WG Working Group
Internet-Draft
Intended status: Standards Track
Expires: 7 May 2026

C. Wendt
R. ナ嗟iwa
Somos Inc.

3 November 2025

VESPER Out-of-Band OOB
draft-wendt-stir-vesper-oob-01

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

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 framework by introducing support for vetted delegate certificates using authority tokens and certificate transparency logs and monitoring to enhance reliability and trust for the delegation of telephone number specific certificates and the associated claims authorized to be made by the use of those certificates for signed PASSporTs. The use cases motivating these enhancements are outlined in [I-D.wendt-stir-vesper-use-cases].

This document describes how to expand the VESPER framework to use an out-of-band delivery mechanism corresponding to the model described in [RFC8816]. The VESPER framework defines how delegate certificates are issued based on authority tokens that attest to the vetting and authorization of the entity to use a telephone number and assert other related claim information. This specification extends this to enable authorized delegate certificate holders, who sign calls via a STIR Authentication Service, to deliver PASSporTs containing authorized, verifiable claims over a non-SIP-based path. These PASSporTs can be retrieved and validated by a STIR Verification Service, similar to SIP-based STIR as defined in [RFC8224], thereby maintaining continuity of trust across heterogeneous networks.

OOB delivery is critical in extending the utility of STIR to networks where SIP identity headers cannot be delivered end-to-end. It provides a verifiable alternative path for transmitting PASSporTs and proving the originating telephone number’s association to the signing identity.

The Vesper OOB delivery model assumes a one-way publish-and-retrieve interface based on a defined 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 based on [I-D.sliwa-stir-oob-transparent-discovery] with the corresponding Vesper requirement to utilize and verify STI certificate transparency receipts with delegate certificates used in Vesper OOB.

It also optionally supports the STIR concept of Connected Identity adopted in VESPER framework as well, where not only the originator of a call or message can authenticate their telephone number, but the destination party can also prove their telephone number back to the originator to have a full end-to-end bi-directional trust relationship. This is based on Connected Identity defined in [I-D.ietf-stir-rfc4916-update] and 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 Entity STIR Passport Entity Representation [I-D.wendt-stir-vesper].

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

Delegate Certificate: A certificate issued to an enterprise or user entity asserting right-to-use for a telephone number, based on an authority token, defined in [RFC9060].

Authority Token: A signed assertion that authorizes the issuance of a delegate certificate and represents the vetting of a subject’s control over 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: Defines the use of STI certificate transparency log monitoring and CPS URI extension in certificates for announcing CPS locations for certificates [I-D.sliwa-stir-oob-transparent-discovery].

3. Vesper OOB Architectural Overview

The VESPER OOB architecture consists of three main functional components that work together to enable 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 both simulataneous and parallel in-band and out-of-band delivery of telephone number authentication information, ensuring

interoperability across a variety of telephony related network environments. Figure 1 illustrates the flow of identity data between the authentication service, the out-of-band Call Placement Service (CPS), and the verification service.

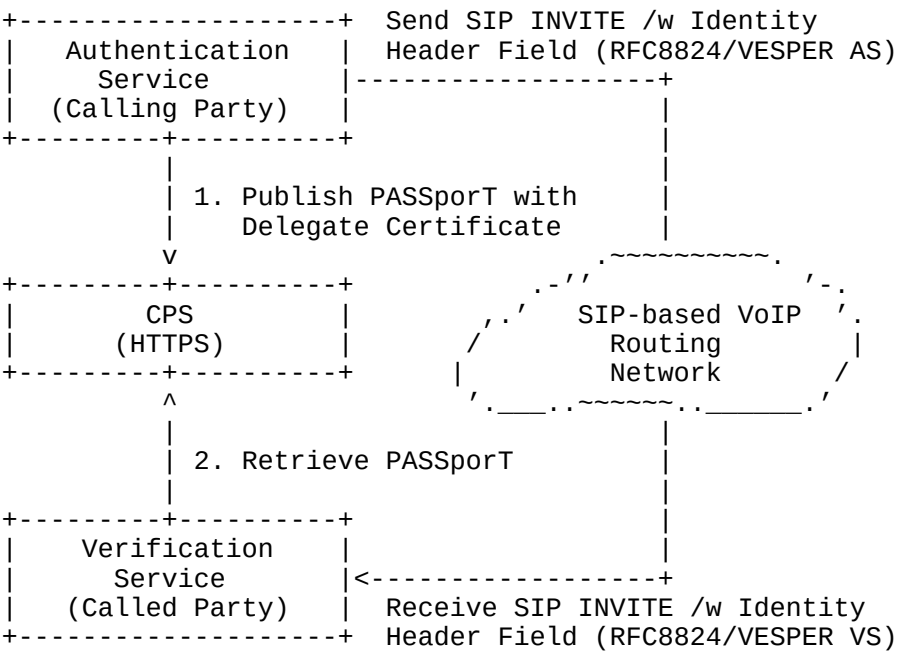

Figure 1 - Architecture showing both in-band and out-of-band PASSporT delivery

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. The POST endpoint MUST require authentication Access JWT. The GET endpoint MAY be unauthenticated. 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 a STIR eco-system trusted root.

