Final
D. Recordon
Six Apart
M. Jones
Microsoft
J. Bufu, Ed.
Independent
J. Daugherty, Ed.
JanRain
N. Sakimura
NRI
December 30, 2008

OpenID Provider Authentication Policy Extension 1.0

[30 December 2008]

Abstract

This extension to the OpenID Authentication protocol provides a mechanism by which a Relying Party can request that particular authentication policies be applied by the OpenID Provider when authenticating an End User. This extension also provides a mechanism by which an OpenID Provider may inform a Relying Party which authentication policies were used. Thus a Relying Party can request that the End User authenticate, for example, using a phishing-resistant or multi-factor authentication method.

This extension also provides a mechanism by which a Relying Party can request that the OpenID Provider communicate the levels of authentication used, as defined within one or more sets of requested custom Assurance Levels, and for the OpenID Provider to communicate the levels used.

This extension is not intended to provide all information regarding the quality of an OpenID Authentication assertion. Rather, it is designed to be balanced with information the Relying Party already has with regard to the OpenID Provider and the level of trust it places in it. If additional information is needed about processes such as new End User enrollment on the OpenID Provider, such information should either be transmitted out-of-band or in other extensions such as OpenID Attribute Exchange. Other aspects (e.g. security characteristics, credential provisioning, etc) could be dealt with in the future.

This extension is optional, though its use is certainly recommended. This extension can be used with OpenID Authentication versions 1.1 and 2.0.

While none of the information transmitted using this extension can be verified by the Relying Party using technology alone, this does not limit the utility of this extension. Because there is no trust model specified by OpenID, Relying Parties must decide for themselves which Providers are trustworthy; likewise, RPs can decide whether to trust authentication policy claims from such OpenID Providers as well. As with other OpenID

extensions, it is the Relying Party's responsibility to implement policy relative to the OpenID Provider's response.

Table of Contents

---

1. Definitions

1.1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] (Bradner, B., “Key words for use in RFCs to Indicate Requirement Levels,” 1997.) .

1.2. Conventions

Throughout this document, values are quoted to indicate that they are to be taken literally. When using these values in protocol messages, the quotes MUST NOT be used as part of the value.

All OpenID 2.0 messages that contain a Provider Authentication Policy Extension (PAPE) element MUST contain the following extension namespace declaration, as

specified in the Extensions section of [OpenIDAuthentication2.0] (specs@openid.net, “OpenID Authentication 2.0,” 2007.) .

```
openid.ns.<alias>=http://specs.openid.net/extensions/pape/1.0
```

The actual extension namespace alias should be determined on a per-message basis by the party composing the messages, in such a manner as to avoid conflicts between multiple extensions. For the purposes of this document and when constructing OpenID 1.1 messages, the extension namespace alias SHALL be "pape".

Additionally, this specification uses name spaces for the custom authentication level identification. It is in the form of

```
openid.pape.auth_level.ns.<cust>=http://some.authlevel.uri
```

The actual extension namespace alias should be determined on a per-message basis by the party composing the messages, in such a manner as to avoid conflicts between multiple extensions. For the purposes of this document and when constructing OpenID 1.1 messages, the one custom authentication level identification extension namespace defined by this specification is "nist". Others may also be defined and used by implementations, for example, "jisa".

1.3. Terminology

The following terms are defined in [OpenIDAuthentication2.0] (specs@openid.net, “OpenID Authentication 2.0,” 2007.) :

- Identifier
- OpenID Provider (OP)
- Relying Party (RP)
- User-Agent

Authentication Method:
: An Authentication Method is a single mechanism by which the End User authenticated to their OpenID Provider, for example, a password or a hardware credential.

Authentication Policy:
: An Authentication Policy is a plain-text description of requirements that dictate which Authentication Methods can be used by an End User when authenticating to their OpenID Provider. An Authentication Policy is defined by a URI which must be previously agreed upon by one or more OPs and RPs.

---

2. Extension Overview

1. As part of the [Yadis] (Miller, J., Ed., “Yadis Specification 1.0,” 2005.) Discovery process, OpenID Providers can optionally add supported authentication policies

to an End User's XRDS document. This aids Relying Parties in choosing between multiple listed OPs depending on authentication policy requirements.
2. The Relying Party includes parameters in the OpenID Authentication request describing its preferences for authentication policy for the current assertion.
3. The OpenID Provider processes the PAPE request, prompting the End User to fulfill the requested policies during the authentication process.
4. As part of the OpenID Provider's response to the Relying Party, the OP includes PAPE information around the End User's authentication. An OP MAY include this response information even if not requested by the RP.
5. When processing the OpenID Provider's response, the Relying Party takes the PAPE information into account when determining if the End User should be sent through additional verification steps or if the OpenID login process cannot proceed due to not meeting policy requirements.

