|  |
| --- |
| **[TOC](https://openid.net/specs/openid-financial-api-part-2-1_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 2: Advanced**

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

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**](http://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**](http://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 2 of FAPI Security Profile 1.0 that specifies an advanced security profile of OAuth that is suitable to be used for protecting APIs with high inherent risk. Examples include APIs that give access to highly sensitive data or that can be used to trigger financial transactions (e.g., payment initiation). This document specifies the controls against attacks such as: authorization request tampering, authorization response tampering including code injection, state injection, and token request phishing. Additional details are available in the security considerations section.

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 8.7 before embarking on a 'from scratch' implementation.

**Notational Conventions**

The keywords "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 keywords are not used as dictionary terms such that any occurrence of them shall be interpreted as keywords and are not to be interpreted with their natural language meanings.

**Table of Contents**

[**1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#scope)**Scope**[**2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#normative-references)**Normative references**[**3.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#terms-and-definitions)**Terms and definitions**[**4.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#symbols-and-abbreviated-terms)**Symbols and Abbreviated terms**[**5.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#advanced-security-profile)**Advanced security profile**[**5.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#introduction-1)**Introduction**[**5.1.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#id-token-as-detached-signature)**ID Token as Detached Signature**[**5.1.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#jwt-secured-authorization-response-mode-for-oauth-2-0-jarm)**JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)**[**5.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#advanced-security-provisions)**Advanced security provisions**[**5.2.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#introduction-2)**Introduction**[**5.2.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#authorization-server)**Authorization server**[**5.2.2.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#id-token-as-detached-signature-1)**ID Token as detached signature**[**5.2.2.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#jarm)**JARM**[**5.2.3.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#confidential-client)**Confidential client**[**5.2.3.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#id-token-as-detached-signature-2)**ID Token as detached signature**[**5.2.3.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#jarm-1)**JARM**[**5.2.4.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#withdrawn)**(withdrawn)**[**5.2.5.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#withdrawn-1)**(withdrawn)**[**6.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#accessing-protected-resources-using-tokens)**Accessing protected resources (using tokens)**[**6.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#introduction-3)**Introduction**[**6.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#advanced-access-provisions)**Advanced access provisions**[**6.2.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#protected-resources-provisions)**Protected resources provisions**[**6.2.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#client-provisions)**Client provisions**[**7.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#withdrawn-2)**(Withdrawn)**[**8.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#security-considerations)**Security considerations**[**8.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#introduction-4)**Introduction**[**8.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#uncertainty-of-resource-server-handling-of-access-tokens)**Uncertainty of resource server handling of access tokens**[**8.3.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#attacks-using-weak-binding-of-authorization-server-endpoints)**Attacks using weak binding of authorization server endpoints**[**8.3.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#introduction-5)**Introduction**[**8.3.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#client-credential-and-authorization-code-phishing-at-token-endpoint)**Client credential and authorization code phishing at token endpoint**[**8.3.3.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#identity-provider-idp-mix-up-attack)**Identity provider (IdP) mix-up attack**[**8.3.4.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#removed)**(removed)**[**8.3.5.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#access-token-phishing)**Access token phishing**[**8.4.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#attacks-that-modify-authorization-requests-and-responses)**Attacks that modify authorization requests and responses**[**8.4.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#introduction-6)**Introduction**[**8.4.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#authorization-request-parameter-injection-attack)**Authorization request parameter injection attack**[**8.4.3.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#authorization-response-parameter-injection-attack)**Authorization response parameter injection attack**[**8.5.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#tls-considerations)**TLS considerations**[**8.6.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#algorithm-considerations)**Algorithm considerations**[**8.6.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#encryption-algorithm-considerations)**Encryption algorithm considerations**[**8.7.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#incomplete-or-incorrect-implementations-of-the-specifications)**Incomplete or incorrect implementations of the specifications**[**8.8.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#session-fixation)**Session Fixation**[**8.9.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#jwks-uris)**JWKS URIs**[**8.10.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#multiple-clients-sharing-the-same-key)**Multiple clients sharing the same key**[**8.11.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#duplicate-key-identifiers)**Duplicate Key Identifiers**[**9.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#privacy-considerations)**Privacy considerations**[**9.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#introduction-7)**Introduction**[**10.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#acknowledgement)**Acknowledgement**[**11.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#bibliography)**Bibliography**[**12.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#iana-considerations)**IANA Considerations**[**12.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#additions-to-jwt-claims-registry)**Additions to JWT Claims Registry**[**12.1.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#registry-contents)**Registry Contents**[**Appendix A.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#examples)**Examples**[**A.1.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#example-request-object)**Example request object**[**A.2.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#example-signed-id-token-for-authorization-endpoint-response)**Example signed id\_token for authorization endpoint response**[**A.3.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#example-signed-and-encrypted-id-token-for-authorization-endpoint-response)**Example signed and encrypted id\_token for authorization endpoint response**[**A.4.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#example-jarm-response)**Example JARM response**[**A.5.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#example-private-key-jwt-client-assertion)**Example private\_key\_jwt client assertion**[**Appendix B.**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#copyright-notice-license)**Copyright notice & license**[**§**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#rfc.authors)**Authors' Addresses**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**1.  Scope**

This part of the document specifies the method of

* applications to obtain the OAuth tokens in an appropriately secure manner for higher risk access to data;
* applications to use OpenID Connect to identify the customer; and
* using tokens to interact with the REST endpoints that provides protected data;

This document is applicable to higher risk use cases which includes commercial and investment banking and other similar industries.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_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

[**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

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

[**RFC7519**](https://tools.ietf.org/html/rfc7519) - JSON Web Token (JWT)

[**RFC7591**](https://tools.ietf.org/html/rfc7591) - OAuth 2.0 Dynamic Client Registration Protocol

[**RFC7592**](https://tools.ietf.org/html/rfc7592) - OAuth 2.0 Dynamic Client Registration Management Protocol

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

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

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

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

[**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md) - Financial-grade API: JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)

[**PAR**](https://tools.ietf.org/html/draft-ietf-oauth-par) - OAuth 2.0 Pushed Authorization Requests

[**JAR**](https://tools.ietf.org/html/draft-ietf-oauth-jwsreq) - OAuth 2.0 JWT Secured Authorization Request

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_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**](http://openid.net/specs/openid-connect-core-1_0.html) and [**ISO29100**](http://standards.iso.org/ittf/PubliclyAvailableStandards/c045123_ISO_IEC_29100_2011.zip) apply.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_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

**OIDF** - OpenID Foundation

**REST** – Representational State Transfer

**TLS** – Transport Layer Security

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.  Advanced security profile**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.1.  Introduction**

The OIDF Financial-grade API (FAPI) security profile specifies security requirements for high risk 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.

