [RFC6585] and SHOULD be informative when possible.

If the server supports Connected Identity, the response body MAY include a response_uuid that the called party can use in follow-up Connected Identity methods. This UUID [RFC4122] is generated by the CPS and serves as a transaction-specific identifier for subsequent API calls.

4.2.2.7. Example Response

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "status": 201,
  "message": "Created",
  "response_uuid": "123e4567-e89b-12d3-a456-426614174000"
}
```

4.2.2.8. Response Body Fields

* status: HTTP status code indicating result of publish request (e.g., 201 for success).

* message: A human-readable message describing the outcome of the request.

* response_uuid: (Optional) A UUID [RFC4122] generated by the CPS for Connected Identity. Returned only if the CPS supports Connected Identity response workflows.

4.2.2.9. Example Success and Error Responses

Success Response (201 Created):

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "status": 201,
  "message": "Created",
  "response_uuid": "123e4567-e89b-12d3-a456-426614174000"
}
```

Error Response (400 Bad Request):

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "status": 400,
  "error": "Missing required field: passports"
}
```

Error Response (401 Unauthorized):

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json

{
  "status": 401,
  "error": "Access JWT is invalid or expired"
}
```

4.2.3. Retrieve Method: GET /passports/{DEST}/{ORIG}

This method allows the called party to retrieve PASSporTs published by the originating party for a given ORIG/DEST combination.

4.2.3.1. Request Definition

Method: GET
Path: /passports/{DEST}/{ORIG}
Authentication: Access JWT with "action": "retrieve"

4.2.3.2. Request Headers

Authorization: Bearer <Access JWT>

4.2.3.3. Request Parameters

* DEST: Percent-encoded and canonicalized destination telephone number or URI, representing the final called party after any retargeting.

* ORIG: Percent-encoded and canonicalized calling party TN or URI, typically from the SIP From or P-Asserted-Identity header.

Canonicalization of TNs follows [RFC8224] and percent encoding of URIs follows [RFC3986].

Note: The path ordering places {DEST} before {ORIG} to align with the lookup pattern used by the called party, which typically knows its own number (DEST) and resolves PASSporTs based on the calling party (ORIG). The CPS MUST validate that the orig and dest claims in the Access JWT match the {ORIG} and {DEST} path parameters respectively; a mismatch MUST result in a 403 Forbidden response.

4.2.3.4. Authorization JWT Requirements

The JWT used to authorize this request MUST include:

* "action": "retrieve"

All other JWT validation requirements are defined in Section 4.1 and MUST also be enforced by the CPS.

4.2.3.5. Prerequisite Check

Before accepting a Connected Identity response, the CPS SHOULD verify that the PASSporT associated with the given response_uuid was previously retrieved by a party whose iss claim matches the dest TN of the original transaction. This ensures that the responding party has had the opportunity to validate the originating PASSporT before asserting its own identity. If the CPS enforces this check and no prior retrieval has occurred, it SHOULD return 409 Conflict with a descriptive error indicating that retrieval must precede response submission.

4.2.3.6. Response Definition

Success:

200 OK - PASSporT(s) retrieved successfully

Failure:

401 Unauthorized - JWT missing or invalid
403 Forbidden - Certificate constraints violated
404 Not Found - No PASSporTs available
429 Too Many Requests - Rate limits exceeded
503 Service Unavailable - CPS temporarily unavailable

Status codes MUST follow [RFC6585]. On 5xx failures, retrying another CPS endpoint MAY be allowed.

Response Body (on success):

```
{
  "passports": [
    "eyJhbGciOiJFUzI1NiIsIn..."
  ],
  "response_uuid": "123e4567-e89b-12d3-a456-426614174000"
}
```

* passports: An array of one or more PASSporT strings published by the originating party, in compact JWS serialization format as per [RFC8225].

* response_uuid: OPTIONAL. If present, provides the Connected Identity transaction UUID [RFC4122] to which the called party can submit an identity response PASSporT using the appropriate API method. This value is provided only if included in the corresponding publish operation.