---

3. Advertising Supported Authentication Policies

Via the use of [Yadis] (Miller, J., Ed., “Yadis Specification 1.0,” 2005.) within OpenID, Relying Parties are able to discover OpenID Provider service information in an automated fashion. This is used within OpenID Authentication for a RP to discover what version of the protocol each OP listed supports as well as any extensions, such as this one, that are supported. To aide in the process of a Relying Party selecting which OP they wish to interact with, it is STRONGLY RECOMMENDED that the following information be added to the End User's XRDS document. An OP may choose to advertise both custom levels and supported polices in the same <xrd:Service>. An OP should only advertise the authentication policies and custom assurance level namespaces that it supports.

When advertising supported policies, each policy URI MUST be added as the value of an <xrd:Type> element of an OpenID <xrd:Service> element in an XRDS document.

Example:

```
<xrd>
  <Service>
    <Type>http://specs.openid.net/auth/2.0/signon</Type>
    <Type>
  http://schemas.openid.net/pape/policies/2007/06/phishing-resistant
    </Type>
    <URI>https://example.com/server</URI>
  </Service>
</xrd>
```

When advertising supported custom Assurance Level name spaces, each name space URI MUST be added as the value of an <xrd:Type> element of an OpenID <xrd:Service> element in an XRDS document.

Example:

```
<xrd>
```

```
  <Service>
    <Type>http://specs.openid.net/auth/2.0/signon</Type>
    <Type>
  http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf
    </Type>
    <URI>https://example.com/server</URI>
  </Service>
</xrd>
```

---

4. Defined Authentication Policies

The following are defined policies and policy identifiers describing how the End User may authenticate to an OP. Additional policies can be specified elsewhere and used without making changes to this document. The policies described below are designed to be a starting point to cover the most common use-cases. Additional polices can be found at http://schemas.openid.net/pape/policies/.

When multiple policies are listed in the Relying Party's request, the OpenID Provider SHOULD satisfy as many of the requested policies as possible. This may require, for instance, that a user who has already been authenticated using one authentication method be re-authenticated with different or additional methods that satisfy the request made by the Relying Party. It is always the responsibility of the RP to determine whether the particular authentication performed by the OP satisfied its requirements; this determination may involve information contained in the PAPE response, specific knowledge that the RP has about the OP, and additional information that it may possess or obtain about the particular authentication performed.

- Phishing-Resistant Authentication

  `http://schemas.openid.net/pape/policies/2007/06/phishing-resistant`

  An authentication mechanism where a party potentially under the control of the Relying Party can not gain sufficient information to be able to successfully authenticate to the End User's OpenID Provider as if that party were the End User. (Note that the potentially malicious Relying Party controls where the User-Agent is redirected to and thus may not send it to the End User's actual OpenID Provider).

- Multi-Factor Authentication

  `http://schemas.openid.net/pape/policies/2007/06/multi-factor`

  An authentication mechanism where the End User authenticates to the OpenID Provider by providing more than one authentication factor. Common authentication factors are something you know, something you have, and something you are. An example would be authentication using a password and a software token or digital certificate.

- Physical Multi-Factor Authentication

`http://schemas.openid.net/pape/policies/2007/06/multi-factor-physical`

An authentication mechanism where the End User authenticates to the OpenID Provider by providing more than one authentication factor where at least one of the factors is a physical factor such as a hardware device or biometric. Common authentication factors are something you know, something you have, and something you are. This policy also implies the Multi-Factor Authentication policy (http://schemas.openid.net/pape/policies/2007/06/multi-factor) and both policies MAY BE specified in conjunction without conflict. An example would be authentication using a password and a hardware token.

Of the policies defined above, two are not independent. All authentications satisfying the Multi-Factor Physical policy also satisfy the Multi-Factor policy. Therefore, whenever the OP returns a result saying that Multi-Factor Physical authentication was performed it MUST also indicate that Multi-Factor authentication was performed.

4.1. Custom Assurance Level Name Spaces

Custom Assurance Levels are optional. The namespaces may be defined by various parties, such as country or industry specific standards bodies, or other groups or individuals.

The namespace URI should be chosen with care to be unambiguous when used as a <xrd:Type> element to advertise the namespaces supported by the OP.

The custom Assurance Level namespace should define the meaning of the strings that are returned by the OP in the openid.pape.auth_level.<cust> element.