There are different levels of risks associated with access to these APIs. For example, read and write access to a bank API has a higher financial risk than read-only access. As such, the security profiles of the authorization framework protecting these APIs are also different.

This profile describes security provisions for the server and client that are appropriate for Financial-grade APIs by defining the measures to mitigate:

* attacks that leverage the weak binding of endpoints in [**RFC6749**](https://openid.net/specs/e.g.%20malicious%20endpoint%20attacks,%20IdP%20mix-up%20attacks), and
* attacks that modify authorization requests and responses unprotected in [**RFC6749**](https://tools.ietf.org/html/rfc6749).

This profile does not support public clients.

The following ways are specified to protect against modifications of authorization responses: Implementations can leverage OpenID Connect's Hybrid Flow that returns an ID Token in the authorization response or they can utilize the JWT Secured Authorization Response Mode for OAuth 2.0 ([**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md)) that returns and protects all authorization response parameters in a JWT.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.1.1.  ID Token as Detached Signature**

While the name ID Token (as used in the OpenID Connect Hybrid Flow) suggests that it is something that provides the identity of the resource owner (subject), it is not necessarily so. While it does identify the authorization server by including the issuer identifier, it is perfectly fine to have an ephemeral subject identifier. In this case, the ID Token acts as a detached signature of the issuer to the authorization response and it was an explicit design decision of OpenID Connect Core to make the ID Token act as a detached signature.

This document leverages this fact and protects the authorization response by including the hash of all of the unprotected response parameters, e.g. code and state, in the ID Token.

While the hash of the code is defined in [**OIDC**](http://openid.net/specs/openid-connect-core-1_0.html), the hash of the state is not defined. Thus this document defines it as follows.

**s\_hash**

State hash value. Its value is the base64url encoding of the left-most half of the hash of the octets of the ASCII representation of the state value, where the hash algorithm used is the hash algorithm used in the alg header parameter of the ID Token's JOSE header. For instance, if the alg is HS512, hash the state value with SHA-512, then take the left-most 256 bits and base64url encode them. The s\_hash value is a case sensitive string.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.1.2.  JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)**

An authorization server may protect authorization responses to clients using the "JWT Secured Authorization Response Mode" [**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md).

[**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md) allows a client to request that an authorization server encodes the authorization response (of any response type) in a JWT. It is an alternative to utilizing ID Tokens as detached signatures for providing financial-grade security on authorization responses and can be used with plain OAuth.

This specification facilitates use of [**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md) in conjunction with the response type code.

**NOTE:** [**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md) can be used to protect OpenID Connect authentication responses. In this case, the OpenID RP would use response type code, response mode jwt and scope openid. This means [**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md) protects the authentication response (instead of the ID Token) and the ID Token containing End-User Claims is obtained from the token endpoint. This facilitates privacy since no End-User Claims are sent through the front channel. It also provides decoupling of message protection and identity providing since a client (or RP) can basically use [**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md) to protect all authorization responses and turn on OpenID if needed (e.g. to log the user in).

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.2.  Advanced security provisions**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.2.1.  Introduction**

API resources may contain sensitive data and/or have increased security requirements. In order to fulfill different security needs, FAPI Security Profile 1.0 defines an advanced profile that is beyond the baseline security requirements defined in the [**Part 1: Baseline**](https://openid.net/specs/openid-financial-api-part-1-1_0.html) document.

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

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.2.2.  Authorization server**

The authorization server shall support the provisions specified in clause 5.2.2 of [**Financial-grade API Security Profile 1.0 - Part 1: Baseline**](https://openid.net/specs/openid-financial-api-part-1-1_0.html), with the exception that Section 5.2.2-7 (enforcement of [**RFC7636**](https://tools.ietf.org/html/rfc7636)) is not required.

In addition, the authorization server

1. shall require a JWS signed JWT request object passed by value with the request parameter or by reference with the request\_uri parameter;
2. shall require
   1. the response\_type value code id\_token, or
   2. the response\_type value code in conjunction with the response\_mode value jwt;
3. (moved to 5.2.2.1);
4. (moved to 5.2.2.1);
5. shall only issue sender-constrained access tokens;
6. shall support [**MTLS**](https://tools.ietf.org/html/rfc8705) as mechanism for constraining the legitimate senders of access tokens;
7. (withdrawn);
8. (moved to 5.2.2.1);
9. (moved to 5.2.2.1);
10. shall only use the parameters included in the signed request object passed via the request or request\_uri parameter;
11. may support the pushed authorization request endpoint as described in [**PAR**](https://tools.ietf.org/html/draft-ietf-oauth-par);
12. (withdrawn);
13. shall require the request object to contain an exp claim that has a lifetime of no longer than 60 minutes after the nbf claim;
14. shall authenticate the confidential client using one of the following methods (this overrides [**FAPI Security Profile 1.0 - Part 1: Baseline**](https://openid.net/specs/openid-financial-api-part-1-1_0.html) clause 5.2.2-4):
    1. tls\_client\_auth or self\_signed\_tls\_client\_auth as specified in section 2 of [**MTLS**](https://tools.ietf.org/html/rfc8705), or
    2. private\_key\_jwt as specified in section 9 of [**OIDC**](http://openid.net/specs/openid-connect-core-1_0.html);
15. shall require the aud claim in the request object to be, or to be an array containing, the OP's Issuer Identifier URL;
16. shall not support public clients;
17. shall require the request object to contain an nbf claim that is no longer than 60 minutes in the past; and
18. shall require [**PAR**](https://tools.ietf.org/html/draft-ietf-oauth-par) requests, if supported, to use PKCE ([**RFC7636**](https://tools.ietf.org/html/rfc7636)) with S256 as the code challenge method.

