|  |
| --- |
| **[TOC](https://openid.net/specs/openid-financial-api-part-1-1_0.html" \l "toc)** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Final | N. Sakimura | |  | Nat Consulting | |  | J. Bradley | |  | Yubico | |  | E. Jay | |  | Illumila | |  | March 12, 2021 | |

**Financial-grade API Security Profile 1.0 - Part 1: Baseline**

**Foreword**

The OpenID Foundation (OIDF) promotes, protects and nurtures the OpenID community and technologies. As a non-profit international standardizing body, it is comprised by over 160 participating entities (workgroup participants). The work of preparing implementer drafts and final international standards is carried out through OIDF workgroups in accordance with the OpenID Process. Participants interested in a subject for which a workgroup has been established have the right to be represented in that workgroup. International organizations, governmental and non-governmental, in liaison with OIDF, also take part in the work. OIDF collaborates closely with other standardizing bodies in the related fields.

Final drafts adopted by the Workgroup through consensus are circulated publicly for the public review for 60 days and for the OIDF members for voting. Publication as an OIDF Standard requires approval by at least 50 % of the members casting a vote. There is a possibility that some of the elements of this document may be the subject to patent rights. OIDF shall not be held responsible for identifying any or all such patent rights.

Financial-grade API Security Profile 1.0 consists of the following parts:

* Financial-grade API Security Profile 1.0 - Part 1: Baseline
* [**Financial-grade API Security Profile 1.0 - Part 2: Advanced**](https://openid.net/specs/openid-financial-api-part-2-1_0.html)

These parts are intended to be used with [**RFC6749**](https://tools.ietf.org/html/rfc6749), [**RFC6750**](https://tools.ietf.org/html/rfc6750), [**RFC7636**](https://tools.ietf.org/html/rfc7636), and [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html).

**Introduction**

The Financial-grade API is a highly secured OAuth profile that aims to provide specific implementation guidelines for security and interoperability. The Financial-grade API security profile can be applied to APIs in any market area that requires a higher level of security than provided by standard [**OAuth**](https://tools.ietf.org/html/rfc6749) or [**OpenID Connect**](https://openid.net/specs/openid-connect-core-1_0.html). Among other security enhancements, this specification provides a secure alternative to screen scraping. Screen scraping accesses user's data and functions by impresonating a user through password sharing. This brittle, inefficient, and insecure practice creates security vulnerabilities which require financial institutions to allow what appears to be an automated attack against their applications.

This document is Part 1 of FAPI Security Profile 1.0. It specifies a baseline security profile of OAuth that is suitable for protecting APIs with a moderate inherent risk. Importantly, this profile does not provide non-repudiation (signing of authorization requests and responses) and sender-constrained access tokens. If such features or a higher level of security is desired, the use of [**Financial-grade API Security Profile 1.0 - Part 2: Advanced**](https://openid.net/specs/openid-financial-api-part-2-1_0.html) is recommended.

Although it is possible to code an OpenID Provider and Relying Party from first principles using this specification, the main audience for this specification is parties who already have a certified implementation of OpenID Connect and want to achieve a higher level of security. Implementers are encouraged to understand the security considerations contained in Section 7.6 before embarking on a 'from scratch' implementation.

**Notational Conventions**

The key words "shall", "shall not", "should", "should not", "may", and "can" in this document are to be interpreted as described in [**ISO Directive Part 2**](https://www.iso.org/sites/directives/current/part2/index.xhtml). These key words are not used as dictionary terms such that any occurrence of them shall be interpreted as key words and are not to be interpreted with their natural language meanings.