---

5. Authentication Protocol

5.1. Request Parameters

The following parameters MUST be included during an OpenID Authentication request (specs@openid.net, “OpenID Authentication 2.0,” 2007.) [OpenIDAuthentication2.0] by the Relying Party that uses this extension unless marked as optional.

- openid.ns.pape

  Value:

  `http://specs.openid.net/extensions/pape/1.0`

- openid.pape.max_auth_age

  (Optional) If the End User has not actively authenticated to the OP within the number of seconds specified in a manner fitting the requested policies, the OP SHOULD authenticate the End User for this request using the requested policies. The OP MUST actively authenticate the user and not rely on a browser cookie from a previous authentication.

  Value: Integer value greater than or equal to zero in seconds.

  If an OP does not satisfy a request for timely authentication, the RP may decide not to grant the End User access to the services provided by the RP. If this parameter is absent from the request, the OP should authenticate the user at its own discretion.

- openid.pape.preferred_auth_policies

  Zero or more authentication policy URIs representing authentication policies that the OP SHOULD satisfy when authenticating the user. If multiple policies are requested, the OP SHOULD satisfy as many of them as it can.

  Value: Space separated list of authentication policy URIs.

  If no policies are requested, the RP may be interested in other information such as the authentication age.

  Example:

```
openid.pape.preferred_auth_policies=
  http://schemas.openid.net/pape/policies/2007/06/phishing-
resistant
  http://schemas.openid.net/pape/policies/2007/06/multi-factor
```

- openid.pape.auth_level.ns.<cust>

  (Optional) The name space for the custom Assurance Level. Assurance levels and their name spaces are defined by various parties, such as country or industry specific standards bodies, or other groups or individuals.

  Value: URL that represents this Assurance Level.

  Example:

```
openid.pape.auth_level.ns.nist=
  http://csrc.nist.gov/publications/nistpubs/800-63/SP800-
63V1_0_2.pdf
openid.pape.auth_level.ns.jisa=
  http://www.jisa.or.jp/spec/auth_level.html
```

- openid.pape.preferred_auth_level_types

(Optional) A list of the name space aliases for the custom Assurance Level name spaces that the RP requests be present in the response, in the order of its preference.

Value: Space separated list of the name space aliases, in the order of the RP's preference.

Example:

```
openid.pape.preferred_auth_levels=jisa nist
```

5.2. Response Parameters

In response to a Relying Party's request, the following parameters MUST be included in the OpenID Authentication Response. All response parameters MUST be included in the signature of the Authentication Response. It is RECOMMENDED that an OP supporting this extension include the following parameters even if not requested by the Relying Party.

All response parameters MUST describe the End User's current session with the OpenID Provider.

- openid.ns.pape

  Value:

  ```
  http://specs.openid.net/extensions/pape/1.0
  ```

- openid.pape.auth_policies

  One or more authentication policy URIs representing policies that the OP satisfied when authenticating the End User.

  Value: Space separated list of authentication policy URIs.

  Note: If no policies were met though the OP wishes to convey other information in the response, this parameter MUST be included with the value of http://schemas.openid.net/pape/policies/2007/06/none.

  Example:

  ```
  openid.pape.auth_policies=
    http://schemas.openid.net/pape/policies/2007/06/multi-factor
    http://schemas.openid.net/pape/policies/2007/06/multi-factor-
  physical
  ```

- openid.pape.auth_time

(Optional) The most recent timestamp when the End User has actively authenticated to the OP in a manner fitting the asserted policies.

Value: The timestamp MUST be formatted as specified in section 5.6 of [RFC3339] (Klyne, G. and C. Newman, “Date and Time on the Internet: Timestamps,” .) , with the following restrictions:

- All times must be in the UTC time zone, indicated with a "Z".
- No fractional seconds are allowed

Example:

```
2005-05-15T17:11:51Z
```

Note: If the RP's request included the "openid.pape.max_auth_age" parameter then the OP MUST include "openid.pape.auth_time" in its response. If "openid.pape.max_auth_age" was not requested, the OP MAY choose to include "openid.pape.auth_time" in its response.

- openid.pape.auth_level.ns.<cust>

  (Optional) The name space for the custom Assurance Level defined by various parties, such as a country or industry specific standards body, or other groups or individuals.

  Value: URL that represents this Assurance Level.

  Example:

```
openid.pape.auth_level.ns.nist=
  http://csrc.nist.gov/publications/nistpubs/800-63/SP800-
63V1_0_2.pdf
openid.pape.auth_level.ns.jisa=
  http://www.jisa.or.jp/spec/auth_level.html
```

- openid.pape.auth_level.<cust>

  (Optional) The Assurance Level as defined by the above standards body, group, or individual that corresponds to the authentication method and policies employed by the OP when authenticating the End User. A custom Assurance Level definition MAY define additional subparameter values that are expressed within its namespace, although for reasons of simplicity, this SHOULD be avoided if possible.

