

Secure RESTful Interfaces: Draft Profiles for the Use of OpenID Connect

Draft Version 1.4

January 12, 2015

The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation.

Approved for Public Release; Distribution Unlimited. Case Number 14-3178

This technical data was produced for the U. S. Government under Contract Numbers VA791-P-0032, VA791-9-0042, VA798A-11-P-0015, and VA118A-13-D-0037 and is subject to Federal Acquisition Regulation Clause 52.227-14, Rights in Data – General, Alt. II (JUN 1987), Alt. III (JUN 1987) and Alt. IV (JUN 1987).

No other use other than that granted to the U. S. Government under that Clause is authorized without the express written permission of The MITRE Corporation.

For further information, please contact The MITRE Corporation, Contracts Office, 7515 Colshire Drive, McLean, VA 22102-7539, (703) 983-6000.

©2014 The MITRE Corporation. This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

Sponsor: Department of Veterans Affairs

Contract No.: VA791-P-0042

Project No.: 40144028

**McLean, VA**

J. Richer

M. Russell

January 2015

This page intentionally left blank.

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 1.0 | 6/30/2014 | J. Richer, M. Russell | Initial draft |
| 1.1 | 8/15/2014 | M. Russell | Incorporated additional edits from J. Richer;  |
| 1.2 | 9/11/2014 | M. Russell | Reorganized document, moving VA-specific considerations to the introduction to enable the profile to be shared broadly |
| 1.3 | 10/15/2014 | M. Russell | Added notices and rights legend for public release |
| 1.4 | 1/7/2015 | J. Richer | Added examples, clarified caching and other minor behaviors |

Table of Contents

Table of Contents iv

Introduction iv

Draft Secure RESTful Interface Profile for the Use of OpenID Connect 1.0 1

1. Terminology 1

2. ID Tokens 2

3. UserInfo Endpoint 3

4. Request Objects 4

5. Authentication Context 4

6. Discovery 4

List of Acronyms 7

References 8

# Introduction

The attached profile for OpenID Connect 1.0 was developed in support of the Secure RESTful Interfaces Profile project for the Department of Veterans Affairs (VA) Deputy Chief Information Officer (CIO), Office of Information and Technology (OIT), Architecture, Strategy, and Design (ASD).

These profiles are intended for general application across a wide range of VA use cases. They build on a prior OpenID Connect profile developed by MITRE in support of the RESTful Health Exchange (RHEx) initiative. While RHEx and BlueButton+ addressed specific health care use cases, this document defines general guidance applicable both within and beyond the health care domain.

It is anticipated that these draft profiles will be revised through outreach discussions with VA stakeholders and external organizations including the Office of the National Coordinator for Health IT (ONC) and the Open Source Electronic Health Record Alliance (OSEHRA). In addition, emerging security threat information, such as novel attacks on OAuth or advances in cryptanalysis, may require updates to these profiles.

One recommended action for VA identified in drafting this profile is to engage with its mission partners to establish standard values for the Authentication Methods Reference (*amr*) claim. This claim enables an identity provider to notify the relying party of the specific mechanism(s) used to authenticate the user (e.g., memorized password + one-time password [OTP] sent to mobile device over SMS). Previous standards developed for SAML, such as *Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0*, [1] provide detailed classifications and references for many different authentication mechanisms that could provide a basis for standard *amr* values.

# Draft Secure RESTful Interface Profile for the Use of OpenID Connect 1.0

This document profiles the OpenID Connect specification for use in providing identity information supporting secure Representational State Transfer (RESTful) interfaces. Because OpenID Connect is built on OAuth 2.0, this profile inherits all requirements of the Draft Profiles for the use of OAuth 2.0. All requirements herein are in addition to the OAuth 2.0 profile.

# Terminology

This profile inherits terminology from RFC6749 [2], RFC6750 [3], RFC2119 [4], OpenID Connect Core [5].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [4].

Table 1 below defines some of the terms defined in the standards referenced above and used in this profile specification. These definitions are provided for the reader’s convenience and not meant to supersede or modify the terms’ original, authoritative definitions in the respective standards.