**NOTE:** MTLS is currently the only mechanism for sender-constrained access tokens that has been widely deployed. Future versions of this specification are likely to allow other mechanisms for sender-constrained access tokens.

**NOTE:** [**PAR**](https://tools.ietf.org/html/draft-ietf-oauth-par) does not present any additional security concerns that necessitated the requirement to use PKCE - the reason PKCE is not required in other cases is merely to be backwards compatible with earlier drafts of this standard.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.2.2.1.  ID Token as detached signature**

In addition, if the response\_type value code id\_token is used, the authorization server

1. shall support [**OIDC**](http://openid.net/specs/openid-connect-core-1_0.html);
2. shall support signed ID Tokens;
3. should support signed and encrypted ID Tokens;
4. shall return ID Token as a detached signature to the authorization response;
5. shall include state hash, s\_hash, in the ID Token to protect the state value if the client supplied a value for state. s\_hash may be omitted from the ID Token returned from the Token Endpoint when s\_hash is present in the ID Token returned from the Authorization Endpoint; and
6. should not return sensitive PII in the ID Token in the authorization response, but if it needs to, then it should encrypt the ID Token.

**NOTE:** The authorization server may return more claims in the ID Token from the token endpoint than in the one from the authorization response

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.2.2.2.  JARM**

In addition, if the response\_type value code is used in conjunction with the response\_mode value jwt, the authorization server

1. shall create JWT-secured authorization responses as specified in [**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md), Section 4.3.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.2.3.  Confidential client**

A confidential client shall support the provisions specified in clause 5.2.3 and 5.2.4 of [**Financial-grade API Security Profile 1.0 - Part 1: Baseline**](https://openid.net/specs/openid-financial-api-part-1-1_0.html), except for [**RFC7636**](https://tools.ietf.org/html/rfc7636) support.

In addition, the confidential client

1. shall support [**MTLS**](https://tools.ietf.org/html/rfc8705) as mechanism for sender-constrained access tokens;
2. shall include the request or request\_uri parameter as defined in Section 6 of [**OIDC**](http://openid.net/specs/openid-connect-core-1_0.html) in the authentication request;
3. shall ensure the Authorization Server has authenticated the user to an appropriate Level of Assurance for the client's intended purpose;
4. (moved to 5.2.3.1);
5. (withdrawn);
6. (withdrawn);
7. (moved 5.2.3.1);
8. shall send all parameters inside the authorization request's signed request object;
9. shall additionally send duplicates of the response\_type, client\_id, and scope parameters/values using the OAuth 2.0 request syntax as required by Section 6.1 of the OpenID Connect specification if not using [**PAR**](https://tools.ietf.org/html/draft-ietf-oauth-par);
10. shall send the aud claim in the request object as the OP's Issuer Identifier URL;
11. shall send an exp claim in the request object that has a lifetime of no longer than 60 minutes;
12. (moved to 5.2.3.1);
13. (moved to 5.2.3.1);
14. shall send a nbf claim in the request object;
15. shall use [**RFC7636**](https://tools.ietf.org/html/rfc7636) with S256 as the code challenge method if using [**PAR**](https://tools.ietf.org/html/draft-ietf-oauth-par); and
16. shall additionally send a duplicate of the client\_id parameter/value using the OAuth 2.0 request syntax to the authorization endpoint, as required by Section 5 of [**JAR**](https://tools.ietf.org/html/draft-ietf-oauth-jwsreq), if using [**PAR**](https://tools.ietf.org/html/draft-ietf-oauth-par).

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.2.3.1.  ID Token as detached signature**

In addition, if the response\_type value code id\_token is used, the client

1. shall include the value openid into the scope parameter in order to activate [**OIDC**](http://openid.net/specs/openid-connect-core-1_0.html) support;
2. shall require JWS signed ID Token be returned from endpoints;
3. shall verify that the authorization response was not tampered using ID Token as the detached signature;
4. shall verify that s\_hash value is equal to the value calculated from the state value in the authorization response in addition to all the requirements in 3.3.2.12 of [**OIDC**](http://openid.net/specs/openid-connect-core-1_0.html); and **NOTE:** This enables the client to verify that the authorization response was not tampered with, using the ID Token as a detached signature.
5. shall support both signed and signed & encrypted ID Tokens.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.2.3.2.  JARM**

In addition, if the response\_type value code is used in conjunction with the response\_mode value jwt, the client

1. shall verify the authorization responses as specified in [**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md), Section 4.4.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.2.4.  (withdrawn)**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**5.2.5.  (withdrawn)**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**6.  Accessing protected resources (using tokens)**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_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-2-1_0.html#toc) |

**6.2.  Advanced access provisions**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**6.2.1.  Protected resources provisions**

The protected resources supporting this document

1. shall support the provisions specified in clause 6.2.1 [**Financial-grade API Security Profile 1.0 - Part 1: Baseline**](https://openid.net/specs/openid-financial-api-part-1-1_0.html); and
2. shall adhere to the requirements in [**MTLS**](https://tools.ietf.org/html/rfc8705).

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**6.2.2.  Client provisions**

The client supporting this document shall support the provisions specified in clause 6.2.2 of [**Financial-grade API Security Profile 1.0 - Part 1: Baseline**](https://openid.net/specs/openid-financial-api-part-1-1_0.html).