**Table of Contents**

[**1.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#scope)**Scope**[**2.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#normative-references)**Normative references**[**3.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#terms-and-definitions)**Terms and definitions**[**4.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#symbols-and-abbreviated-terms)**Symbols and abbreviated terms**[**5.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#baseline-security-profile)**Baseline security profile**[**5.1.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#introduction-1)**Introduction**[**5.2.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#baseline-security-provisions)**Baseline security provisions**[**5.2.1.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#introduction-2)**Introduction**[**5.2.2.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#authorization-server)**Authorization server**[**5.2.2.1.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#returning-authenticated-user-s-identifier)**Returning authenticated user's identifier**[**5.2.2.2.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#client-requesting-openid-scope)**Client requesting openid scope**[**5.2.2.3.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#clients-not-requesting-openid-scope)**Clients not requesting openid scope**[**5.2.3.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#public-client)**Public client**[**5.2.4.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#confidential-client)**Confidential client**[**6.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#accessing-protected-resources)**Accessing Protected Resources**[**6.1.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#introduction-3)**Introduction**[**6.2.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#baseline-access-provisions)**Baseline access provisions**[**6.2.1.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#protected-resources-provisions)**Protected resources provisions**[**6.2.2.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#client-provisions)**Client provisions**[**7.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#security-considerations)**Security considerations**[**7.1.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#tls-and-dnssec-considerations)**TLS and DNSSEC considerations**[**7.2.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#message-source-authentication-failure)**Message source authentication failure**[**7.3.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#message-integrity-protection-failure)**Message integrity protection failure**[**7.4.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#message-containment-failure)**Message containment failure**[**7.4.1.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#authorization-request-and-response)**Authorization request and response**[**7.4.2.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#token-request-and-response)**Token request and response**[**7.4.3.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#resource-request-and-response)**Resource request and response**[**7.5.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#native-apps)**Native Apps**[**7.6.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#incomplete-or-incorrect-implementations-of-the-specifications)**Incomplete or incorrect implementations of the specifications**[**7.7.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#discovery-multiple-brands)**Discovery & Multiple Brands**[**8.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#privacy-considerations)**Privacy considerations**[**8.1.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#introduction-4)**Introduction**[**9.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#acknowledgement)**Acknowledgement**[**10.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#bibliography)**Bibliography**[**Appendix A.**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#copyright-notice-license)**Copyright notice & license**[**§**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#rfc.authors)**Authors' Addresses**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**1.  Scope**

This document specifies the method for an application to:

* obtain OAuth tokens in a moderately secure manner for access to protected data;
* use OpenID Connect (OIDC) to identify the customer (user); and
* use tokens to access REST APIs in a moderately secure manner.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**2.  Normative references**

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applied. For undated references, the latest edition of the referenced document (including any amendments) applies.

[**ISODIR2**](https://www.iso.org/sites/directives/current/part2/index.xhtml) - ISO/IEC Directives Part 2

[**RFC4122**](https://tools.ietf.org/html/rfc4122) - A Universally Unique IDentifier (UUID) URN Namespace

[**RFC6749**](https://tools.ietf.org/html/rfc6749) - The OAuth 2.0 Authorization Framework

[**RFC6750**](https://tools.ietf.org/html/rfc6750) - The OAuth 2.0 Authorization Framework: Bearer Token Usage

[**RFC7636**](https://tools.ietf.org/html/rfc7636) - Proof Key for Code Exchange by OAuth Public Clients

[**RFC6125**](https://tools.ietf.org/html/rfc6125) - Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)

[**BCP212**](https://tools.ietf.org/html/bcp212) - OAuth 2.0 for Native Apps

[**RFC6819**](https://tools.ietf.org/html/rfc6819) - OAuth 2.0 Threat Model and Security Considerations

[**BCP195**](https://tools.ietf.org/html/bcp195) - Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)

[**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html) - OpenID Connect Core 1.0 incorporating errata set 1

[**X.1254**](https://www.itu.int/rec/T-REC-X.1254) - Entity authentication assurance framework

[**MTLS**](https://tools.ietf.org/html/rfc8705) - OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens

[**RFC8414**](https://tools.ietf.org/html/rfc8414) - OAuth 2.0 Authorization Server Metadata

[**OIDD**](http://openid.net/specs/openid-connect-discovery-1_0.html) - OpenID Connect Discovery 1.0 incorporating errata set 1

[**RFC7231**](https://tools.ietf.org/html/rfc7231) - Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**3.  Terms and definitions**

For the purpose of this document, the terms defined in [**RFC6749**](https://tools.ietf.org/html/rfc6749), [**RFC6750**](https://tools.ietf.org/html/rfc6750), [**RFC7636**](https://tools.ietf.org/html/rfc7636), [**OpenID Connect Core**](https://openid.net/specs/openid-connect-core-1_0.html) apply.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**4.  Symbols and abbreviated terms**

**API** – Application Programming Interface

**CSRF** - Cross Site Request Forgery

**FAPI** - Financial-grade API

**HTTP** – Hyper Text Transfer Protocol

**REST** – Representational State Transfer

**TLS** – Transport Layer Security

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**5.  Baseline security profile**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**5.1.  Introduction**

The OIDF Financial-grade API (FAPI) security profile specifies security requirements for API resources protected by the OAuth 2.0 Authorization Framework that consists of [**RFC6749**](https://tools.ietf.org/html/rfc6749), [**RFC6750**](https://tools.ietf.org/html/rfc6750), [**RFC7636**](https://tools.ietf.org/html/rfc7636), and other specifications.

