Bringing Semantics to Authentication: An OpenID Connect Extension

Brendon Vicente R. Silva[1], **Frederico Schardong**[2], **Ricardo F. Custódio**[1]

[1]Laboratório de Segurança em Computação (LabSEC)
Universidade Federal de Santa Catarina (UFSC)

[2]Instituto Federal do Rio Grande do Sul (IFRS)

bredstone13@gmail.com, frederico.schardong@rolante.ifrs.edu.br,
ricardo.custodio@ufsc.br

Abstract. *OpenID Connect (OIDC) is a widely adopted authentication protocol, yet it offers limited expressiveness when conveying details about how a user was authenticated. The Authentication Methods References (`amr`) claim used for this purpose lacks structure and semantic clarity, hindering scenarios that require higher assurance. This paper proposes an extension to OIDC that introduces the `amr_details` claim — a structured, interoperable mechanism for describing authentication factors along with relevant metadata, such as assurance levels and trust frameworks. By enhancing the protocol's expressiveness without compromising compatibility, the extension enables granular access control, thereby contributing to increased trust in distributed identity systems.*

1. Introduction

Over the past decade, digital authentication mechanisms have undergone a significant evolution. What was once a simple exchange of credentials has evolved into complex architectures, forming interconnected ecosystems that involve multiple services, devices, and agents [Schardong and Custódio 2022]. This transformation has not only broadened the scope of issues related to digital identities but also significantly advanced the field, making elements such as usability, interoperability, auditability, and privacy fundamental requirements in the development and adoption of such systems [Bonneau et al. 2012].

In response to these demands, Identity Providers (IdPs) have adopted strategies seeking to balance user convenience with resilience against complex cybersecurity threats [Ometov et al. 2018]. The current diversity of available authentication methods — such as passwords, physical tokens, and Time-based One-Time Passwords (TOTP) [M'Raihi et al. 2011] — reflects this effort to strengthen identity verification processes. Simultaneously, the need for standardization and interoperability has driven the emergence of widely adopted specifications such as OAuth 2.0 [Hardt 2012] and the OpenID Connect (OIDC) [Sakimura et al. 2014] protocol family.

However, these frameworks do not fully accommodate the emerging demands of an ever-evolving digital ecosystem [Parecki et al. 2019]. In OIDC, the use of claims — structured pieces of information embedded in tokens that convey attributes about the user or the authentication event — is established as a means to transmit relevant data. One specific claim is the Authentication Methods References (`amr`), which indicates the authentication factors[1] employed by the user. However, this claim only allows a purely

[1] In this paper, the terms *authentication method* and *authentication factor* are used interchangeably.

enumerative representation, listing identifiers such as `pwd` for password or `face` for biometrics. As a result, important contextual information is lost, such as the origin of each method, specific parameters related to the factor's nature, or metadata useful for audit and governance, such as the number of times someone attempted to enter their password.

In this context, this paper proposes an extension to the OpenID Connect specification that enables a more structured and detailed description of the methods used in an authentication session. The `amr_details` claim is introduced: a complementary element to `amr`, designed to provide contextual and semantic information about each authentication factor in a fully compatible manner with OIDC and existing applications. This approach is expected to enhance transparency, auditability, and interoperability among entities in identity ecosystems. This paper is organized as follows: Section 2 details the proposal, and Section 3 its benefits, applications, and potential future developments.

2. Extending OpenID Connect

The structure of the `amr_details` claim adheres to conventions established by extensions such as the OpenID Connect for Identity Assurance (OIDC4IA) specification [Lodderstedt et al. 2022], particularly in the use of well-defined fields, semantic labels, and metadata in the verification context. This claim is defined as a list of JSON objects [Bray 2017], each representing an authentication method employed by the user. Each entry includes both required and optional fields that describe the relevant characteristics of the factor in a standardized manner. The main fields of the proposed schema are outlined below:

- **`auth_method`** (REQUIRED): String identifier of the authentication method used (e.g., `pwd`, `hwk`, `sms`, or `face`). This value MUST match one of the values present in the `amr` claim.
- **`src`** (REQUIRED): A JSON object describing the source and context for this authentication method, detailed below:
  - **`iss`** (REQUIRED): Identifier of the IdP responsible for performing the authentication. Its value should be a case-sensitive URL containing scheme, host, and, optionally, port number and path components, with no query or fragment components.
  - **`trust_framework`** (`OPTIONAL`): Identifier of the trust framework used for the authentication. Values MAY come from known registries such as the Internet Assigned Numbers Authority (IANA) Trust Frameworks registry [2], or be defined by the IdP in accordance with its own policies.
  - **`assurance_level`** (OPTIONAL): The assurance level assigned to the authentication factor, according to the specified `trust_framework`.
  - **`time`** (REQUIRED): Timestamp indicating when the authentication occurred, formatted as an ISO 8601 string.
  - **`location`** (OPTIONAL): A JSON object containing approximate geographic or postal information about the origin of the authentication request. In addition to the standard `address` fields defined in OIDC Core, it MAY also include network and geospatial attributes — such as `ip_address`, `latitude`, `longitude`, and `precision` — depending on the parameters of the authentication request or the policies of the IdP.