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**7.  (Withdrawn)**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.  Security considerations**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.1.  Introduction**

As a profile of the OAuth 2.0 Authorization Framework, this specification references the security considerations defined in Section 10 of [**RFC6749**](https://tools.ietf.org/html/rfc6749), as well as [**RFC6819**](https://tools.ietf.org/html/rfc6819) - OAuth 2.0 Threat Model and Security Considerations, which details various threats and mitigations. The security of OAuth 2.0 has been proven formally - under certain assumptions - in [**OAUTHSEC**](https://arxiv.org/abs/1601.01229). A detailed security analysis of FAPI Security Profile 1.0 can be found in [**FAPISEC**](https://arxiv.org/abs/1901.11520).

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.2.  Uncertainty of resource server handling of access tokens**

There is no way that the client can find out whether the resource access was granted for a bearer or sender-constrained access token. The two differ in the risk profile and the client may want to differentiate them. The protected resources that conform to this document differentiate them. The protected resources that conform to this document shall not accept a bearer access token. They shall only support sender-constrained access tokens via [**MTLS**](https://tools.ietf.org/html/rfc8705).

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.3.  Attacks using weak binding of authorization server endpoints**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.3.1.  Introduction**

In [**RFC6749**](https://tools.ietf.org/html/rfc6749) and [**RFC6750**](https://tools.ietf.org/html/rfc6750), the endpoints that the authorization server offers are not tightly bound together. There is no notion of authorization server identifier (issuer identifier) and it is not indicated in the authorization response unless the client uses different redirection URI per authorization server. While it is assumed in the OAuth model, it is not explicitly spelled out and thus many clients use the same redirection URI for different authorization servers exposing an attack surface. Several attacks have been identified and the threats are explained in detail in [**RFC6819**](https://tools.ietf.org/html/rfc6819).

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.3.2.  Client credential and authorization code phishing at token endpoint**

In this attack, the client developer is socially engineered into believing that the token endpoint has changed to the URL that is controlled by the attacker. As a result, the client sends the code and the client secret to the attacker, which will be replayed subsequently.

When the FAPI Security Profile 1.0 client uses [**MTLS**](https://tools.ietf.org/html/rfc8705), the client's secret (the private key corresponding to its TLS certificate) is not exposed to the attacker, which therefore cannot authenticate towards the token endpoint of the authorization server.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.3.3.  Identity provider (IdP) mix-up attack**

In this attack, the client has registered multiple IdPs and one of them is a rogue IdP that returns the same client\_id that belongs to one of the honest IdPs. When a user clicks on a malicious link or visits a compromised site, an authorization request is sent to the rogue IdP. The rogue IdP then redirects the client to the honest IdP that has the same client\_id. If the user is already logged on at the honest IdP, then the authentication may be skipped and a code is generated and returned to the client. Since the client was interacting with the rogue IdP, the code is sent to the rogue IdP's token endpoint. At the point, the attacker has a valid code that can be exchanged for an access token at the honest IdP. See [**OAUTHSEC**](https://arxiv.org/abs/1601.01229) for a detailed description of the attack.

This attack is mitigated by the use of OpenID Connect Hybrid Flow in which the honest IdP's issuer identifier is included as the value of iss or [**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md) where the iss included in the response JWT. On receiving the authorization response, the client compares the iss value from the response with the issuer URL of the IdP it sent the authorization request to (the rogue IdP). The client detects the conflicting issuer values and aborts the transaction.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.3.4.  (removed)**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.3.5.  Access token phishing**

Various mechanisms in this specification aim at preventing access token phishing, e.g., the requirement of exactly matching redirect URIs and the restriction on response types that do not return access tokens in the front channel. As a second layer of defense, FAPI Security Profile 1.0 Advanced clients use [**MTLS**](https://tools.ietf.org/html/rfc8705) meaning the access token is bound to the client's TLS certificate. Even if an access token is phished, it cannot be used by the attacker. An attacker could try to trick a client under his control to make use of the access token as described in [FAPISEC] ("Cuckoo's Token Attack" and "Access Token Injection with ID Token Replay"), but these attacks additionally require a rogue AS or misconfigured token endpoint.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.4.  Attacks that modify authorization requests and responses**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.4.1.  Introduction**

In [**RFC6749**](https://tools.ietf.org/html/rfc6749) the authorization request and responses are not integrity protected. Thus, an attacker can modify them.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.4.2.  Authorization request parameter injection attack**

In [**RFC6749**](https://tools.ietf.org/html/rfc6749), the authorization request is sent as a query parameter. Although [**RFC6749**](https://tools.ietf.org/html/rfc6749) mandates the use of TLS, the TLS is terminated in the browser and thus not protected within the browser; as a result an attacker can tamper the authorization request and insert any parameter values.

The use of a request object or request\_uri in the authorization request will prevent tampering with the request parameters.

The IdP confusion attack reported in [**SoK: Single Sign-On Security – An Evaluation of OpenID Connect**](https://www.nds.rub.de/media/ei/veroeffentlichungen/2017/01/30/oidc-security.pdf) is an example of this kind of attack.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.4.3.  Authorization response parameter injection attack**

This attack occurs when the victim and attacker use the same relying party client. The attacker is somehow able to capture the authorization code and state from the victim's authorization response and uses them in his own authorization response.

This can be mitigated by using OpenID Connect Hybrid Flow where the c\_hash, at\_hash, and s\_hash can be used to verify the validity of the authorization code, access token, and state parameters. It can also be mitigated using [**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md) by verifying the integrity of the authorization response JWT.