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'	Timestamp indicating the time the call is guaranteed to complete
'jti'	Unique token ID. SHOULD be used for replay prevention and audit.
'action'	Operation intent: "publish", "retrieve", or "respond".
'aud'	The CPS hostname (e.g., "cps.example.net"). MUST match the target server.
'iss'	The SPC or TN representing the signer. MUST match TNAuthList in cert.
'sub'	Same value as iss. Identifies the subscriber authorized to act.
'orig'	Object with TN/URI of the originating party.
'dest'	Object with TN/URI of the destination party.
'passports'	OPTIONAL. For 'publish', SHA-256 JCS digest of the canonicalized passports.
'rsp_passport'	OPTIONAL. For 'respond', SHA-256 JCS digest of the rsp_passport.

Table 1: Access JWT Claims

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,
  "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,
  "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 hash MUST match the canonicalized JSON in the request body using JSON Canonicalization Scheme (JCS) [RFC8785].

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] supports a "re-publish" action, because the VESPER-OOB discovery mechanism is different and re-publishing PASSporTs is not required for VESPER-OOB, CPSs that support this specification are not dependent on support the initiation of this action or otherwise communicate to other CPSs supporting this specification including the inclusion of "token" fields, but the intent is to be compatible with implementations that support both specifications

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].

4.2.2.4. Request Body

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

* passports: REQUIRED. An array of PASSporT strings signed by the calling party.

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].

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. 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..." // Base64-encoded PASSporT string(s)
  ],
  "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.

4.2.3.6. Example Request

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

4.2.3.7. Example Response

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

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

4.2.3.8. 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"
}
```

4.2.3.9. 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 via:

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

Server-Sent Events (SSE): GET /passports/response/stream/{UUID} (with Accept: text/event-stream header)

These interfaces allow immediate delivery of Connected Identity responses when available.

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..." // Signed PASSporT by calling party
  ]
}
```

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.wendt-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 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 the delegate certificates through the CPS URI extension defined in [I-D.sliwa-stir-cert-cps-ext]. Verifiers are expected to obtain the CPS URI for a specific telephone number via transparency-enabled discovery mechanisms described in [I-D.sliwa-stir-oob-transparent-discovery]. The CPS URI identifies the base URL for the Call Placement Service responsible for publishing and serving PASSporTs for calls associated with that telephone number.

The CPS URI MUST resolve to a reachable and operational CPS that supports the VESPER OOB interface defined in this document. It is assumed that the CPS implements the endpoints defined in the HTTPS interface specification, including '/health', '/passports/{DEST}/{ORIG}', and appropriate authorization mechanisms. The CPS will provide a response_uuid in its response to the publish operation, which is used by the calling and called parties in subsequent API calls for Connected Identity.

////////// Delegate certificates MUST reference a CPS via the CPS URI extension and MUST be resolvable through the Discovery service specified for Vesper OOB. The Discovery service MUST return multiple

CPS instances for each delegate certificate to provide redundancy. Operators SHOULD advertise regional or edge CPS instances to improve latency and availability for verifiers and retrievers.

Verifiers and retrievers MUST implement endpoint failover across the set of CPS instances provided by Discovery and SHOULD select among them using local policy (e.g., lowest latency or geographically closest instances).

Deployments that also support CPS-to-CPS replication MAY perform inter-CPS propagation by invoking the publish API using "action": "republish" semantics. In this mode, a CPS acts as a client to peer CPS servers to broaden availability of published PASSporTs. The wire format and validation requirements are otherwise identical to "action": "publish", except the policy MUST authorize the republish CPS to perform this operation. ////////////

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: Retrieve the CPS URI from an appropriate CPS discovery service as discussed and defined in [I-D.sliwa-stir-oob-transparent-discovery] to locate the specific ’/passports/{DEST}/{ORIG}’ endpoint.

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

* 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 provided certificate referenced in the ’x5c’ Header.

* Verify that the delegate certificate:

  - Is valid and chains to a trusted authority.

  - Contains valid SCTs proving inclusion in a certificate transparency log.

  - Was issued under a valid, verifiable authority token (directly or via reference).

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

* Optionally, verify the transparency receipt (if present) that correlates the certificate and signing event.

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. Privacy Considerations

The VESPER OOB framework facilitates the transmission and verification of signed identity assertions that may include personally identifiable information (PII), such as telephone numbers and organizational names. This section outlines key privacy considerations to ensure implementations protect individual privacy and comply with applicable regulations.

9.1. Minimization of Identity Claims

PASSporTs exchanged via VESPER OOB SHOULD contain only the minimum necessary identity claims to establish the intended trust relationship. The inclusion of unnecessary claims in the PASSporT payload or certificate extensions may reveal sensitive information about users or organizations. Implementations SHOULD avoid including additional metadata beyond what is required for call verification.