Table 1 – Referenced Terms

| Term | Definition |
| --- | --- |
| Access Token | A token issued to the client and passed to the protected resource as proof of access authorization |
| Authorization Endpoint | An authorization server endpoint to which clients submit authorization requests (via resource owner user agent redirection); the authorization endpoint authenticates resource owners and prompts them to approve or deny authorization requests |
| Authorization Server | A server that implements the OAuth 2.0 authorization endpoint, token endpoint, and optionally other OAuth endpoints, and issues authorization tokens to clients. |
| Client | A software application that requests OAuth access tokens from an authorization server |
| ID Token | A JavaScript Object Notation (JSON) Web Token (JWT) issued by an OpenID Provider to a Relying Party to communicate the authenticated user’s identity and other optional claims about the user’s attributes or authentication context |
| OpenID Provider | An OAuth 2.0 authorization server that responds to OpenID Connect authentication requests and issues ID Tokens |
| Protected Resource | A resource (API, application, or data) to which authorization is granted through the OAuth framework  |
| Relying Party  | A system or application that consumes authentication information from an identity provider (e.g., an OpenID Provider) |
| Resource Owner | A user with the authority to authorize clients to access protected resources on his/her behalf |
| Token Endpoint | An authorization server endpoint to which clients submit authorization codes or client credentials in order to receive access tokens  |
| Token Introspection Endpoint | An authorization server endpoint to which protected resources can submit tokens received from clients in order to obtain token metadata such as validity period or approved scope  |
| Token Revocation Endpoint | An authorization server endpoint to which clients can submit token revocation requests to indicate that tokens will no longer be used |
| UserInfo Endpoint | A protected resource that responds to UserInfo requests from clients that have been authorized and issued access tokens by an OpenID Provider, and returns claims about the authenticated user |

# ID Tokens

All ID Tokens MUST be signed by the OpenID Provider’s private signature key. All clients MUST validate the signature of an ID Token before accepting it using the public key of the issuing server, which is published in JWK format. ID Tokens MAY be encrypted using the appropriate key of the requesting client.

All clients MUST verify the following in received ID tokens:

* *iss:* The “issuer” field is the Uniform Resource Locator (URL) of the expected issuer
* *aud:* The “audience” field contains the client ID of the client
* *exp, iat, nbf:* The “expiration”, “issued at”, and “not before” timestamps for the token are dates (integer number of seconds since from 1970-01-01T00:00:00Z UTC) within acceptable ranges

The ID Token MUST expire and SHOULD have an active lifetime no longer than five minutes.

This example ID token has been signed using the server’s RSA key:

eyJhbGciOiJSUzI1NiJ9.eyJhdXRoX3RpbWUiOjE0MTg2OTg3ODIsImV4cCI6MTQxODY5OTQxMiwic3ViIjoiNldaUVBwblF4ViIsIm5vbmNlIjoiMTg4NjM3YjNhZjE0YSIsImF1ZCI6WyJjMWJjODRlNC00N2VlLTRiNjQtYmI1Mi01Y2RhNmM4MWY3ODgiXSwiaXNzIjoiaHR0cHM6XC9cL2lkcC1wLm1pdHJlLm9yZ1wvIiwiaWF0IjoxNDE4Njk4ODEyfQ.mQc0rtL56dnJ7\_zO\_fx8-qObsQhXcn-qN-FC3JIDBuNmP8i11LRA\_sgh\_omRRfQAUhZD5qTRPAKbLuCD451lf7ALAUwoGg8zAASI5QNGXoBVVn7buxPd2SElbSnHxu0o8ZsUZZwNpircWNUlYLje6APJf0kre9ztTj-5J1hRKFbbHodR2I1m5q8zQR0ql-FoFlOfPhvfurXxCRGqP1xpvLLBUi0JAw3F8hZt\_i1RUYWMqLQZV4VU3eVNeIPAD38qD1fxTXGVEd2XDJpmlcxjrWxzJ8fGfJrbsiHCzmCjflhv34O22zb0lJpC0d0VScqxXjNTa2-ULyCoehLcezmssg

Its claims are as follows:

 {

 "auth\_time": 1418698782,

 "exp": 1418699412,

 "sub": "6WZQPpnQxV",

 "nonce": "188637b3af14a",

 "aud": [

 "c1bc84e4-47ee-4b64-bb52-5cda6c81f788"

 ],

 "iss": "https:\\/\\/idp-p.mitre.org\\/",

 "iat": 1418698812

}

# UserInfo Endpoint

Servers MUST support the UserInfo Endpoint and, at a minimum, the “openid” scope and “sub” (subject) claims returned from there for all users. The client sends a request to the UserInfo Endpoint like the following:

GET /userinfo HTTP/1.1

Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJleHAiOjE0MTg3MDI0MTIsImF1ZCI6WyJjMWJjODRlNC00N2VlLTRiNjQtYmI1Mi01Y2RhNmM4MWY3ODgiXSwiaXNzIjoiaHR0cHM6XC9cL2lkcC1wLm1pdHJlLm9yZ1wvIiwianRpIjoiZDNmN2I0OGYtYmM4MS00MGVjLWExNDAtOTc0YWY3NGM0ZGUzIiwiaWF0IjoxNDE4Njk4ODEyfQ.iHMz\_tzZ90\_b0QZS-AXtQtvclZ7M4uDAs1WxCFxpgBfBanolW37X8h1ECrUJexbXMD6rrj\_uuWEqPD738oWRo0rOnoKJAgbF1GhXPAYnN5pZRygWSD1a6RcmN85SxUig0H0e7drmdmRkPQgbl2wMhu-6h2Oqw-ize4dKmykN9UX\_2drXrooSxpRZqFVYX8PkCvCCBuFy2O-HPRov\_SwtJMk5qjUWMyn2I4Nu2s-R20aCA-7T5dunr0iWCkLQnVnaXMfA22RlRiU87nl21zappYb1\_EHF9ePyq3Q353cDUY7vje8m2kKXYTgc\_bUAYuW-W3SMSw5UlKaHtSZ6PQICoA

Accept: text/plain, application/json, application/\*+json, \*/\*

Host: idp-p.mitre.org

Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.2.3 (java 1.5)

And receives a document in response like the following:

HTTP/1.1 200 OK

Date: Tue, 16 Dec 2014 03:00:12 GMT

Access-Control-Allow-Origin: \*

Content-Type: application/json;charset=ISO-8859-1

Content-Language: en-US

Content-Length: 333

Connection: close

{

 "sub": "6WZQPpnQxV",

 "name": "Steve Emeritus",

 "preferred\_username": "steve",

 "given\_name": "Stephen",

 "family\_name": "Emeritus",

 "nickname": "Steve",

 "gender": "M",

 "updated\_time": "2014-09-24 14:27:43.701000",

 "birthdate": "1980-01-01",

 "email": "steve.e@example.com",

 "email\_verified": true,

 "phone\_number": "857-555-1234",

 "phone\_number\_verified": true

}

Servers MUST support the generation of JavaScript Object Signing and Encryption (JOSE)-protected responses from the UserInfo Endpoint in addition to unsigned JSON objects. Signed responses MUST be signed by the OpenID Provider’s key, and encrypted responses MUST be encrypted with the authorized client’s key. The OpenID Provider MUST support the RS256 signature method (the Rivest, Shamir, and Adleman (RSA) signature algorithm with a 256-bit hash) and MAY use other asymmetric signature and encryption methods listed in the JSON Web Algorithms (JWA) specification [6].

# Request Objects

Clients MAY optionally send requests to the authorization endpoint as signed or encrypted request objects. Servers MUST accept requests containing a request object signed by the client’s private key. Servers MUST validate the signature on such requests against the client’s registered public key. Clients must register their keys during client registration as described in Section 4 of the Secure RESTful Interface Profiles for OAuth 2.0 [7]. Servers MUST accept request objects encrypted to the server’s public key.

Servers MAY accept request objects by reference (using the *request\_uri* parameter).

# Authentication Context

OpenID Providers MUST provide *acr* (authentication context class reference, equivalent to the Security Assertion Markup Language (SAML) element of the same name) and *amr* (authentication methods reference) values in ID tokens.

The standardized Uniform Resource Identifiers (URIs) established by the Federal Identity, Credential, and Access Management (FICAM) Trust Framework [8] should be used for the *acr* values, depending on the Level of Assurance (LOA) of the authentication performed by the OpenID Provider:

* LOA 1: <http://idmanagement.gov/ns/assurance/loa/1>
* LOA 2: <http://idmanagement.gov/ns/assurance/loa/2>
* LOA 3: <http://idmanagement.gov/ns/assurance/loa/3>

Note that OpenID Connect cannot provide LOA 4 identity assertions.

The *amr* value is an array of strings describing the set of mechanisms used to authenticate the user to the OpenID Provider. Providers that require multi-factor authentication will typically provide multiple values (for example, memorized password plus hardware-token-generated one-time password). The specific values must be agreed upon and understood between the OpenID Provider and any Relying Parties. FICAM has not published standard values that would be suitable for this field, so interconnected partners will need to agree to common values for this claim.