The server can verify that the state is the same as what was stored in the browser session at the time of the authorization request.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.5.  TLS considerations**

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

Section 7.1 of [**Financial-grade API Security Profile 1.0 - Part 1: Baseline**](https://openid.net/specs/openid-financial-api-part-1-1_0.html) shall apply, with the following additional requirements:

1. For TLS versions below 1.3, only the following 4 cipher suites shall be permitted:
   * TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
   * TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
   * TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
   * TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
2. For the authorization\_endpoint, the authorization server MAY allow additional cipher suites that are permitted by the latest version of [**BCP195**](https://tools.ietf.org/html/bcp195), if necessary to allow sufficient interoperability with users' web browsers or are required by local regulations. **NOTE:** Permitted cipher suites are those that [**BCP195**](https://tools.ietf.org/html/bcp195) does not explicity say MUST NOT use.
3. When using the TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 or TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 cipher suites, key lengths of at least 2048 bits are required.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.6.  Algorithm considerations**

For JWS, both clients and authorization servers

1. shall use PS256 or ES256 algorithms;
2. should not use algorithms that use RSASSA-PKCS1-v1\_5 (e.g. RS256); and
3. shall not use none.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.6.1.  Encryption algorithm considerations**

For JWE, both clients and authorization servers

1. shall not use the RSA1\_5 algorithm.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.7.  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-2-1_0.html#toc) |

**8.8.  Session Fixation**

An attacker could prepare an authorization request URL and trick a victim into authorizing access to the requested resources, e.g. by sending the URL via e-Mail or utilizing it on a fake site.

OAuth 2.0 prevents this kind of attack since the process for obtaining the access token (code exchange, CSRF protection etc.) is designed in a way that the attacker will be unable to obtain and use the token as long as it does not control the victim's browser.

However, if the API allows execution of any privileged action in the course of the authorization process before the access token is issued, these controls are rendered ineffective. Implementers of this specification therefore shall ensure any action is executed using the access token issued by the authorization process.

For example, payments shall not be executed in the authorization process but after the Client has exchanged the authorization code for a token and sent an "execute payment" request with the access token to a protected endpoint.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.9.  JWKS URIs**

This profile requires both Clients and Authorization Servers to verify payloads with keys from the other party. The AS verifies request objects and private\_key\_jwt assertions. The Client verifies ID Tokens and authorization response JWTs. For AS's this profile strongly recommends the use of JWKS URI endpoints to distribute public keys. For Clients this profile recommends either the use of JWKS URI endpoints or the use of the jwks parameter in combination with [**RFC7591**](https://tools.ietf.org/html/rfc7591) and [**RFC7592**](https://tools.ietf.org/html/rfc7592).

The definition of the AS jwks\_uri can be found in [**RFC8414**](https://tools.ietf.org/html/rfc8414), while the definition of the Client jwks\_uri can be found in [**RFC7591**](https://tools.ietf.org/html/rfc7591).

In addition, this profile

1. requires that jwks\_uri endpoints shall be served over TLS;
2. recommends that JOSE headers for x5u and jku should not be used; and
3. recommends that the JWK set does not contain multiple keys with the same kid.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.10.  Multiple clients sharing the same key**

The use of [**MTLS**](https://tools.ietf.org/html/rfc8705) for client authentication and sender constraining access tokens brings significant security benefits over the use of shared secrets. However in some deployments the certificates used for [**MTLS**](https://tools.ietf.org/html/rfc8705) are issued by a Certificate Authority at an organization level rather than a client level. In such situations it may be common for an organization with multiple clients to use the same certificates (or certificates with the same DN) across clients. Implementers should be aware that such sharing means that a compromise of any one client, would result in a compromise of all clients sharing the same key.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**8.11.  Duplicate Key Identifiers**

JWK sets should not contain multiple keys with the same kid. However, to increase interoperability when there are multiple keys with the same kid, the verifier shall consider other JWK attributes, such as kty, use, alg, etc., when selecting the verification key for the particular JWS message. For example, the following algorithm could be used in selecting which key to use to verify a message signature:

1. find keys with a kid that matches the kid in the JOSE header;
2. if a single key is found, use that key;
3. if multiple keys are found, then the verifier should iterate through the keys until a key is found that has a matching alg, use, kty, or crv that corresponds to the message being verified.

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**9.  Privacy considerations**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**9.1.  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 applies to OAuth or OpenID Connect and 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) Clients 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 authorisation 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 (RS) store personal data. If a RS 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 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-2-1_0.html#toc) |

**10.  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-2-1_0.html#toc) |

**11.  Bibliography**

* [**Part1**](https://openid.net/specs/openid-financial-api-part-1-1_0.html) Financial-grade API Security Profile 1.0 - Part 1: Baseline
* [**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
* [ISO29184] ISO/IEC 29184 Information technology — Online privacy notices and consent
* [**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
* [**RFC6819**](https://tools.ietf.org/html/rfc6819) OAuth 2.0 Threat Model and Security Considerations
* [**RFC7519**](https://tools.ietf.org/html/rfc7519) JSON Web Token (JWT)
* [**RFC7591**](https://tools.ietf.org/html/rfc7591) OAuth 2.0 Dynamic Client Registration Protocol
* [**RFC7592**](https://tools.ietf.org/html/rfc7592) OAuth 2.0 Dynamic Client Registration Management Protocol
* [**RFC8414**](https://tools.ietf.org/html/rfc8414) OAuth 2.0 Authorization Server Metadata
* [**OIDC**](http://openid.net/specs/openid-connect-core-1_0.html) OpenID Connect Core 1.0 incorporating errata set 1
* [**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)
* [**MTLS**](https://tools.ietf.org/html/rfc8705) OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens
* [**JARM**](https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md) Financial-grade API: JWT Secured Authorization Response Mode for OAuth 2.0
* [**PAR**](https://tools.ietf.org/html/draft-ietf-oauth-par) OAuth 2.0 Pushed Authorization Requests
* [**JAR**](https://tools.ietf.org/html/draft-ietf-oauth-jwsreq) OAuth 2.0 JWT Secured Authorization Request
* [**SoK**](https://www.nds.ruhr-uni-bochum.de/media/ei/veroeffentlichungen/2017/01/30/oidc-security.pdf) Mainka, C., Mladenov, V., Schwenk, J., and T. Wich: SoK: Single Sign-On Security – An Evaluation of OpenID Connect
* [**FAPISEC**](https://arxiv.org/abs/1901.11520) Fett, D., Hosseyni, P., Kuesters, R.: An Extensive Formal Security Analysis of the OpenID Financial-grade API
* [**OAUTHSEC**](https://arxiv.org/abs/1601.01229) Fett, D., Kuesters, R., Schmitz, G.: A Comprehensive Formal Security Analysis of OAuth 2.0

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**12.  IANA Considerations**

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**12.1.  Additions to JWT Claims Registry**

This specification adds the following values to the "JSON Web Token Claims" registry established by [**RFC7519**](https://tools.ietf.org/html/rfc7519).