[2] `https://www.iana.org/assignments/loa-profiles`

- **`auth_details`** (OPTIONAL): A JSON object containing metadata specific to the authentication factor used. The set of attributes depends on the value of `auth_method` and adheres to a standardized vocabulary. For example, when `auth_method` is `pwd`, the `auth_details` object MAY include the attributes `hash_algo`, `hash_iterations`, `created_at`, `last_changed`, and `last_verified`. Additional authentication methods and their corresponding attributes are omitted due to space limitations.

To facilitate gradual adoption, the `amr_details` claim is defined as optional. Its inclusion in tokens can be determined by the IdP, either by default or in response to client requests. They may request it explicitly using the `claims` parameter or through a custom `scope` value, in accordance with OIDC's extensibility mechanisms and mutual agreement between the parties.

Figure 1 illustrates two examples of how the `amr_details` mechanism can be used. The first example illustrates the details of a digital certificate's Proof-of-Possession (PoP) key authentication using a hardware-secured key, including relevant metadata about the digital certificate itself. The second example shows an authentication scenario where a user employs two factors: a password and an OTP sent via SMS. The latter is provided by an IdP different from the former. The trust relationships between IdPs, and the mechanisms by which `amr` and `amr_details` claims are validated and accepted across entities, are considered out of the scope of this proposal.

```
{
 "amr": ["hwk"],
 "amr_details": [ {
  "auth_method": "hwk",
  "src": {
   "iss": "https://idp.example.com"
   "trust_framework": "eidas",
   "assurance_level": "substantial",
   "time": "2025-03-12T19:31:16Z",
   "location": {
    "ip_address": "203.0.113.42",
    "country": "BR"
   }
  },
  "auth_details": {
   "subject": "Brendon Vicente",
   "issuer": "VALID",
   "serial_number": 123,
   "valid_from": "2024-06-17T10:00:00Z",
   "valid_to": "2027-06-17T10:00:00Z"
  }
 } ]
}
```

(a) `amr_details` in a PoP for hardware-secured key authentication flow.

```
{
 "amr": ["pwd", "sms"],
 "amr_details": [ {
  "auth_method": "sms",
  "src": {
   "iss": "https://external.com"
   "time": "2025-04-23T18:25:20Z",
  },
  "auth_details": {
   "otp_length": 6,
   "otp_alg": "TOTP",
   "otp_ttl": 300,
   "delivery_time": "2025-04-23T18:24:13Z
      ",
   "attempts": 1
  }
 }, {
  "auth_method": "pwd",
  "src": {
   "iss": "https://idp.example.com"
   "time": "2025-04-23T18:24:12Z",
  },
  "auth_details": {
   "hash_algo": "pbkdf2-sha256",
   "hash_iterations": 27500,
   "created_at": "2021-07-12T09:48:21Z"
  }
 } ]
}
```

(b) `amr_details` in an authentication flow with multiple IdPs.

Figure 1. `amr_details` examples.

3. Final Remarks

The introduction of the `amr_details` claim addresses an important gap in the OpenID Connect specification related to how authentication methods used in an authentication session are represented. While the existing `amr` claim only lists opaque identifiers, this proposed extension provides a structured and interoperable description of essential information. This includes details such as assurance levels, the source of authentication, the time of validation, and method-specific metadata.

This new approach enhances the auditability of authentication processes and allows applications to make more informed decisions. The benefits of this extension are evident in several practical scenarios. For instance, in risk-based authorization systems, attributes such as location, authentication method employed, and the number of login attempts can be used to dynamically adjust the permissions granted to a user.

Another key point of this extension is the semantic compatibility with other OpenID Foundation protocols, especially the OIDC4IA profile. By adopting already-established terminology, the model aligns with recognized extensions, reinforcing its integration capabilities with complementary solutions and increasing its relevance. This alignment also facilitates progressive adoption, including in legacy environments, as the introduction of the `amr_details` claim remains fully compatible with the infrastructure already in use by IdPs implementing the OIDC specifications.

Finally, we are investigating whether this extension can also enable applications to request specific authentication factors from the IdP — a capability not currently covered by OpenID specifications. If feasible, this would give clients explicit control over authentication processes while preserving the protocol's extensibility principles.

References

Bonneau, J., Herley, C., Van Oorschot, P. C., and Stajano, F. (2012). The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE symposium on security and privacy*, pages 553–567. IEEE.

Bray, T. (2017). The JavaScript Object Notation (JSON). RFC 8259.

Hardt, D. (2012). The OAuth 2.0 Authorization Framework. RFC 6749, IETF.

Lodderstedt, T., Fett, D., Haine, M., Pulido, A., Lehmann, K., and Koiwai, K. (2022). OpenID Connect for Identity Assurance 1.0.

M'Raihi, D., Rydell, J., Pei, M., and Machani, S. (2011). TOTP: Time-Based One-Time Password Algorithm. RFC 6238.

Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., and Koucheryavy, Y. (2018). Multi-Factor Authentication: A Survey. *Cryptography*, 2(1).

Parecki, A., Hardt, D., and Lodderstedt, T. (2019). OAuth 2.1. *URL https://www. ietf. org/proceedings/106/slides/slides-106-oauth-sessa-oauth-21-00. pdf. IETF*, 106.

Sakimura, N., Bradley, J., Jones, M., De Medeiros, B., and Mortimore, C. (2014). OpenID Connect Core 1.0. *The OpenID Foundation, specification*, 335.

Schardong, F. and Custódio, R. (2022). Self-Sovereign Identity: A Systematic Review, Mapping and Taxonomy. *Sensors*, 22(15).