# Discovery

All OpenID Connect servers are uniquely identified by a URL known as the “issuer”. This URL serves as the prefix of a service discovery endpoint as specified in the OpenID Connect Discovery standard [9]. The discovery document MUST contain the following fields:

* *issuer*: The fully qualified issuer URL of the server
* *authorization\_endpoint*: The fully qualified URL of the server’s authorization endpoint defined by RFC 6749 [2]
* *token\_endpoint*: The fully qualified URL of the server’s token endpoint defined by RFC 6749 [2]
* *introspection\_endpoint*: The fully qualified URL of the server’s introspection endpoint defined by [10]
* *revocation\_endpoint*: The fully qualified URL of the server’s revocation endpoint defined by [11]
* *jwks\_uri*: The fully qualified URI of the server’s public key in JSON Web Key Set (JWKS) format

The following example shows the JSON document found at a discovery endpoint for an authorization server:

{

 "request\_parameter\_supported": true,

 "id\_token\_encryption\_alg\_values\_supported": [

 "RSA-OAEP", "RSA1\_5", "RSA-OAEP-256"

 ],

 "registration\_endpoint": ["https://idp-p.mitre.org/register"](https://idp-p.mitre.org/register),

 "userinfo\_signing\_alg\_values\_supported": [

 "HS256", "HS384", "HS512", "RS256", "RS384", "RS512"

 ],

 "token\_endpoint": ["https://idp-p.mitre.org/token"](https://idp-p.mitre.org/token),

 "request\_uri\_parameter\_supported": false,

 "request\_object\_encryption\_enc\_values\_supported": [

 "A192CBC-HS384", "A192GCM", "A256CBC+HS512",

 "A128CBC+HS256", "A256CBC-HS512",

 "A128CBC-HS256", "A128GCM", "A256GCM"

 ],

 "token\_endpoint\_auth\_methods\_supported": [

 "client\_secret\_post",

 "client\_secret\_basic",

 "client\_secret\_jwt",

 "private\_key\_jwt",

 "none"

 ],

 "userinfo\_encryption\_alg\_values\_supported": [

 "RSA-OAEP", "RSA1\_5",

 "RSA-OAEP-256"

 ],

 "subject\_types\_supported": [

 "public", "pairwise"

 ],

 "id\_token\_encryption\_enc\_values\_supported": [

 "A192CBC-HS384", "A192GCM", "A256CBC+HS512",

 "A128CBC+HS256", "A256CBC-HS512", "A128CBC-HS256",

 "A128GCM", "A256GCM"

 ],

 "claims\_parameter\_supported": false,

 "jwks\_uri": ["https://idp-p.mitre.org/jwk"](https://idp-p.mitre.org/jwk),

 "id\_token\_signing\_alg\_values\_supported": [

 "HS256", "HS384", "HS512", "RS256", "RS384", "RS512", "none"

 ],

 "authorization\_endpoint": ["https://idp-p.mitre.org/authorize"](https://idp-p.mitre.org/authorize),

 "require\_request\_uri\_registration": false,

 "introspection\_endpoint": ["https://idp-p.mitre.org/introspect"](https://idp-p.mitre.org/introspect),

 "request\_object\_encryption\_alg\_values\_supported": [

 "RSA-OAEP", ”RSA1\_5", "RSA-OAEP-256"

 ],

 "service\_documentation": ["https://idp-p.mitre.org/about"](https://idp-p.mitre.org/about),

 "response\_types\_supported": [

 "code", "token"

 ],

 "token\_endpoint\_auth\_signing\_alg\_values\_supported": [

 "HS256", "HS384", "HS512", "RS256", "RS384", "RS512"

 ],

 "revocation\_endpoint": ["https://idp-p.mitre.org/revoke"](https://idp-p.mitre.org/revoke),

 "request\_object\_signing\_alg\_values\_supported": [

 "HS256", "HS384", "HS512", "RS256", "RS384", "RS512"

 ],

 "claim\_types\_supported": [

 "normal"

 ],

 "grant\_types\_supported": [

 "authorization\_code",

 "implicit",

 "urn:ietf:params:oauth:grant-type:jwt-bearer",

 "client\_credentials",

 "urn:ietf:params:oauth:grant\_type:redelegate"

 ],

 "scopes\_supported": [

 "profile", "openid", "email", "address", "phone", "offline\_access"

 ],

 "userinfo\_endpoint": ["https://idp-p.mitre.org/userinfo"](https://idp-p.mitre.org/userinfo),

 "userinfo\_encryption\_enc\_values\_supported": [

 "A192CBC-HS384", "A192GCM", "A256CBC+HS512","A128CBC+HS256",

 "A256CBC-HS512", "A128CBC-HS256", "A128GCM", "A256GCM"

 ],

 "op\_tos\_uri": ["https://idp-p.mitre.org/about"](https://idp-p.mitre.org/about),

 "issuer": ["https://idp-p.mitre.org/"](https://idp-p.mitre.org/),

 "op\_policy\_uri": ["https://idp-p.mitre.org/about"](https://idp-p.mitre.org/about),

 "claims\_supported": [

 "sub", "name", "preferred\_username", "given\_name", "family\_name",

 "middle\_name", "nickname", "profile", "picture", "website",

 "gender", "zone\_info", "locale", "updated\_time", "birthdate",

 "email", "email\_verified", "phone\_number", "address"

 ]

}

Clients and protected resources SHOULD cache this discovery information. It is RECOMMENDED that servers provide cache information through HTTP headers and make the cache valid for at least one week.