4.2.3.7. Example Request

```
GET /passports/19032469103/12013776051 HTTP/1.1
Host: cps.example.com
Authorization: Bearer <Access JWT>
```

4.2.3.8. Example Response

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

{
  "passports": [
    "eyJhbGciOiJFUzI1NiIsIn..."
  ],
  "response_uuid": "123e4567-e89b-12d3-a456-426614174000"
}
```

4.2.3.9. Example Success and Error Responses

Success Response (200 OK):

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

{
  "passports": [
    "eyJhbGciOiJFUzI1NiIsIn..."
  ],
  "response_uuid": "123e4567-e89b-12d3-a456-426614174000"
}
```

Error Response (404 Not Found):

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
  "status": 404,
  "error": "No PASSporTs available for the requested origin and
    destination"
}
```

Error Response (403 Forbidden):

```
HTTP/1.1 403 Forbidden
Content-Type: application/json

{
  "status": 403,
  "error": "Caller is not authorized to retrieve PASSporTs for
    this identity"
}
```

4.2.3.10. Response Body Fields

* passports: Array of PASSporT strings published by the originating party, encoded in compact JWS serialization.

* response_uuid: (Optional) UUID [RFC4122] that identifies a Connected Identity response transaction. Provided only if the CPS returned it during publish.

4.2.4. Respond Method: POST /respond/{UUID}

This method allows the called party to submit a response PASSporT (rsp_passport) asserting their identity in a Connected Identity exchange. The UUID [RFC4122] corresponds to the response_uuid originally returned by the CPS during the publish operation.

4.2.4.1. Request Definition

```
Method: POST
Path: /respond/{UUID}
Authentication: Access JWT with "action": "respond"
```

4.2.4.2. Request Headers

```
Content-Type: application/json
Authorization: Bearer <Access JWT>
```

4.2.4.3. Request Parameters

* UUID: A unique response transaction identifier [RFC4122] returned by the CPS in the publish response as response_uuid. This identifies the call session context for Connected Identity.

4.2.4.4. Request Body

```
{
  "rsp_passport": "eyJhbGciOiJFUzI1NiIsIn..."
}
```

* rsp_passport: REQUIRED. The PASSporT signed by the called party delegate certificate for Connected Identity.

4.2.4.5. Authorization JWT Requirements

The JWT used to authorize this request MUST include:

* "action": "respond"

All other JWT validation requirements are defined in Section 4.1 and MUST be enforced by the CPS.

4.2.4.6. Response Definition

Success:

201 Created - The Connected Identity response was accepted.

Failure:

401 Unauthorized - JWT missing or invalid.
403 Forbidden - Certificate constraints violated.
404 Not Found - UUID not found or expired.
409 Conflict - A response has already been submitted.
429 Too Many Requests - Rate limits exceeded.
503 Service Unavailable - CPS temporarily unavailable.

Status codes MUST follow [RFC6585]. Connected Identity response PASSporTs SHOULD be retained only for a short period unless longer retention is explicitly required by policy.

4.2.4.7. Example Request

```
POST /respond/123e4567-e89b-12d3-a456-426614174000 HTTP/1.1
Host: cps.example.net
Content-Type: application/json
Authorization: Bearer <Access JWT>

{
  "rsp_passport": "eyJhbGciOiJFUzI1NiIsIn..."
}
```

4.2.4.8. Example Response

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "status": 201,
  "message": "Connected Identity Stored"
}
```

4.2.4.9. Example Success and Error Responses

Success Response (201 Created):

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "status": 201,
  "message": "Connected Identity Stored"
}
```

Error Response (409 Conflict):

```
HTTP/1.1 409 Conflict
Content-Type: application/json

{
  "status": 409,
  "error": "A response for this UUID has already been submitted"
}
```

Error Response (404 Not Found):

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
  "status": 404,
  "error": "UUID not found or expired"
}
```

4.2.5. Retrieving Connected Identity Responses

Once a response is submitted using the response_uuid, the originating party may retrieve it in two ways using a polling interface (GET method) or via an optional push interface using WSS as detailed in the following methods.

4.2.6. Retrieve Response Method: GET /passports/response/{UUID}

This method allows the originating (calling) party to retrieve a Connected Identity response PASSporT, if one has been submitted by the called party. The UUID in this path is the same value (response_uuid) previously provided by the CPS in the response to the POST /passports/{DEST}/{ORIG} method.