FAPI Security Profile 1.0 - Part 1: Baseline and [**Part 2: Advanced**](https://openid.net/specs/openid-financial-api-part-2-1_0.html) specify different levels of security. The characteristics required of the tokens are different and the methods to obtain tokens are explained separately. This document specifies the baseline security provisions.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**5.2.  Baseline security provisions**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**5.2.1.  Introduction**

Some APIs, such as ones that provide potentially sensitive information, require a greater level of protection than basic [**RFC6749**](https://tools.ietf.org/html/rfc6749) requires. FAPI provides such greater protection.

As a profile of the OAuth 2.0 Authorization Framework, this document mandates the following to the baseline profile of the FAPI Security Profile 1.0.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**5.2.2.  Authorization server**

The authorization server

1. shall support confidential clients;
2. should support public clients;
3. shall provide a client secret that adheres to the requirements in Section 16.19 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html) if a symmetric key is used;
4. shall authenticate the confidential client using one of the following methods:
   1. Mutual TLS for OAuth Client Authentication as specified in Section 2 of [**MTLS**](https://tools.ietf.org/html/rfc8705), or
   2. client\_secret\_jwt or private\_key\_jwt as specified in Section 9 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html);
5. shall require and use a key of size 2048 bits or larger for RSA algorithms;
6. shall require and use a key of size 160 bits or larger for elliptic curve algorithms;
7. shall require [**RFC7636**](https://tools.ietf.org/html/rfc7636) with S256 as the code challenge method;
8. shall require redirect URIs to be pre-registered;
9. shall require the redirect\_uri in the authorization request;
10. shall require the value of redirect\_uri to exactly match one of the pre-registered redirect URIs;
11. shall require user authentication to an appropriate Level of Assurance for the operations the client will be authorized to perform on behalf of the user;
12. shall require explicit approval by the user to authorize the requested scope if it has not been previously authorized;
13. shall reject an authorization code (Section 1.3.1 of [**RFC6749**](https://tools.ietf.org/html/rfc6749)) if it has been previously used;
14. shall return token responses that conform to Section 4.1.4 of [**RFC6749**](https://tools.ietf.org/html/rfc6749);
15. shall return the list of granted scopes with the issued access token if the request was passed in the front channel and was not integrity protected;
16. shall provide non-guessable access tokens, authorization codes, and refresh token (where applicable), with sufficient entropy such that the probability of an attacker guessing the generated token is computationally infeasible as per [**RFC6749**](https://tools.ietf.org/html/rfc6749) Section 10.10;
17. should clearly identify the details of the grant to the user during authorization as in 16.18 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html);
18. should provide a mechanism for the end-user to revoke access tokens and refresh tokens granted to a client as in 16.18 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html);
19. shall return an invalid\_client error as defined in 5.2 of [**RFC6749**](https://tools.ietf.org/html/rfc6749) when mis-matched client identifiers were provided through the client authentication methods that permits sending the client identifier in more than one way;
20. shall require redirect URIs to use the https scheme;
21. should issue access tokens with a lifetime of under 10 minutes unless the tokens are sender-constrained; and
22. shall support [**OIDD**](http://openid.net/specs/openid-connect-discovery-1_0.html), may support [**RFC8414**](https://tools.ietf.org/html/rfc8414) and shall not distribute discovery metadata (such as the authorization endpoint) by any other means.  
    **NOTE**: The use of refresh tokens instead of long-lived access tokens for both public and confidential clients is recommended.  
    **NOTE**: The Financial-grade API Security Profile 1.0 server may limit the scopes for the purpose of not implementing certain APIs.  
    **NOTE**: Clients are expected to treat access tokens as opaque strings and replay them as is. Authorization servers can issue unstructured or structured access tokens (for example, a signed JWT).  
    **NOTE**: The requirement to return the list of granted scopes allows clients to detect when the authorization request was modified to include different scopes. Servers must still return the granted scopes if they are different from those requested.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**5.2.2.1.  Returning authenticated user's identifier**