9.2. Use of Connected Identity

The Connected Identity feature allows both parties in a communication to share independently signed identity assertions. While this can enhance trust, it also introduces a risk of correlation between calling and called parties. Implementers SHOULD consider allowing users to opt out of responding with Connected Identity or restrict participation to enterprise contexts where such correlation is expected.

The response_uuid MUST only be disclosed to the authenticated parties authorized to retrieve the original publish. Servers SHOULD keep the response_uuid lifetime short and MUST NOT expose it via unauthenticated endpoints or logs.

9.3. Compliance with Regional Privacy Regulations

Operators deploying VESPER OOB MUST assess their processing of PASSporTs and related metadata for compliance with applicable data protection laws (e.g., GDPR, CCPA). This includes evaluating:

* Whether telephone numbers are treated as personal data
* Lawful basis for processing and retention
* User transparency and rights of access, rectification, and erasure

Audit mechanisms and data subject request workflows SHOULD be implemented when operating in regulated jurisdictions.

9.4. Transparency and Logging

While logging of CPS activity is important for fraud detection and accountability, implementations MUST avoid logging full PASSporT payloads or tokens unless strictly necessary. Where logs include sensitive fields, they SHOULD be protected with access controls and subject to audit.

The use of transaction-specific UUIDs instead of callback URLs minimizes the privacy exposure associated with publishing service endpoints. Only parties with the appropriate authorization token (Access JWT) can retrieve or respond to a PASSporT exchange, which helps ensure that identity data is not leaked to unauthorized entities. Connected Identity responses are associated only with the UUID provided to the intended recipient, reducing correlation risk across sessions.

10. Security Considerations

10.1. Trust Anchors and Certificate Transparency

All JWTs and PASSporTs MUST be signed using delegate certificates anchored in a trusted STI-CA root and SHOULD be accompanied by Signed Certificate Timestamps (SCTs) to prove log inclusion. Verifiers SHOULD validate SCT presence and match against a known CT log set.

10.2. Cross-Origin and CORS

CPS servers that expose web-facing endpoints MAY implement CORS headers to restrict origin access to approved domains or application scopes.

10.3. Logging and Audit

CPS operators SHOULD log authentication attempts, JWT usage (by jti), PASSporT publication, and response_url usage for auditing and potential fraud investigation. Logs SHOULD be retained securely and in accordance with privacy regulations.

10.4. UUID-Based Transaction Integrity

The specification relies on cryptographically random UUIDs as transaction identifiers for Connected Identity responses. These UUIDs MUST be generated by the CPS using secure random generation techniques and MUST be unguessable to prevent targeted scraping or brute-force enumeration of published PASSporTs or responses.

10.5. Replay and Reuse Mitigation

The use of the ’jti’ (JWT ID) field in Access JWTs supports replay protection and auditability. CPS implementations SHOULD maintain short-term caches of recent JTIs and reject duplicate requests. JWTs MUST have short time-to-live values (e.g., 5 minutes) to reduce exposure from replay attacks.

10.6. CPS Operator Responsibilities

CPS operators MUST enforce authorization controls and rate limiting across all endpoints. They are responsible for securing logs, ensuring endpoint availability, monitoring for anomalies, and maintaining certificate trust anchors. Any retained identity data MUST be stored securely and retained only as long as operationally necessary.

11. IANA Considerations

This document has no IANA actions.

12. References

12.1. 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-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-00, 5 September 2025, <https://datatracker.ietf.org/doc/html/draft-sliwa-stir-cert-cps-ext-00>.

[I-D.sliwa-stir-oob-transparent-discovery]
ナ嗟iwa, R. and C. Wendt, "Transparent Discovery of STIR Out-of-Band Call Placement Services", Work in Progress, Internet-Draft, draft-sliwa-stir-oob-transparent-discovery-00, 5 September 2025, <https://datatracker.ietf.org/doc/html/draft-sliwa-stir-oob-transparent-discovery-00>.

[I-D.wendt-stir-certificate-transparency]
Wendt, C., ナ嗟iwa, R., Fenichel, A., and V. A. Gaikwad, "STI Certificate Transparency", Work in Progress, Internet-Draft, draft-wendt-stir-certificate-transparency-06, 11 June 2025, <https://datatracker.ietf.org/doc/html/draft-wendt-stir-certificate-transparency-06>.

[I-D.wendt-stir-vesper]
Wendt, C. and R. ナ嗟iwa, "VESPER - Framework for VErifiable STI Personas", Work in Progress, Internet-Draft, draft-

wendt-stir-vesper-05, 5 September 2025,
<https://datatracker.ietf.org/doc/html/draft-wendt-stir-vesper-05>.

[I-D.wendt-stir-vesper-use-cases]
Wendt, C., "Verifiable STI Persona (VESPER) Use Cases and Requirements", Work in Progress, Internet-Draft, draft-wendt-stir-vesper-use-cases-02, 11 August 2025,
<https://datatracker.ietf.org/doc/html/draft-wendt-stir-vesper-use-cases-02>.

[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>.

[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>.

12.2. 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