  Value: Strings defined according to this Assurance Level.

  Example:

```
openid.pape.auth_level.nist=1
openid.pape.auth_level.jisa=2
```

6. Security Considerations

Per commonly accepted security practices, it should be noted that the overall strength of any authentication is only as strong as its weakest step. It is thus recommended that provisioning of phishing-resistant and other credentials stronger than shared secrets should be accomplished using methods that are at least as strong as the credential being provisioned. By counter-example, allowing people to retrieve a phishing-resistant credential using only a phishable shared secret negates much of the value provided by the phishing-resistant credential itself. Similarly, sometimes using a phishing-resistant method when a phishable method continues to also sometimes be employed may still enable phishing attacks to compromise the OpenID.

OPs SHOULD attempt to satisfy the authentication policies requested by the RP and the reply SHOULD minimally contain at least the subset of the requested policies that the authentication performed satisfied. The OP MAY also choose to return additional policies that the authentication performed satisfied, even if not requested.

If the RP requested that an authentication level or levels be returned and the OP supports some or all of those level types, then the OP SHOULD return the actual level value for each of the supported types requested, if available.

6.1. NIST Assurance Levels

National Institute of Standards and Technology (NIST) in Special Publication 800-63 (Burr, W., Dodson, D., and W. Polk, Ed., “Electronic Authentication Guideline,” April 2006.) [NIST_SP800-63] defines a set of Assurance Levels from 1 to 4. These may be returned by the OP to the RP to communicate which NIST level the identity proofing, authentication method, and policies employed by the OP when authenticating the End User corresponds to.

Value: Integer value between 0 and 4 inclusive.

Note: Level 0 is not an assurance level defined by NIST, but rather SHOULD be used to signify that the OP recognizes the parameter and the End User authentication did not meet the requirements of Level 1. See Appendix A.1.2 (NIST Authentication Mechanism Levels) for high-level example classifications of authentication methods within the defined levels. Authentication using a long-lived browser cookie, for instance, is one example where the use of "level 0" is appropriate. Authentications with level 0 should never be used to authorize access to any resource of any monetary value whatsoever. The previous sentence should not be construed as implying that any of the other levels are recommended or appropriate for accessing resources with monetary value either without the Relying Party doing an appropriate risk assessment of the particular OpenID provider asserting them and their issuance and authentication procedures as they apply to the particular online interaction in question.

Depending on the particular use case being satisfied by the authentication response and PAPE information, the OpenID Provider will have to make a decision, ideally with the consent of the End User, as if it will include the "openid.pape.auth_level.nist" parameter. This information is designed to give Relying Parties more information around the strength of credentials used without actually disclosing the specific credential type. Disclosing the specific credential type can be considered a potential privacy or security risk.

It is RECOMMENDED that this parameter always be included in the response from the OP. This holds true even in cases where the End User authentication does not meet one of the defined Authentication Policies. For example, if the End User is authenticating using a password via HTTPS there is still value to the RP in knowing if the strength of the Password corresponds to the entropy requirements laid out by Level 1 or 2 or that it does not even meet the minimum requirement for the lowest level. With that said, discretion needs to be used by OP's as conveying that one of their End User's has a weak password to an "un-trustworthy" RP would not generally be considered a good idea.

Appendix A. Examples

Appendix A.1. Authentication Method Classifications

This non-normative section illustrates classification of various common authentication methods and their respective conformance within the defined policies and levels.

Appendix A.1.1. Authentication Policy Examples

This table provides examples of common authentication technologies and their mapping to the Authentication Policies defined in Section 4 (Defined Authentication Policies) .

Method	Phishing-Resistant	Multi-Factor	Physical Multi-Factor
Password via HTTPS			
Visual secret via HTTPS			
PIN and digital certificate via HTTPS	X	X	
PIN and "soft" OTP token via HTTPS		X	
PIN and "hard" OTP token via HTTPS		X	X
PIN and "hard" crypto token via	X	X	X

HTTPS		
Information Card via HTTPS	X	X

Appendix A.1.2. NIST Authentication Mechanism Levels

This section is designed to highlight the Authentication Mechanism Levels described in [NIST_SP800-63] (Burr, W., Dodson, D., and W. Polk, Ed., “Electronic Authentication Guideline,” April 2006.) . All normative and authoritative text can be found in [NIST_SP800-63] (Burr, W., Dodson, D., and W. Polk, Ed., “Electronic Authentication Guideline,” April 2006.) . Note that assurance level is not only comprised of Authentication Mechanism employed but also the nature of the identity proofing performed. The overall assurance level is determined as a combination of these factors.