4.2.6.1. Request Definition

Method: GET
Path: /passports/response/{UUID}
Headers: Authorization: Bearer <JWT>

4.2.6.2. Response Definition

Success:

200 OK - Connected Identity response PASSporT retrieved successfully

Failure:

404 Not Found - No response is available yet

4.2.6.3. Response Body

```
{
  "rsp": {
    "passport": "eyJhbGciOiJFUzI1NiIsIn..."
  }
}
```

4.2.6.4. Response Body Fields

* rsp: An object containing the Connected Identity response.
  - passport: A PASSporT string signed by the called party using its delegate certificate.

4.2.6.5. Example Success and Error Responses

Success Response (200 OK):

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

{
  "rsp": {
    "passport": "eyJhbGciOiJFUzI1NiIsIn..."
  }
}
```

Error Response (404 Not Found):

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
  "status": 404,
  "error": "No Connected Identity response has been submitted for
    this UUID"
}
```

4.2.7. Retrieve Response Push Methods (Optional)

The CPS MAY support real-time delivery of Connected Identity responses via push interfaces as an alternative to polling.

4.2.7.1. Server-Sent Events (SSE)

```
GET /passports/response/stream/{UUID}
Accept: text/event-stream
Authorization: Bearer <Access JWT>
```

The SSE endpoint uses the same Access JWT authentication as the polling GET endpoint. The JWT MUST include "action": "retrieve" and the iss claim MUST match the originating party of the transaction. The CPS MUST validate the JWT before initiating the event stream.

4.2.7.2. WebSocket

```
wss://cps.example.net/stream/respond/{UUID}
```

Because the WebSocket upgrade request does not support the Authorization header in all client implementations, the Access JWT MUST be conveyed using one of the following mechanisms, listed in order of preference:

1. The Sec-WebSocket-Protocol subprotocol negotiation, using the format access_token.<JWT>.

2. A query parameter ?token=<JWT> on the connection URI. When this method is used, CPS operators MUST ensure the token is not logged in access logs.

3. An initial text frame sent immediately after connection establishment, containing the Access JWT. The CPS MUST NOT transmit any response data until the JWT has been received and validated.

The CPS MUST close the WebSocket connection with status code 1008 (Policy Violation) if the JWT is missing, invalid, or unauthorized.

Both push interfaces MUST enforce the same authorization constraints as the polling GET endpoint: only the authenticated originating party of the transaction (as identified by iss) is permitted to receive the response.

5. Example VESPER OOB Request/Response Flow

This example illustrates a full transaction using the Connected Identity UUID-based pattern.

5.1. Calling Party Publishes a PASSporT

```
POST /passports/19035551234/12015550100 HTTP/1.1
Host: cps.example.net
Content-Type: application/json
Authorization: Bearer <jwt-from-calling-party>

Body:

{
  "passports": [
    "eyJhbGciOiJFUzI1NiIsIn..."
  ]
}

Response:

HTTP/1.1 201 Created
Content-Type: application/json

{
  "status": 201,
  "message": "Created",
  "response_uuid": "123e4567-e89b-12d3-a456-426614174000"
}
```

5.2. Called Party Retrieves PASSporT and Extracts response_uuid

```
GET /passports/19035551234/12015550100 HTTP/1.1
Host: cps.example.net
Authorization: Bearer <jwt-from-called-party>

Response:

{
  "passports": [
    "eyJhbGciOiJFUzI1NiIsIn..."
  ],
  "response_uuid": "123e4567-e89b-12d3-a456-426614174000"
}
```

5.3. Called Party Submits a Connected Identity rsp PASSporT

```
POST /respond/123e4567-e89b-12d3-a456-426614174000 HTTP/1.1
Host: cps.example.net
Content-Type: application/json
Authorization: Bearer <jwt-from-called-party>

Body:

{
  "rsp_passport": "eyJhbGciOiJFUzI1NiIsIn..."
}

Response:

HTTP/1.1 201 Created
Content-Type: application/json

{"status":201,"message":"Connected Identity Stored"}
```

5.4. Calling Party Polls for the rsp PASSporT

```
GET /passports/response/123e4567-e89b-12d3-a456-426614174000 HTTP/1.1
Host: cps.example.net
Authorization: Bearer <jwt-from-calling-party>

Response:

{
  "rsp": {
    "passport": "eyJhbGciOiJFUzI1NiIsIn..."
  }
}
```

This flow demonstrates the full cycle from publish to response using the Connected Identity UUID-based model. Optionally, the final step may use SSE or WSS push interfaces instead of polling.

