Financial-grade API (FAPI) Profiles

Comparison of Available FAPI Profiles and Recommendations for New Markets Looking to Implement FAPI as their Security Profile

Prepared by: Chris Michael, Freddi Gyara, Joseph Heenan, Torsten Lodderstedt, Dima Postnikov, and Dave Tonge

Version: 1.0 - 27 Jul 2022
Classification: Public

1. Introduction

1.1 Background

The Financial Grade API (FAPI) profile is a layer on top of OAuth 2.0 and OpenID Connect which "hardens" OAuth / OpenID Connect by specifying a set of constraints - called a *profile* - that limit or enforce the alternatives provided by OAuth / OIDC.

FAPI is now being used as the basis for almost all open banking and open finance standards around the world, including the UK (OBIE), Brazil Open Finance, Australia CDR, Bahrain Open Banking, FDX and SAMA (KSA).

However, some of these standards (e.g., UK/OBIE) are based on profiles/versions that are several years old and thus do not benefit from improvements and enhancements in later versions.

Furthermore, most (if not all) of these standards are using a slightly different profile/version, which is not in the best interests of global interoperability.

1.2 Available FAPI profiles

The following are the FAPI profiles which are either in use by multiple implementers or which are being actively developed by the OpenID Foundation's FAPI working group:

- FAPI 1 Implementers Draft 6 (OBIE Profile): https://openid.net/specs/openid-financial-api-part-2-wd-06.html
- FAPI 1 Baseline: https://openid.net/specs/openid-financial-api-part-1-1_0.html
- FAPI 1 Advanced: https://openid.net/specs/openid-financial-api-part-2-1_0.html
- Brazil Security Standard: https://openbanking-brasil.github.io/specs-seguranca/open-banking-brasil-financial-api-1_ID3-ptbr.html
- FAPI 2: https://openid.net/specs/fapi-2_0-baseline-01.html
- FAPI 2 Message Signing: https://bitbucket.org/openid/fapi/src/master/FAPI_2_0_Advanced_Profile.md

1.3 Purpose of this paper

We will not regurgitate FAPI in its entirety here, rather we will focus on the following areas where FAPI enhances security, and provide recommendations for each area:

1. Choice of cryptographic algorithms
2. Transport security
3. Client authentication method
4. Parameter passing for authorization grants
5. Response validation

We will then set out a number of different FAPI profiles and assess the relative suitability of each for any new/emerging open banking/finance standard.

This paper is thus designed to provide clear recommendations, especially for any new/emerging open finance standard or ecosystem, as to which profile/version of FAPI should be used.

2. Considerations and Recommendations

2.1 Choice of cryptographic algorithms

All versions of FAPI require the use of asymmetric cryptographic algorithms wherever anything is signed or encrypted.

Of note, is the exclusion of RS256 as a safe algorithm. Only the early drafts of the OBIE standard allowed for this as there was insufficient library support for the safer PS256 algorithm at that time.

In addition to identifying safe algorithms, FAPI 2 adds reference to the much more comprehensive RFC8725 / BCP225 - JSON Web Token Best Current Practices.

Recommendation 1: The specification for safe cryptographic algorithms in FAPI 1 Advanced should be made mandatory.

Recommendation 2: Adherence to RFC8725 / BCP225 - JSON Web Token Best Current Practices, is recommended but should remain optional for now, but should be made mandatory as/when there is wider support from vendors and implementers.

2.2 Transport security

FAPI 1 allows two methods of client authentication - private-key-jwt and tls-client-auth. Both methods require the client to present a transport certificate so that the authorization server can issue sender-constrained access tokens and use mtls (RFC 8705) for ensuring access tokens are certificate bound.

FAPI 2 has a comprehensive range of network layer protections (e.g., requiring DNSSEC etc) over and above the use of mTLS or Demonstration of Proof of Possession (DPoP).

Recommendation 3: Adhere to the recommendations in FAPI1 Advanced on using mTLS [RFC 8705] for certificate-bound access tokens.

2.3 Client authentication method

One of the largest improvements of OpenID Connect over plain OAuth 2.0 is the ability to provide an extensible set of ways for an OIDC Client to authenticate itself with the authorization server.

This is covered in Section 9 of the OpenID Connect specification with references to RFCs for each of the specific methods.

All FAPI specifications require the use of either tls_client_auth or private_key_jwt as authentication methods, the general principle being that client authentication should rely on an asymmetric algorithm.

Recommendation 4: Adhere to the recommendations in FAPI 1 Advanced and FAPI 2.

2.4 Parameter passing for authorization grants

In an OAuth 2.0 authorization code grant (and even in hybrid flows) the OAuth 2.0 client constructs a call to the authorization server URL with some query parameters.

This is then sent on to the user agent of the resource owner as a redirect. The user agent (browser/mobile app) follows the redirect in turn, eventually hitting the authorization server.

Since this interaction takes place over an unsecured “front channel” (through the user’s browser) a number of security issues are present:

- A man-in-the-middle attacker may have the opportunity to modify the query parameters
- An attacker pretending to be the client may craft a call to the authorization server and get hold of the response
- If the query parameters are sensitive in nature, these are publicly readable as they travel unencrypted over the internet.

The same problems are repeated for responses sent by the authorization server as the response takes the shape of a redirect URI which passes through the end-user’s browser and then on to the client.

The solutions available through various profiles are:

- use of signed request object (as per Section 6.1 of the OIDC specification)
- use of request object by reference (as per Section 6.3 of the OIDC specification)
- use of PAR (RFC-9126), see below

Pushed Authorization Requests (PAR) offers a number of advantages over the other two methods:

- PAR provides a standardised means for a client to create the request object.
- For PAR requests, the authorization server authenticates the client using the client authentication method the client is registered with providing a high degree of security.
- PAR prevents the contents of the request being passed via the browser potentially providing privacy benefits
- The current draft of FAPI 2 mandates the use of PAR with client-authentication.