This table is republished from page 39 of [NIST_SP800-63] (Burr, W., Dodson, D., and W. Polk, Ed., “Electronic Authentication Guideline,” April 2006.) .

Token Type	**Level 1**	**Level 2**	**Level 3**	**Level 4**
Hard crypto token	X	X	X	X
One-time password device	X	X	X	
Soft crypto token	X	X	X	
Passwords & PINs	X	X		

This table is republished from page 39 of [NIST_SP800-63] (Burr, W., Dodson, D., and W. Polk, Ed., “Electronic Authentication Guideline,” April 2006.) .

Protect Against	**Level 1**	**Level 2**	**Level 3**	**Level 4**
On-line guessing	X	X	X	X
Replay	X	X	X	X
Eavesdropper		X	X	X
Verifier impersonation			X	X
Man-in-the-middle			X	X
Session hijacking				X

The following table illustrates the minimum number of factors required at each Authentication Mechanism Level.

Level	**Factors**
1	1
2	1
3	2
4	2

In all cases, implementing a commonly accepted nonce and cross-site scripting protection when entering authentication credentials is required to satisfy all four Authentication Mechanism Levels. All examples below assume this requirement is met.

It should be noted that NIST Authentication Mechanism Levels 1 and 2 have differing password entropy requirements. When working with passwords, you should refer to the [NIST_SP800-63] (Burr, W., Dodson, D., and W. Polk, Ed., “Electronic Authentication Guideline,” April 2006.) specification for more details. All examples below assume the password meets these requirements.

This table provides examples of common authentication technologies and their mapping to NIST Authentication Mechanism Levels, please be aware that there are details not represented in these examples that may bear on the resulting Authentication Mechanism Level.

Method	**Level 1**	**Level 2**	**Level 3**	**Level 4**
Password via HTTP	Yes, if challenge-response			
Password via HTTPS	Yes	Yes		
PIN and Digital Certificate via HTTPS	Yes	Yes	Yes	
PIN and "soft" OTP token via HTTPS	Yes	Yes	Yes	
PIN and "hard" OTP token via HTTPS	Yes	Yes	Yes	
PIN and "hard" crypto token via HTTPS	Yes	Yes	Yes	Yes, if FIPS 140-2 Level 2 crypto and Level 3 physical

---

Appendix B. Acknowledgements

The authors would like to thank Andrew Arnott, John Bradley, Kim Cameron, Barry Ferg, George Fletcher, Dick Hardt, Nate Klingstein, Gary Krall, Ben Laurie, Arun Nanda, Drummond Reed, Tatsuki Sakushima, and Allen Tom for their feedback when drafting this specification. David Recordon would also like to acknowledge VeriSign who employed him during the original authoring of this specification.

---

7. Normative References

[NIST_SP800-63] Burr, W., Dodson, D., and W. Polk, Ed., “Electronic Authentication Guideline,” April 2006.

[OpenIDAuthentication2.0] specs@openid.net, “OpenID Authentication 2.0,” 2007 (TXT, HTML).

[RFC2119] Bradner, B., “Key words for use in RFCs to Indicate Requirement Levels,” RFC 2119, 1997.

[RFC3339] Klyne, G. and C. Newman, “Date and Time on the Internet: Timestamps,” RFC 3339.

[Yadis] Miller, J., Ed., “Yadis Specification 1.0,” 2005 (PDF, ODT).

Authors' Addresses

David Recordon
Six Apart, Ltd.
548 4th Street
San Francisco, CA 94107
USA
Email: david@sixapart.com
URI: http://www.sixapart.com/

Michael B. Jones
Microsoft Corporation
One Microsoft Way, Building 40/5138
Redmond, WA 98052
USA
Email: mbj@microsoft.com
URI: http://www.microsoft.com/

Johnny Bufu (editor)
Independent
Email: johnny.bufu@gmail.com
URI:

Jonathan Daugherty (editor)
JanRain
5331 SW Macadam Ave. #375
Portland, OR 97239
USA
Email: cygnus@janrain.com
URI: http://janrain.com/

Nat Sakimura
Nomura Research Institute, Ltd.
Marunouchi Kitaguchi Building, 1-6-5 Marunouchi
Chiyoda-ku, Tokyo 100-0005
Japan
Email: n-sakimura@nri.co.jp
URI: http://www.nri.co.jp/