The VESPER OOB interface specification offers a modular architecture for telephony identity authentication. It supports both simple publish/retrieve workflows and bidirectional identity binding through Connected Identity.

6. Authentication Service Procedures for VESPER OOB

When participating in VESPER OOB, Authentication Services that sign PASSporTs MUST adhere to all requirements of the core VESPER specification [I-D.wendt-stir-vesper] and additional procedures specified herein to ensure the integrity of out-of-band transactions and compatibility with verifier expectations.

6.1. Delegate Certificate Requirements

Delegate certificates used to sign PASSporTs in VESPER OOB MUST be issued under authority tokens that represent an explicit right-to-use a telephone number. These certificates MUST include:

* One or more Signed Certificate Timestamps (SCTs) from certificate transparency logs as defined in [I-D.ietf-stir-certificate-transparency].

* A CPS URI in the Call Placement Service (CPS) X.509 extension, enabling discovery of the associated OOB Call Placement Service (CPS) as defined in [I-D.sliwa-stir-cert-cps-ext].

6.2. PASSporT Construction Requirements

PASSporTs signed in a VESPER OOB deployment MUST meet the following conditions:

* The PASSporT MUST be signed with a delegate certificate whose authority token authorizes the use of the specific originating telephone number.

* The ’orig’ claim MUST contain the telephone number or URI as authorized by the delegate certificate.

* The ’dest’ claim MUST reflect the final destination of the call after any retargeting.

* The ’iat’ claim MUST represent a timestamp within an acceptable freshness window (e.g., 5 minutes).

* The JWT ’x5c’ header MUST contain the certificate chain including the delegate certificate and its SCT(s).

* The JWT ’x5u’ header MUST contain the HTTPS URL of the delegate certificate at its location in the domain-controlled repository, and the domain in that URL MUST match the dNSName SubjectAltName of the signing certificate.

The Authentication Service MUST also publish the signed PASSporT to the CPS endpoint identified by the CPS URI in the delegate certificate.

7. CPS URI and OOB CPS Discovery

CPS URIs are associated with VESPER delegate certificates through the CPS URI extension defined in [I-D.sliwa-stir-cert-cps-ext]. This extension embeds an HTTPS URI identifying the CPS endpoint responsible for publishing and serving PASSporTs for the telephone numbers and SPCs covered by the certificate’s TNAuthList.

When a VESPER delegate certificate containing a CPS URI extension is submitted to a STI-CT log, the CPS URI becomes publicly visible and verifiable. Parties that wish to discover the CPS for a given telephone number do so by monitoring STI-CT logs for delegate certificates that include a CPS URI extension, extracting the TNAuthList and CPS URI from each certificate, and associating the covered TNs or SPCs with the indicated CPS endpoint. This approach provides a transparent, cryptographically verifiable discovery mechanism that does not require bilateral provisioning or static configuration between service providers.

The discovery process follows these steps:

1. A VESPER delegate certificate containing a TNAuthList and CPS URI extension is issued and submitted to a STI-CT log, generating an SCT.

2. A monitoring party observes the log, verifies the certificate chain to a trusted STI root, validates the SCT, and extracts the TN-to-CPS and SPC-to-CPS mappings.

3. Authentication Services and Verification Services consult these mappings to identify the appropriate CPS endpoint for a given call.

4. PASSporTs are published or retrieved using the discovered CPS URI as part of the OOB authentication process.

Implementations MAY maintain local caches of TN-to-CPS mappings, respecting certificate validity periods when using extracted data. CPS operators SHOULD publish delegate certificates in multiple STI-CT logs to ensure broad visibility. The CPS URI MUST resolve to a reachable and operational CPS that supports the VESPER OOB interface defined in this document.

To support resilience, operators SHOULD advertise multiple CPS instances, including regional or edge instances to improve latency and availability. Implementations SHOULD implement endpoint failover across available CPS instances, selecting among them using local policy such as lowest latency or geographic proximity.

8. Verification Service Procedures for VESPER OOB

Verification Services that retrieve and validate PASSporTs via the VESPER OOB model MUST implement the following procedures in addition to those defined fundamentally in [RFC8224] and specific to VESPER defined in [I-D.wendt-stir-vesper].

8.1. Retrieval and Validation Process

* CPS URI Resolution: Determine the CPS URI for the given TN or SPC by consulting TN-to-CPS mappings derived from monitoring STI-CT logs for delegate certificates containing the CPS URI extension, as described in the CPS URI and OOB CPS Discovery section of this document.