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**12.1.1.  Registry Contents**

* Claim name: s\_hash
* Claim Description: State hash value
* Change Controller: OpenID Foundation Financial-Grade API Working Group - openid-specs-fapi@lists.openid.net
* Reference: Section 5 of [[ this specification ]]

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**Appendix A.  Examples**

The following are non-normative examples of various objects compliant with this specification, with line wraps within values for display purposes only.

The examples signed by the client may be verified with the following JWK:

{

"kty": "RSA",

"e": "AQAB",

"use": "sig",

"kid": "client-2020-08-28",

"alg": "PS256",

"n": "i0Ybm4TJyErnD5FIs-6sgAdtP6fG631FXbe5gcOGYgn9aC2BS2h9Ah5cRGQpr3aLLVKCRWU6

HRfnGseUBOejo57vI-kgab2YsQJSwedAxvtKrIrJlgKn1gTXMNsz-NQd1LyLSV50qJVEy5l9RtsdDzOV

8\_kLCbzroEL3rc00iqVZBcQiYm8Bx4z0G8LYZ4oMJAG462Mf\_znJkKXsuSIH735xnSmx74CC8TOe6G-V

0Wi\_wVSJ9bHPphSki\_kWUtjVGcnyjYuQVE0LRj3qrGPAX9bsVKSqs8T9AM41TB9oV5Sjz5YhggwICvvC

CGwil9qhUoQRkeXtWuGCfvCSeTdawQ"

}

The examples signed by the server may be verified with the following JWK:

{

"kty": "RSA",

"e": "AQAB",

"use": "sig",

"kid": "server-2020-08-28",

"alg": "PS256",

"n": "pz6g0h7Cu63SHE8\_Ib4l3hft8XuptZ-Or7v\_j1EkCboyAEn\_ZCuBrQOmpUIoPKrA0JNWK\_fF

eZ2q1\_26Gvn3E4dQlcOWpiWkKmxAhYCWnNDv3urVgldDp\_kw0Dx2H8yn9tmFW28E\_WvrZRwHEF5Czigb

xlmFIrkniMHRzjyYQTHRU0gW3DRV9MrQQrmP71McvfLPeMBPPgsHgLo7KmUBDoUjsgnwgycEOWPm8MWJ

13dpTsVnoWNIFQqVNz1L5pRU3Uoknl0MGoE6v0M9lfgQgzxIX9gSB1VGp5zZRcsnZGU3MFpwBhOWwiCU

wqztoX0H5P0g7OWocspHrDn6YOgxHw"

}

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**A.1.  Example request object**

eyJraWQiOiJjbGllbnQtMjAyMC0wOC0yOCIsImFsZyI6IlBTMjU2In0.eyJhdWQiOiJodHRwczpcL1wv

ZmFwaS1hcy5leGFtcGxlLmNvbVwvIiwibmJmIjoxNTk0MTQwMDMwLCJzY29wZSI6Im9wZW5pZCBwYXlt

ZW50cyIsImlzcyI6IjUyNDgwNzU0MDUzIiwicmVzcG9uc2VfdHlwZSI6ImNvZGUgaWRfdG9rZW4iLCJy

ZWRpcmVjdF91cmkiOiJodHRwczpcL1wvZmFwaS1jbGllbnQuZXhhbXBsZS5vcmdcL2ZhcGktYXMtY2Fs

bGJhY2siLCJzdGF0ZSI6IlZnU1VJRW5mbG5EeFRlMXZBdHI1NG8iLCJleHAiOjE1OTQxNDAzOTAsIm5v

bmNlIjoiN3hEQ0h2aXVQTVNYSklpZ2tIT2NEaSIsImNsaWVudF9pZCI6IjUyNDgwNzU0MDUzIn0.VSo5

VWN3lOiCry2KItU5RI62i9KG2KQlBdpsDT0DI0vSMK-q85aJZvsMiHBNBv1PQ9qAWmU3oJS-yi-Ks\_lD

lP6lIMFrOL\_Ym3VxJ\_SM6lrc8JSZH\_nNx6sqxPpeMQTF4SFPx30vHrlBVJaCGfnCMVC6Nbzwef0vOEpN

ixZT-9cwa3dZ-pddAyt58dKGxS76NR\_wxdBaSKN0AfPoui0HSSaAkIdRds21NKIOf4r9BjV5lr1Oi-4I

JUQp-xdeLCPD3fD6Y-TJbHFToJ4FsQzglN83BfNYaeXV\_yTtK7yeSw2R-ee0b3uMV0iD1ee77b7bbcjR

3msLISFjM40d9Pv8qQ

which when decoded has the following body:

{

"aud": "https://fapi-as.example.com/",

"nbf": 1594140030,

"scope": "openid payments",

"iss": "52480754053",

"response\_type": "code id\_token",

"redirect\_uri": "https://fapi-client.example.org/fapi-as-callback",

"state": "VgSUIEnflnDxTe1vAtr54o",

"exp": 1594140390,

"nonce": "7xDCHviuPMSXJIigkHOcDi",

"client\_id": "52480754053"