Further, if it is desired to provide the authenticated user's identifier to the client in the token response, the authorization server:

1. shall support the authentication request as in Section 3.1.2.1 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html);
2. shall perform the authentication request verification as in Section 3.1.2.2 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html);
3. shall authenticate the user as in Section 3.1.2.2 and 3.1.2.3 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html);
4. shall provide the authentication response as in Section 3.1.2.4 and 3.1.2.5 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html) depending on the outcome of the authentication;
5. shall perform the token request verification as in Section 3.1.3.2 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html); and
6. shall issue an ID Token in the token response when openid was included in the requested scope as in Section 3.1.3.3 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html) with its sub value corresponding to the authenticated user and optional acr value in ID Token.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**5.2.2.2.  Client requesting openid scope**

If the client requests the openid scope, the authorization server

1. shall require the nonce parameter defined in Section 3.1.2.1 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html) in the authentication request.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**5.2.2.3.  Clients not requesting openid scope**

If the client does not requests the openid scope, the authorization server

1. shall require the state parameter defined in Section 4.1.1 of [**RFC6749**](https://tools.ietf.org/html/rfc6749).

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**5.2.3.  Public client**

A public client

1. shall support [**RFC7636**](https://tools.ietf.org/html/rfc7636);
2. shall use S256 as the code challenge method for the [**RFC7636**](https://tools.ietf.org/html/rfc7636);
3. shall use separate and distinct redirect URI for each authorization server that it talks to;
4. shall store the redirect URI value in the resource owner's user-agents (such as browser) session and compare it with the redirect URI that the authorization response was received at, where, if the URIs do not match, the client shall terminate the process with error;
5. (withdrawn); and
6. shall implement an effective CSRF protection.  
   Further, if it is desired to obtain a persistent identifier of the authenticated user, then the public client
7. shall include openid in the scope value; and
8. shall include the nonce parameter defined in Section 3.1.2.1 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html) in the authentication request.  
   If openid is not in the scope value, then the public client
9. shall include the state parameter defined in Section 4.1.1 of [**RFC6749**](https://tools.ietf.org/html/rfc6749);
10. shall verify that the scope received in the token response is either an exact match, or contains a subset of the scope sent in the authorization request; and
11. shall only use Authorization Server metadata obtained from the metadata document published by the Authorization Server at its well known endpoint as defined in [**OIDD**](http://openid.net/specs/openid-connect-discovery-1_0.html) or [**RFC8414**](https://tools.ietf.org/html/rfc8414).  
    **NOTE**: Adherence to [**RFC7636**](https://tools.ietf.org/html/rfc7636) means that the token request includes code\_verifier parameter in the request.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**5.2.4.  Confidential client**

In addition to the provisions for a public client, a confidential client

1. shall support the following methods to authenticate against the token endpoint:
   1. Mutual TLS for OAuth Client Authentication as specified in Section 2 of [**MTLS**](https://tools.ietf.org/html/rfc8705), and
   2. client\_secret\_jwt or private\_key\_jwt as specified in Section 9 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html);
2. shall use RSA keys with a minimum 2048 bits if using RSA cryptography;
3. shall use elliptic curve keys with a minimum of 160 bits if using Elliptic Curve cryptography; and
4. shall verify that its client secret has a minimum of 128 bits if using symmetric key cryptography.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**6.  Accessing Protected Resources**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**6.1.  Introduction**

The FAPI endpoints are OAuth 2.0 protected resource endpoints that return protected information for the resource owner associated with the submitted access token.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**6.2.  Baseline access provisions**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**6.2.1.  Protected resources provisions**