The server MUST provide its public key in JWK Set format, such as the following 2048-bit RSA key:

{

 "keys": [

 {

 "alg": "RS256",

 "e": "AQAB",

 "n": "o80vbR0ZfMhjZWfqwPUGNkcIeUcweFyzB2S2T-hje83IOVct8gVg9FxvHPK1ReEW3-p7-A8GNcLAuFP\_8jPhiL6LyJC3F10aV9KPQFF-w6Eq6VtpEgYSfzvFegNiPtpMWd7C43EDwjQ-GrXMVCLrBYxZC-P1ShyxVBOzeR\_5MTC0JGiDTecr\_2YT6o\_3aE2SIJu4iNPgGh9MnyxdBo0Uf0TmrqEIabquXA1-V8iUihwfI8qjf3EujkYi7gXXelIo4\_gipQYNjr4DBNlE0\_\_RI0kDU-27mb6esswnP2WgHZQPsk779fTcNDBIcYgyLujlcUATEqfCaPDNp00J6AbY6w",

 "kty": "RSA",

 "kid": "rsa1"

 }

 ]

}

Clients and protected resources SHOULD cache this key. It is RECOMMENDED that servers provide cache information through HTTP headers and make the cache valid for at least one week.

# List of Acronyms

| Acronym | Definition |
| --- | --- |
| ASD | Architecture, Strategy, and Design |
| FICAM | Federal Identity, Credential, and Access Management |
| JOSE | JavaScript Object Signing and Encryption |
| JSON | JavaScript Object Notation |
| JWA | JSON Web Algorithms |
| JWT | JSON Web Token |
| JWKS | JSON Web Key Set |
| LOA | Level of Assurance |
| OIT | Office of Information & Technology |
| ONC | Office of the National Coordinator |
| OSEHRA | Open Source Electronic Health Record Alliance |
| REST | Representational State Transfer |
| RHEx | RESTful Health Exchange |
| SAML | Security Assertion Markup Language |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| VA | Veterans Affairs |

# References

|  |  |
| --- | --- |
| [1]  | J. Kemp, S. Cantor, P. Mishra, R. Philpott and E. Maler, "Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0," 15 March 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf. [Accessed June 2014]. |
| [2]  | D. Hardt, "The OAuth 2.0 Authorization Framework," October 2012. [Online]. Available: http://tools.ietf.org/html/rfc6749. [Accessed May 2014]. |
| [3]  | M. Jones and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage," October 2012. [Online]. Available: http://tools.ietf.org/html/rfc6750. [Accessed June 2014]. |
| [4]  | S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," March 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2119.txt. [Accessed June 2014]. |
| [5]  | N. Sakimura, "OpenID Connect Core 1.0," 25 February 2014. [Online]. Available: http://openid.net/specs/openid-connect-core-1\_0.html. [Accessed May 2014]. |
| [6]  | M. Jones, "JSON Web Algorithms (JWA)," 30 December 2014. [Online]. Available: https://tools.ietf.org/html/draft-ietf-jose-json-web-algorithms-39. [Accessed January 2015]. |
| [7]  | The MITRE Corporation, "Secure RESTful Interfaces: Draft Profiles for the Use of OAuth 2.0: Draft Version 1.4," January 2015. |
| [8]  | Federal Identity, Credential, and Access Management Trust Framework Solutions, "Identity Scheme and Protocol Profile Adoption Process," 7 February 2014. [Online]. Available: http://www.idmanagement.gov/sites/default/files/documents/FICAM\_TFS\_SchemeProfile\_Adoption\_Process.pdf. [Accessed June 2016]. |
| [9]  | N. Sakimura, J. Bradley, M. Jones and E. Jay, "OpenID Connect Discovery 1.0," 25 February 2014. [Online]. Available: http://openid.net/specs/openid-connect-discovery-1\_0.html. [Accessed June 2014]. |
| [10]  | J. Richer, "OAuth Token Introspection," 23 December 2014. [Online]. Available: https://tools.ietf.org/html/draft-ietf-oauth-introspection-04. [Accessed January 2015]. |
| [11]  | T. Lodderstedt, S. Dronia and M. Scurtescu, "OAuth 2.0 Token Revocation," August 2013. [Online]. Available: http://tools.ietf.org/html/rfc7009. [Accessed June 2014]. |