}

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**A.2.  Example signed id\_token for authorization endpoint response**

eyJraWQiOiJzZXJ2ZXItMjAyMC0wOC0yOCIsImFsZyI6IlBTMjU2In0.eyJzdWIiOiIxMDAxIiwiYXVk

IjoiNTI0ODA3NTQwNTMiLCJjX2hhc2giOiJRUjJ6dWNmWVpraUxyYktCS0RWcGdRIiwic19oYXNoIjoi

OXM2Q0JiT3hpS0U2NWQ5LVFyMFFJUSIsImF1dGhfdGltZSI6MTU5NDE0MDA5MCwiaXNzIjoiaHR0cHM6

XC9cL2ZhcGktYXMuZXhhbXBsZS5jb21cLyIsImV4cCI6MTU5NDE0MDM5MCwiaWF0IjoxNTk0MTQwMDkw

LCJub25jZSI6Ijd4RENIdml1UE1TWEpJaWdrSE9jRGkifQ.Z-LpQRuYoiTqEBfVfctn-e6bLwSMqi8wA

3TuARGW6GyD05gPF6TVlUwHgJnSUlhETrzhEUAKKiyGDxGspuBU0OAnB4qepgrEBizk980NjCEVXNkog

v0ANv9VX\_01Lcl0d\_6\_c-AUjwDSuKY8rDfvggKSJFzRilbQuB8b1drAIAZpc6kMObY3PcQZ\_vKTMsQ8l

HCuXXRuAo\_\_0xRE6l\_iiRCos\_940GrJr0Sih9uTQpnCWBoEab1dC0l-vUp4lP0TQRKNpDoPoMOj10KJA

8T8pKhjZ8TKM-wo9A4qH2LBgUIYJyjd8bWfKTZxCNmLRzRr-\_JBG7fF\_fpOUhGT\_DhzMw

which when decoded has the following body:

{

"sub": "1001",

"aud": "52480754053",

"c\_hash": "QR2zucfYZkiLrbKBKDVpgQ",

"s\_hash": "9s6CBbOxiKE65d9-Qr0QIQ",

"auth\_time": 1594140090,

"iss": "https://fapi-as.example.com/",

"exp": 1594140390,

"iat": 1594140090,

"nonce": "7xDCHviuPMSXJIigkHOcDi"

}

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**A.3.  Example signed and encrypted id\_token for authorization endpoint response**

eyJraWQiOiJjbGllbnQtZW5jLTIwMjAtMDgtMjgiLCJjdHkiOiJKV1QiLCJlbmMiOiJBMjU2R0NNIiwi

YWxnIjoiUlNBLU9BRVAtMjU2In0.LFvxFCzJ-1NRl48pXTUs8f2axm5MRe9Cv0dgV6sXTRKwkT3nC2SJ

QlutOol36VARLd3uaIoj4Z7LVV\_MrdIYYvDci2WLlKSlI\_NRgR3qJ25N3S6fCqNEYRgDDbNzSr15MDRc

WQR5Jdl3VP8g748cowD\_2gaopaCzZWTa3r\_J2VOEETfcBAIMX0NbtVA3hHW-rQ0aCC7UIbP0\_oEB2YF0

u6qAXCXuC02nO6coMSpSHTDZwkqkmFiFEKERM\_Gayz3lVddlgfcPR2k76bCUjWy934-rOrOBGcLyS1Ww

aTIqMUS3WEIsAwCDr1Jt4pAioryRLZfLmWNff4QZSBxWejRqpw.uRANzseIWYB9YeAW.sJGqF2ERkMEE

jm8h62tUA4UeZIBqvVRpkQqjTuae7-4ac-4sSth0A3zeERvlyC5GcP0W2tj7uxMi0I4gpN33OfAOR-tA

9E\_47oCHXrOH-7cpLgVIxxWZFx43dhxUh5QHuBfli4nHErMVUsFq6CzQj8Z5SHvBD2Qx3suPEeCNo\_M2

woohCprwjOKhE-Q\_VkWUJb-Elrq9HxJcBtadw0spolqgYYTIWvV4fcKmbtGANYLac29oKWd5-jyDAsSF

FZrSCNxv-BtJUiUVWUn5eVufjJYCx62Ju-MZ8vsPNTE-\_I5em9RTBja6ylcivjzhW9Ncl6yKVfnB0XJN

cSSHQSFhc6Gvy7oYMBXx1C5G31OsiklkKQX2gsAZlxFQ\_X25AXpMoV8-5xsUwdMdTaPxIIsccbrK2dfA

aP0rUruSV8zrlrbsN3ftjTJSka2XGG3kra76EPAlzSwxy6XdFVtEV31hirV3f9g04Gj\_e-Q7J7HR62eY

3\_09WyARShQL3DVXWOcK\_8YrLr58JjNAbm0s5dAUq-zt9cMv8rl05t\_dE59Gi6Hnl2YAiRdYG6B71FxJ

CE2Uqciy2jLe6mCDFDfqkog4G5R9FzNz5VzhVpmZVm3OJkug-UzayN7nwZ7jsmxQ2ucCM03xq-0MLdsk

H-cleahkFw5S-W40cn5hLrRXSqUoYfKmVSd9RltOZ6T0VrYpw2LaF2uUYEO9w9bMmg2zzfxft4WHsEbD

OlJVb5SE8mUjzBBZAcgaHYSv0Wii70lEJvLSdnVI1r9kuu9ae\_j1Tu08RVyFGfgixYjI9z2L\_sc8uOoO

HJ-Tq1iuncL3lCQJBuwBFoxyINlFgz4YV2AgreNsX8bDfE9XbRB9TnfvSd6rmes9lO0-3VQFlsC0C5dx

VXgp5o05E8nisPwuLmlGO5BTtBzCQ3tIH2SuTLTG-gohTEUVn4fACwIiyuXdPXcF4GxJNRNgNOH7xwxx

55qEM0xl2GuSseV59FiZR-WKMMs.kScy0JLB4XECklDAwTIVNA

which when decrypted using the following key:

{

"kty": "RSA",

"d": "OjDe8EkZXgvB-Gy5A4EdU8fBuAjdHLMyHKAtMaS\_W\_joEJHDvZRhIYbh1jAyHYoR3kFMXu

tCIYpRjDrsUEhjYuVKLm90CVtysoRjjkiXyupcEW3o--X\_HBJhKm1Y-0I7LQ-cA7CotJpTVMR2fRTqP1