The resource server with the FAPI endpoints

1. shall support the use of the HTTP GET method as in Section 4.3.1 of [**RFC7231**](https://tools.ietf.org/html/rfc7231);
2. shall accept access tokens in the HTTP header as in Section 2.1 of OAuth 2.0 Bearer Token Usage [**RFC6750**](https://tools.ietf.org/html/rfc6750);
3. shall not accept access tokens in the query parameters stated in Section 2.3 of OAuth 2.0 Bearer Token Usage [**RFC6750**](https://tools.ietf.org/html/rfc6750);
4. shall verify that the access token is neither expired nor revoked;
5. shall verify that the scope associated with the access token authorizes access to the resource it is representing;
6. shall identify the associated entity to the access token;
7. shall only return the resource identified by the combination of the entity implicit in the access and the granted scope and otherwise return errors as in Section 3.1 of [**RFC6750**](https://tools.ietf.org/html/rfc6750);
8. shall encode the response in UTF-8 if applicable;
9. shall send the Content-type HTTP header Content-Type: application/json if applicable;
10. shall send the server date in HTTP Date header as in Section 7.1.1.2 of [**RFC7231**](https://tools.ietf.org/html/rfc7231);
11. shall set the response header x-fapi-interaction-id to the value received from the corresponding FAPI client request header or to a [**RFC4122**](https://tools.ietf.org/html/rfc4122) UUID value if the request header was not provided to track the interaction, e.g., x-fapi-interaction-id: c770aef3-6784-41f7-8e0e-ff5f97bddb3a;
12. shall log the value of x-fapi-interaction-id in the log entry; and
13. shall not reject requests with a x-fapi-customer-ip-address header containing a valid IPv4 or IPv6 address.  
    **NOTE**: While this document does not specify the exact method to obtain the entity associated with the access token and the granted scope, the protected resource can use OAuth Token Introspection [**RFC7662**](https://tools.ietf.org/html/rfc7662).  
    Further, the resource server
14. should support the use of Cross Origin Resource Sharing (CORS) [CORS] and or other methods as appropriate to enable JavaScript clients to access the endpoint if it decides to provide access to JavaScript clients.  
    **NOTE**: Providing access to JavaScript clients has other security implications. Before supporting those clients [**RFC6819**](https://tools.ietf.org/html/rfc6819) should be consulted.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**6.2.2.  Client provisions**

The client supporting this document

1. shall send access tokens in the HTTP header as in Section 2.1 of OAuth 2.0 Bearer Token Usage [**RFC6750**](https://tools.ietf.org/html/rfc6750); and
2. (withdrawn);  
   Further, the client
3. may send the last time the customer logged into the client in the x-fapi-auth-date header where the value is supplied as a HTTP-date as in Section 7.1.1.1 of [**RFC7231**](https://tools.ietf.org/html/rfc7231), e.g., x-fapi-auth-date: Tue, 11 Sep 2012 19:43:31 GMT;
4. may send the customer’s IP address if this data is available in the x-fapi-customer-ip-address header, e.g., x-fapi-customer-ip-address: 2001:DB8::1893:25c8:1946 or x-fapi-customer-ip-address: 198.51.100.119; and
5. may send the x-fapi-interaction-id request header, in which case the value shall be a RFC4122 UUID to the server to help correlate log entries between client and server, e.g., x-fapi-interaction-id: c770aef3-6784-41f7-8e0e-ff5f97bddb3a.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.  Security considerations**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.1.  TLS and DNSSEC considerations**

As confidential information is being exchanged, all interactions shall be encrypted with TLS (HTTPS).

The recommendations for Secure Use of Transport Layer Security in [**BCP195**](https://tools.ietf.org/html/bcp195) shall be followed, with the following additional requirements:

1. TLS version 1.2 or later shall be used for all communications.
2. A TLS server certificate check shall be performed, as per [**RFC6125**](https://tools.ietf.org/html/rfc6125).

Endpoints for the use by web browsers should use mechanisms to ensure that connections cannot be downgraded using TLS Stripping attacks. A preloaded HTTP Strict Transport Security policy (see [**PRELOAD**](https://hstspreload.org/) and [**RFC6797**](https://tools.ietf.org/html/rfc6797)) can be used for this purpose. Some top-level domains, like .bank and .insurance, have set such a policy and therefore protect all second-level domains below them.

For a comprehensive protection against network attackers, all endpoints should additionally use DNSSEC to protect against DNS spoofing attacks that can lead to the issuance of rogue domain-validated TLS certificates.