* PASSporT Retrieval: Submit a ’GET’ request to the CPS endpoint using a properly formed JWT in the Authorization header.

* Multiple PASSporT Handling: If the retrieved response contains multiple PASSporTs, the verifier MUST validate each PASSporT independently. All PASSporTs MUST share the same orig, dest, and iat values and MUST be signed by the same delegate certificate. If any PASSporT fails validation, the verifier SHOULD reject the entire set and SHOULD log the failure for diagnostic purposes. The verifier MAY apply local policy to determine which PASSporT types are actionable.

* Authentication JWT Validation: Ensure the JWT is:

  - Signed by a valid STI certificate that chains to a trusted root.

  - Contains matching ’iss’ and ’sub’ values as authorized in the certificate’s TNAuthList.

- Has an ’action’ claim set to "retrieve".

- Contains ’orig’ and ’dest’ claims matching the intended retrieval parameters.

8.2. PASSporT Validation

Once retrieved, the verifier MUST:

* Validate the PASSporT signature using the certificate chain in the ’x5c’ header.

* Confirm that the domain in the ’x5u’ URL matches the dNSName SubjectAltName of the signing certificate.

* Verify that the delegate certificate:

  - Is valid and chains to a trusted authority.

  - Contains valid SCTs proving inclusion in a certificate transparency log as defined in [I-D.ietf-stir-certificate-transparency].

  - Was issued under a valid, verifiable authority token.

* Check that the ’iat’ claim is within an acceptable range relative to the call time.

These validation steps ensure end-to-end trust in the originating identity of the call, even across heterogeneous network paths or in the absence of SIP Identity header delivery.

8.3. Connected Identity Validation

When a Connected Identity response PASSporT (rsp) is retrieved by the Verification Service (VS), it MUST be validated in accordance with the procedures defined in [I-D.ietf-stir-rfc4916-update] and the VESPER framework [I-D.wendt-stir-vesper].

Specifically:

The rsp PASSporT MUST be signed using a valid VESPER delegate certificate associated with the dest telephone number of the original call.

The certificate used to sign the rsp PASSporT MUST:

* Be issued under a valid authority token authorizing use of the dest number.

* Contain TNAuthList values that include the dest identifier.

* Include valid Signed Certificate Timestamps (SCTs) from a Certificate Transparency log.

The VS MUST validate the PASSporT signature and the delegate certificate's trust chain, including SCT verification and certificate expiration status.

The VS MUST confirm that the orig and dest claims in the rsp PASSporT match those of the original call. That is:

* The orig claim in the rsp PASSporT MUST match the orig claim of the original PASSporT.

* The dest claim in the rsp PASSporT MUST match the dest claim of the original PASSporT.

The key distinction from typical STIR verification is that the entity signing the rsp PASSporT is asserting control over the dest number, and the delegate certificate used in the signature MUST be valid for that number.

The iat claim in the rsp PASSporT MUST be within an acceptable freshness interval as defined by local policy.

If these validations succeed, the verifier can confirm that the called party has cryptographically asserted its identity using a VESPER-authorized certificate, completing the Connected Identity flow. Any failure in these validations MUST cause the rsp PASSporT to be rejected.

9. Security Considerations

All PASSporTs and Access JWTs MUST be signed using delegate certificates issued under the certificate policy defined in [RFC8226] and containing valid SCTs as defined in [I-D.ietf-stir-certificate-transparency]. Verifiers MUST validate the certificate trust chain and SHOULD verify SCT inclusion against known CT log sets. Access JWTs MUST use the ES256 algorithm, MUST be scoped per transaction with short validity intervals (e.g., 5 minutes), and MUST include a jti claim for replay prevention. CPS implementations MUST cache recent jti values and reject reuse within the validity window. The response_uuid MUST only be disclosed to authenticated parties authorized to retrieve the original publish and

MUST NOT be exposed via unauthenticated endpoints or logs. CPS implementations MUST restrict access to the response_uuid and its associated response endpoint to the authenticated parties of the original transaction. Specifically, the CPS MUST verify that a party requesting the Connected Identity response PASSporT (via GET /passports/response/{UUID} or push interfaces) is the same entity that performed the original publish, as identified by the iss claim in their Access JWT. The CPS MUST NOT disclose whether a response_uuid exists or has a pending response to any other party. CPS implementations SHOULD return 404 (rather than 403) for unauthorized UUID lookups to prevent UUID existence confirmation.