T4FsORAjg9l-fbdpVmeDiZBRbL2zCWmKWhtDpHyy7vbSCRghntihz\_M5Hrchk7r8ito\_K3dFrV9IZSF9

RoEY7kyK5bL36Kpgai44PYCzqOzqP2fteO\_rZ9fn-uK59pI3ySo\_PgSbJ55n14Nd9Z8m70zE9Z4aIeND

EFspZUhavngRwc7MuJ7f\_hVGQ9RFbbkQ",

"e": "AQAB",

"use": "enc",

"kid": "client-enc-2020-08-28",

"n": "jVc92j0ntTV0V1nwZ3mpGaV2bME4d6AMS2SRrJBM0fLehaTEqDNzGu0warz2SC9bhcBOB5

\_q3mYBFjmTwWzSbsk6RYETnAgViXg67PgH7Vkx2NCtwgQW3cNdnUZWRNYHsoevkx\_Ta1X6Vi9ulebU\_B

CKjrF-6CjVcGgEsO\_S5DKcukGHdf81WlQOq3zGQg4h7MLArrbPSTHHORDsu\_87qY9m2EhiYSOBSF5rHs

fDo7zWI5FWNG-\_HO-CBM005bykIIS1aXCXx1jOW1OrKcp5xv3e-BR6MJTxncZJ4o1GtynJI8kLXRgltL

ArSOkbzNEr9GjU9lnSSxKLMtRLKkG2Ow"

}

has the following body:

{

"sub": "1001",

"aud": "2334382354153498",

"acr": "urn:cds.au:cdr:2",

"c\_hash": "BLfy9hvQUZTDq6\_KmF4kDQ",

"s\_hash": "9s6CBbOxiKE65d9-Qr0QIQ",

"auth\_time": 1595827190,

"iss": "https://fapi-as.example.com/",

"exp": 1595827490,

"iat": 1595827190,

"nonce": "7xDCHviuPMSXJIigkHOcDi"

}

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**A.4.  Example JARM response**

eyJraWQiOiJzZXJ2ZXItMjAyMC0wOC0yOCIsImFsZyI6IlBTMjU2In0.eyJhdWQiOiI0NjkxODA2NDgw

MzkwNTEiLCJjb2RlIjoiendrR2FjOWp1TFg4RjhmcmFwRElTaTNLMkZ3bG40cXh3eWZOSUkzQ2p6MCIs

ImlzcyI6Imh0dHBzOlwvXC9mYXBpLWFzLmV4YW1wbGUuY29tXC8iLCJzdGF0ZSI6IlZnU1VJRW5mbG5E

eFRlMXZBdHI1NG8iLCJleHAiOjE1OTQxNDEwOTB9.k\_3df0dIDX6watKxQkzAHOLgf4FBi\_xIPN-n8aT

5hMX3gaBbeDqdUA5NR764L4ugdDgXyQm8dNcZrZldKIPfSfRcjBTtSx9PEdiffn\_xUkwnS18YNAfEoq0

HjvkOQ59F21ImKn113kon00uC2dqBGByRrZcaUYOnvW2DdHCVA0VTW2je5nzbI02z9csLa8uGGGwjWRP

Ec9j9bvR1Adc2m2Z-o0QCRIBl81sZz6\_AnE-wPTw-KZFQBs3FgS-r0FDYOzE7FHIMgDBSKAg1J5tWY3J

wRuIv\_oAbYdSlxdYzrbFQ9grX4MA0p7pk5lS-kwnN845GZ2k1\_yaOLtYYyvRFrw

which when decoded has the following body:

{

"aud": "469180648039051",

"code": "zwkGac9juLX8F8frapDISi3K2Fwln4qxwyfNII3Cjz0",

"iss": "https://fapi-as.example.com/",

"state": "VgSUIEnflnDxTe1vAtr54o",

"exp": 1594141090

}

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**A.5.  Example private\_key\_jwt client assertion**

eyJraWQiOiJjbGllbnQtMjAyMC0wOC0yOCIsImFsZyI6IlBTMjU2In0.eyJzdWIiOiI1MjQ4MDc1NDA1

MyIsImF1ZCI6Imh0dHBzOlwvXC9mYXBpLWFzLmV4YW1wbGUuY29tXC9hcGlcL3Rva2VuIiwiaXNzIjoi

NTI0ODA3NTQwNTMiLCJleHAiOjE1OTQxNDAxNTEsImlhdCI6MTU5NDE0MDA5MSwianRpIjoiNHZCY3RN

U2tLNHdmdU91aTlDeWMifQ.h3i0k2DWc7V6WEiinHAsse-pOFiWxe5kD4KetdGX65Q03orj0Fh6EWfdE

AntCrOodUsypKjM1ia3evbQmsSkhIb4YK5s53hYYtEbJC\_eG9jFnVc4ki7Qc5O-1K-D80w7WT1UI--Ih

Ku-i22Ai\_nMed-71UWLHcPi7W20SCroPHXfaLiFj\_TOsr7I8h7VNsoa7P3-coHlXT5q4cMjIA7t8cRag

sGtKlIgwdFYySlimtSESDM0U-\_NUPperTgnF8FVn7SqtizBJneZNAWwSLJD9AVsnMOH6kOeNLtpopsru

Dcs54S\_aIlroP-BdiHw9R1qRTIVSoX3k\_EStvoWSf8NcQ

which when decoded has the following body:

{

"sub": "52480754053",

"aud": "https://fapi-as.example.com/api/token",

"iss": "52480754053",

"exp": 1594140151,

"iat": 1594140091,

"jti": "4vBctMSkK4wfuOui9Cyc"

}

|  |
| --- |
| [**TOC**](https://openid.net/specs/openid-financial-api-part-2-1_0.html#toc) |

**Appendix B.  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-2-1_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/) |