**NOTE**: Even if an endpoint uses only organization validated (OV) or extended validation (EV) TLS certificates, rogue domain-validated certificates can be used to impersonate the endpoints and conduct man-in-the-middle attacks. CAA records [**RFC8659**](https://tools.ietf.org/html/rfc8659) can help to mitigate this risk.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.2.  Message source authentication failure**

Authorization request and response are not authenticated. For higher risk scenarios, they should be authenticated. See [**Financial-grade API Security Profile 1.0 - Part 2: Advanced**](https://openid.net/specs/openid-financial-api-part-2-1_0.html), which uses request objects to achieve the message source authentication.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.3.  Message integrity protection failure**

The authorization request does not have message integrity protection and hence request tampering and parameter injection are possible. Where such protection is desired, [**Financial-grade API Security Profile 1.0 - Part 2: Advanced**](https://openid.net/specs/openid-financial-api-part-2-1_0.html) should be used.

The response is integrity protected when the ID Token is returned from the authorization endpoint.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.4.  Message containment failure**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.4.1.  Authorization request and response**

In this document, the authorization request is not encrypted. Thus, it is possible to leak the information contained if the web browser is compromised.

Authorization response can be encrypted as ID Token can be encrypted.

It is possible to leak the information through the logs if the parameters were recorded in the logs and the access to the logs are compromised. Strict access control to the logs in such cases should be enforced.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.4.2.  Token request and response**

It is possible to leak information through the logs if the parameters were recorded in the logs and the access to the logs are compromised. Strict access control to the logs in such cases should be enforced.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.4.3.  Resource request and response**

Care should be taken so that the sensitive data will not be leaked through the referrer.

If the access token is a bearer token, it is possible to exercise the stolen token. Since the access token can be used against multiple URIs, the risk of leaking is much larger than the refresh token, which is used only against the token endpoint. Thus, the lifetime of the access token should be much shorter than that of the refresh token. Refer to Section 16.18 of [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html) for more discussion on the lifetimes of access tokens and refresh tokens.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.5.  Native Apps**

When native apps are used as either public clients, dynamically registered confidential clients or user-agents receiving the authorization response for a server based confidential client, the recommendations for OAuth 2.0 for Native Apps in [**BCP212**](https://tools.ietf.org/html/bcp212) shall be followed, with the following additional requirements:

When registering redirect URIs, authorization servers

1. shall not support "Private-Use URI Scheme Redirection"; and
2. shall not support "Loopback Interface Redirection".

These requirements mean that FAPI Security Profile 1.0 compliant implementations can only support native apps through the use of "Claimed https Scheme URI Redirection".

**NOTE**: Nothing in this document seeks to disallow fixed urls in the form https://localhost:port-number/callback, as these are particularly useful in non-production systems or in clients used in development, to facilitate faster and easier development.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.6.  Incomplete or incorrect implementations of the specifications**

To achieve the full security benefits, it is important the implementation of this specification, and the underlying OpenID Connect and OAuth specifications, are both complete and correct.

The OpenID Foundation provides tools that can be used to confirm that an implementation is correct:

[**https://openid.net/certification/**](https://openid.net/certification/)

The OpenID Foundation maintains a list of certified implementations:

[**https://openid.net/developers/certified/**](https://openid.net/developers/certified/)

Deployments that use this specification should use a certified implementation.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**7.7.  Discovery & Multiple Brands**

Organizations who need to support multiple "brands" with individual authorization endpoints from a single Authorization Server deployment shall use a separate issuer per brand. This can be achieved either at the domain level (e.g. https://brand-a.auth.example.com and https://brand-b.auth.example.com) or with different paths (e.g. https://auth.example.com/brand-a and https://auth.example.com/brand-b)

As stated in 5.2.2-22 Clients shall only use metadata values obtained via metadata documents as defined in [**OIDD**](http://openid.net/specs/openid-connect-discovery-1_0.html). Communicating metadata through other means (e.g. via email) opens up a social engineering attack vector.

Note that the requirement to use [**OIDD**](http://openid.net/specs/openid-connect-discovery-1_0.html) is not a requirement to support Dynamic Client Registration.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**8.  Privacy considerations**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**8.1.  Introduction**

There are many factors to be considered in terms of privacy when implementing this document. However, since this document is a profile of OAuth and OpenID Connect, all of them are generic and apply to OAuth or OpenID Connect and are not specific to this document. Implementers are advised to perform a thorough privacy impact assessment and manage identified risks appropriately.