Recommendation 5: PAR should be mandated as the method for parameter passing for authorization grants.

Recommendation 6: The security profile should adopt the profile rules defined by FAPI 2 for PAR to simplify a future migration to FAPI 2.

2.5 Response validation

The simplest method for enforcing validations in responses is by enforcing the use of `code id_token` as the response type. This forces the authorization server to issue an id_token along with the authorization code.

- The id_token is signed by the authorization server which ensures that it cannot be tampered with.
- The id_token contains hashed values of part of the `state` and `code` parameters (`s_hash` and `c_hash`) which can be used to validate these and ensure that they have not been tampered with.

JWT Secured Authorization Response Mode (JARM) provides a means for an authorization server to respond to an authorization code grant with a signed JWT as its response. This ensures the client that the response has not been tampered with by a man-in-the-middle.

The JWT may be encrypted, signed or both, allowing for secrecy of the authorization code that the authorization server responds with.

The solutions available through various profiles are:

- use of bindings in id_token claims
- use of JARM

Recommendation 7: Adhere to the recommendations in FAPI 1 Advanced.

3. Comparison Between Profiles

For the purposes of this comparison, we will exclude the following profiles:

- **FAPI 1 Baseline:** This profile originated from an early segregation of “Read” and “Read-Write” profiles. The Baseline profile is insufficient for our purposes, not least since no other standards body or regulator has considered it due to the lack of functionality and detailed definition.
- **FAPI 2 Message Signing:** This is still an early draft and has not reached sufficient maturity for consideration at this stage.

The table below shows a comparison of the remaining profiles, with a summary, status and implications for each:

PROFILE	OBIE Profile	Brazil Profile	FAPI 1 Advanced	FAPI 1 Advanced with PAR	FAPI 2
Summary	Arose from an early draft of FAPI 1 before that was finalised. Includes a number of 'relaxations' to address lack of support for features in some available products at the time.	A derivative of FAPI 1 Advanced. Contains multiple permitted variations, which does give implementers considerable flexibility as to which to use.	Developed off the back of the OBIE Profile. Includes additional enhancements, clarifications and 'hardening' to address issues seen during implementations.	As stated, the FAPI 1 Advanced profile but with mandatory support for Pushed Authorization Requests (PAR). The Brazil Profile is a superset of this.	An evolution of FAPI 1 Advanced which adds support for Private Key, PAR and JARM (as an optional extension). Easier for implementers to understand and designed to cater for requirements from new and emerging standards bodies globally. Includes a formal attacker model.
Status of implementations	Initially implemented by several UK providers, however all instances should have migrated to FAPI 1 Advanced. Not widely implemented outside the UK.	Mandated by the BCB for all Open Finance implementations, hence actively used by the entire Brazil Open Finance ecosystem. However, multiple variations have now been implemented with little consistency.	Mandated for all providers in Bahrain. Recommended for the latest OBIE standard.	Already implemented by several providers in Brazil.	Although proposed in several roadmaps, not yet formally adopted by any other standard body.
Status of conformance suite	Available, but not actively maintained	Available and actively maintained	Available and actively maintained	Available and actively maintained	In development but not yet available as a final published version.
Suitability for new/emerging standards	No longer recommended for implementations since this was an early draft which is over 5 years old and has been superseded multiple times.	While this profile is fit-for-purpose in principle, the large number of available options is likely to result in fragmentation which will slow down and limit interoperability.	As a core profile, this is fit for purpose. However, recommendations 5 and 6 above clearly set out the benefits and requirements of PAR.	Meets all stated requirements and recommendations above. Also provides a relatively simple upgrade path for implementers to move to FAPI 2 at a later stage.	Some providers and relying parties will be reluctant to implement a profile which is still in draft. The lack of a final version (with a final conformance suite) could limit the ability of any regulator to enforce conformance.

4. Summary

FAPI 2 should be the recommended security profile for open API (e.g., open banking/finance) standards and ecosystems, as it meets all the recommendations in this paper.

However, there is a reluctance from many providers (e.g., banks and financial institutions) to implement a draft specification, especially in a highly regulated sector. There is also a reluctance from some regulators to implement a strong ‘regime’ of conformance and certification where either the profile and/or conformance suite are still in draft.

Therefore, for ecosystems looking to implement open banking/finance before the end of 2022, we recommend the following is adopted as the official security profile:

- FAPI 1 Advanced.
- Mandating the PAR option within FAPI 1 Advanced (to enable an easier migration path to FAPI 2) as the method for parameter passing for authorization grants.

This profile can simply be stated as ”FAPI 1 Advanced with PAR”, so there is no need for any standards body or ecosystem to create any new FAPI profile or derivative.

The benefits of this approach are:

1. Providers and relying parties are implementing a security profile which is mature, well defined and widely used, which will give all parties assurance.
2. There are a large number of vendor solutions supporting this profile, which will speed up implementation for providers and relying parties.
3. There is a robust and comprehensive conformance suite, which will enable a much higher level of conformance in any ecosystem.
4. Both the profile and the conformance suite are actively maintained and supported by the OpenID Foundation, which significantly reduces the work for any other standards body in this regard.
5. There is a relatively simple upgrade path to FAPI 2 for both providers and relying parties, which makes any solution future proof and supports interoperability between ecosystems.

As soon as FAPI 2 and the corresponding conformance suite are finalised, and as soon as the OIDF are able to validate and publish FAPI 2 certifications, then we recommend that standards bodies, ecosystems and implementers migrate to FAPI 2.