CPS operators MUST enforce rate limiting across all endpoints and MUST retain identity data only as long as operationally necessary.

10. IANA Considerations

This document has no IANA actions.

11. References

11.1. Normative References

[I-D.ietf-httpapi-idempotency-key-header]
Jena, J. and S. Dalal, "The Idempotency-Key HTTP Header Field", Work in Progress, Internet-Draft, draft-ietf-httpapi-idempotency-key-header-07, 15 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-httpapi-idempotency-key-header-07>.

[I-D.ietf-stir-certificate-transparency]
Wendt, C., ナ嗟iwa, R., Fenichel, A., and V. A. Gaikwad, "STI Certificate Transparency", Work in Progress, Internet-Draft, draft-ietf-stir-certificate-transparency-01, 23 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-stir-certificate-transparency-01>.

[I-D.ietf-stir-rfc4916-update]
Peterson, J. and C. Wendt, "Connected Identity for STIR", Work in Progress, Internet-Draft, draft-ietf-stir-rfc4916-update-07, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-stir-rfc4916-update-07>.

[I-D.sliwa-stir-cert-cps-ext]
ナ嗟iwa, R. and C. Wendt, "Call Placement Service (CPS) URI Certificate Extension for STI Certificates", Work in

Progress, Internet-Draft, draft-sliwa-stir-cert-cps-ext-01, 3 November 2025, <https://datatracker.ietf.org/doc/html/draft-sliwa-stir-cert-cps-ext-01>.

[I-D.wendt-stir-vesper] Wendt, C. and R. ナ嗟iwa, "VESPER - Verifiable STI Presentation and Evidence for RTU", Work in Progress, Internet-Draft, draft-wendt-stir-vesper-07, 31 March 2026, <https://datatracker.ietf.org/doc/html/draft-wendt-stir-vesper-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>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/rfc/rfc3261>.

[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/rfc/rfc3986>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <https://www.rfc-editor.org/rfc/rfc4122>.

[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>.

[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>.

[RFC8224] Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 8224, DOI 10.17487/RFC8224, February 2018, <https://www.rfc-editor.org/rfc/rfc8224>.

[RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <https://www.rfc-editor.org/rfc/rfc8225>.

[RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <https://www.rfc-editor.org/rfc/rfc8226>.

[RFC8785] Rundgren, A., Jordan, B., and S. Erdtman, "JSON Canonicalization Scheme (JCS)", RFC 8785, DOI 10.17487/RFC8785, June 2020, <https://www.rfc-editor.org/rfc/rfc8785>.

[RFC8816] Rescorla, E. and J. Peterson, "Secure Telephone Identity Revisited (STIR) Out-of-Band Architecture and Use Cases", RFC 8816, DOI 10.17487/RFC8816, February 2021, <https://www.rfc-editor.org/rfc/rfc8816>.

[RFC9060] Peterson, J., "Secure Telephone Identity Revisited (STIR) Certificate Delegation", RFC 9060, DOI 10.17487/RFC9060, September 2021, <https://www.rfc-editor.org/rfc/rfc9060>.

[RFC9447] Peterson, J., Barnes, M., Hancock, D., and C. Wendt, "Automated Certificate Management Environment (ACME) Challenges Using an Authority Token", RFC 9447, DOI 10.17487/RFC9447, September 2023, <https://www.rfc-editor.org/rfc/rfc9447>.

11.2. Informative References

[ATIS-1000105] ATIS, "ATIS-1000105 - Signature-based Handling of Asserted information using Tokens (SHAKEN): Out-of-Band PASSporT Transmission Between Service Providers that Interconnect using TDM", n.d., <https://access.atis.org/higherlogic/ws/public/download/79509/ATIS-1000105.pdf>.

Acknowledgments

The authors thank the contributors of the STIR working group and authors of ATIS-1000105, many of the API mechanisms have been aligned and extended in this document to support the VESPER OOB Framework for PASSporT delivery signed with delegate certificates.

Authors' Addresses

Chris Wendt
Somos Inc.
United States of America
Email: chris@appliedbits.com

Rob ナ嗟iwa
Somos Inc.
United States of America
Email: robjsliwa@gmail.com