**NOTE**: Implementers can consult documents like [**ISO29100**](http://standards.iso.org/ittf/PubliclyAvailableStandards/c045123_ISO_IEC_29100_2011.zip) and [ISO29134] for this purpose.

Privacy threats to OAuth and OpenID Connect implementations include the following:

* (Inappropriate privacy notice) A privacy notice provided at a policy\_url or by other means can be inappropriate.
* (Inadequate choice) Providing a consent screen without adequate choices does not form consent.
* (Misuse of data) An AS, RS or Client can potentially use the data not according to the purpose that was agreed.
* (Collection minimization violation) A client asking for more data than it absolutely needs to fulfil the purpose is violating the collection minimization principle.
* (Unsolicited personal data from the Resource) Some bad resource server implementations may return more data than was requested. If the data is personal data, then this would be a violation of privacy principles.
* (Data minimization violation) Any process that is processing more data than it needs is violating the data minimization principle.
* (RP tracking by AS/OP) AS/OP identifying what data is being provided to which Client/RP.
* (User tracking by RPs) Two or more RPs correlating access tokens or ID Tokens to track users.
* (RP misidentification by User at AS) User misunderstands who the RP is due to a confusing representation of the RP at the AS's authorization page.
* (Mismatch between User’s understanding or what RP is displaying to a user and the actual authorization request) To enhance the trust of the ecosystem, best practice is for the AS to make clear what is included in the authorization request (for example, what data will be released to the RP).
* (Attacker observing personal data in authorization request) Authorization request might contain personal data. This can be observed by an attacker.
* (Attacker observing personal data in authorization endpoint response) In some frameworks, even state is deemed personal data. This can be observed by an attacker through various means.
* (Data leak from AS) AS stores personal data. If AS is compromised, these data can leak or be modified.
* (Data leak from Resource) Some resource servers store personal data. If a resource server is compromised, these data can leak or be modified.
* (Data leak from Clients) Some clients store personal data. If the client is compromised, these data can leak or be modified.

These threats can be mitigated by choosing appropriate options in OAuth or OpenID, or by introducing some operational rules. For example, "Attacker observing personal data in authorization request" can be mitigated by either using authorization request by reference using request\_uri or by encrypting the request object. Similarly, "Attacker observing personal data in authorization endpoint response" can be mitigated by encrypting the ID Token or JARM response.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**9.  Acknowledgement**

The following people contributed to this document:

* Nat Sakimura (NAT Consulting) -- Chair, Editor
* Anoop Saxena (Intuit) -- Co-chair, FS-ISAC Liaison
* Anthony Nadalin (Microsoft) -- Co-chair, SC 27 Liaison
* Edmund Jay (Illumila) -- Co-editor
* Dave Tonge (Moneyhub) -- Co-chair, UK Implementation Entity Liaison
* Paul A. Grassi (NIST) -- X9 Liaison
* Joseph Heenan (Authlete)
* Sascha H. Preibisch (CA)
* Henrik Biering (Peercraft)
* Anton Taborszky (Deutsche Telecom)
* John Bradley (Yubico)
* Tom Jones (Independent)
* Axel Nennker (Deutsche Telekom)
* Daniel Fett (yes.com)
* Torsten Lodderstedt (yes.com)
* Ralph Bragg (Raidiam)
* Brian Campbell (Ping Identity)
* Dima Postnikov (Independent)
* Stuart Low (Biza.io)
* Takahiko Kawasaki (Authlete)
* Vladimir Dzhuvinov (Connect2Id)
* Chris Michael (Open Banking)
* Freddi Gyara (Open Banking)
* Rob Otto (Ping Identity)
* Francis Pouatcha (adorsys)
* Kosuke Koiwai (KDDI)
* Bjorn Hjelm (Verizon)
* Lukasz Jaromin (Cloudentity)
* James Manger

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**10.  Bibliography**

* [**Part2**](https://openid.net/specs/openid-financial-api-part-2-1_0.html) Financial-grade API Security Profile 1.0 - Part 2: Advanced
* [**ISODIR2**](https://www.iso.org/sites/directives/current/part2/index.xhtml) ISO/IEC Directives Part 2
* [**ISO29100**](http://standards.iso.org/ittf/PubliclyAvailableStandards/c045123_ISO_IEC_29100_2011.zip) ISO/IEC 29100 Information technology — Security techniques — Privacy framework
* [ISO29134] ISO/IEC 29134 Information technology — Security techniques — Guidelines for privacy impact assessment
* [**RFC4122**](https://tools.ietf.org/html/rfc4122) A Universally Unique IDentifier (UUID) URN Namespace
* [**RFC6749**](https://tools.ietf.org/html/rfc6749) The OAuth 2.0 Authorization Framework
* [**RFC6750**](https://tools.ietf.org/html/rfc6750) The OAuth 2.0 Authorization Framework: Bearer Token Usage
* [**RFC6797**](https://tools.ietf.org/html/rfc6797) HTTP Strict Transport Security (HSTS)
* [**RFC7636**](https://tools.ietf.org/html/rfc7636) Proof Key for Code Exchange by OAuth Public Clients
* [**RFC7662**](https://tools.ietf.org/html/rfc7662) OAuth 2.0 Token Introspection
* [**RFC6125**](https://tools.ietf.org/html/rfc6125) Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)
* [**BCP212**](https://tools.ietf.org/html/bcp212) OAuth 2.0 for Native Apps
* [**RFC6819**](https://tools.ietf.org/html/rfc6819) OAuth 2.0 Threat Model and Security Considerations
* [**RFC8414**](https://tools.ietf.org/html/rfc8414) OAuth 2.0 Authorization Server Metadata
* [**RFC8659**](https://tools.ietf.org/html/rfc8659) DNS Certification Authority Authorization (CAA) Resource Record
* [**OIDD**](http://openid.net/specs/openid-connect-discovery-1_0.html) OpenID Connect Discovery 1.0 incorporating errata set 1
* [**BCP195**](https://tools.ietf.org/html/bcp195) Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)
* [**OIDC**](https://openid.net/specs/openid-connect-core-1_0.html) OpenID Connect Core 1.0 incorporating errata set 1
* [**X.1254**](https://www.itu.int/rec/T-REC-X.1254) Entity authentication assurance framework
* [**MTLS**](https://tools.ietf.org/html/rfc8705) OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens
* [**PRELOAD**](https://hstspreload.org/) HSTS Preload List Submission

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**Appendix A.  Copyright notice & license**

Copyright (c) 2021 The OpenID Foundation.

The OpenID Foundation (OIDF) grants to any Contributor, developer, implementer, or other interested party a non-exclusive, royalty-free, worldwide copyright license to reproduce, prepare derivative works from, distribute, perform and display, this Implementers Draft or Final Specification solely for the purposes of (i) developing specifications, and (ii) implementing Implementers Drafts and Final Specifications based on such documents, provided that attribution be made to the OIDF as the source of the material, but that such attribution does not indicate an endorsement by the OIDF.

The technology described in this specification was made available from contributions from various sources, including members of the OpenID Foundation and others. Although the OpenID Foundation has taken steps to help ensure that the technology is available for distribution, it takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this specification or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any independent effort to identify any such rights. The OpenID Foundation and the contributors to this specification make no (and hereby expressly disclaim any) warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to this specification, and the entire risk as to implementing this specification is assumed by the implementer. The OpenID Intellectual Property Rights policy requires contributors to offer a patent promise not to assert certain patent claims against other contributors and against implementers. The OpenID Foundation invites any interested party to bring to its attention any copyrights, patents, patent applications, or other proprietary rights that may cover technology that may be required to practice this specification.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-1-1_0.html#toc) |

**Authors' Addresses**

|  |  |
| --- | --- |
|  | Nat Sakimura |
|  | Nat Consulting |
| **Email:** | [**nat@nat.consulting**](mailto:nat@nat.consulting) |
| **URI:** | [**http://nat.sakimura.org/**](http://nat.sakimura.org/) |
|  |  |
|  | John Bradley |
|  | Yubico |
| **Email:** | [**ve7jtb@ve7jtb.com**](mailto:ve7jtb@ve7jtb.com) |
| **URI:** | [**http://www.thread-safe.com/**](http://www.thread-safe.com/) |
|  |  |
|  | Illumila |
|  | Illumila |
| **Email:** | [**ejay@mgi1.com**](mailto:ejay@mgi1.com) |
| **URI:** | [**http://illumi.la/**](http://illumi.la/) |