ARTICLE EXCERPTED FROM:

INFORMATION STANDARDS QUARTERLY
FALL 2014 | VOL 26 | ISSUE 3 | ISSN 1041-0031

ISQ

TOPIC

IDENTITY MANAGEMENT

PRIVACY BY DESIGN
AND THE ONLINE LIBRARY

FROM THE LIBRARY OF CONGRESS
TO THE LIBRARY OF ME

THE INTENTION PUBLISHING ECONOMY:
WHEN PATRONS TAKE CHARGE

A JSON-BASED IDENTITY
PROTOCOL SUITE

NISO
How the information world
CONNECTS

SP [SPOTLIGHT]

Michael B. Jones

MICHAEL B. JONES

A JSON-Based Identity Protocol Suite

Achieving interoperable digital identity systems requires agreement on data representations and protocols among the participants. While there are several suites of successful interoperable identity data representations and protocols, including Kerberos,[1] X.509,[2] SAML 2.0,[3] WS-*,[4, 5, 6] and OpenID 2.0,[7] they have used data representations that have limited or no support in browsers, mobile devices, and modern Web development environments, such as ASN.1,[8] XML,[9] or custom data representations.

A security token is a cryptographically secured set of statements made by an issuer about a subject that can be used by the intended recipient to make trust decisions about the subject.

A new set of open digital identity standards have emerged that utilize JSON[10] data representations and simple REST-based[11] communication patterns. These protocols and data formats are intentionally designed to be easy to use in browsers, mobile devices, and modern Web development environments, which typically include native JSON support. This paper surveys a number of these open JSON-based digital identity protocols and discusses how they are being used to provide practical interoperable digital identity solutions.

THE EMERGING JSON-BASED IDENTITY PROTOCOL SUITE

This section provides an overview of a set of open, JavaScript Object Notation (JSON)-based digital identity protocols that are being collaboratively developed by members of the identity community. These protocols are designed to work together to enable open, interoperable, claims-based identity, authentication, and authorization services to be built for the Web.

JSON Web Token, Signature, Encryption, Key, and Algorithms Specifications

The ability to produce signed and optionally encrypted security tokens containing claims is fundamental to interoperable identity protocols. A security token is a cryptographically secured set of statements made by an issuer about a subject that can be used by the intended recipient to make trust decisions about the subject. Claims are the individual statements in the security token about the subject made by the issuer. This family of JSON-based specifications meets this need.

CONTINUED »

JSON Web Token (JWT)

A JSON Web Token (JWT)[12] is a means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JSON object that is digitally signed using JSON Web Signature (JWS)[13] and optionally encrypted using JSON Web Encryption (JWE).[14] Using a JWT enables the issuer of a token to make statements about the subject of the token to an intended audience in a way receivers can verify that they were made by the issuer. This capability is fundamental to digital identity systems. For instance, OpenID Connect[15] uses a JWT issued by the identity provider, whose audience is the relying party, to make authoritative claims that a particular user (the subject of the JWT) has logged in at the identity provider.

This specification was developed collaboratively based upon inputs from a number of independently developed precursor JSON token, signing, and encryption specifications. Over a dozen independent and interoperable implementations of JWTs are known to exist at this point—many of them in production use—including by Microsoft, Google, Salesforce, Deutsche Telekom, and Mozilla. The IETF OAuth Working Group[16] has requested publication of JWT as a Request for Comment (RFC)—an IETF standard.

The suggested pronunciation of JWT is the same as the English word "jot."

JSON Web Signature (JWS)

JSON Web Signature (JWS) is a means of representing signed content using JSON data structures. Complementary encryption capabilities are described in the closely related JSON Web Encryption (JWE) specification. For instance, the JSON Web Token (JWT) specification uses JWS for the issuer to sign JWTs.

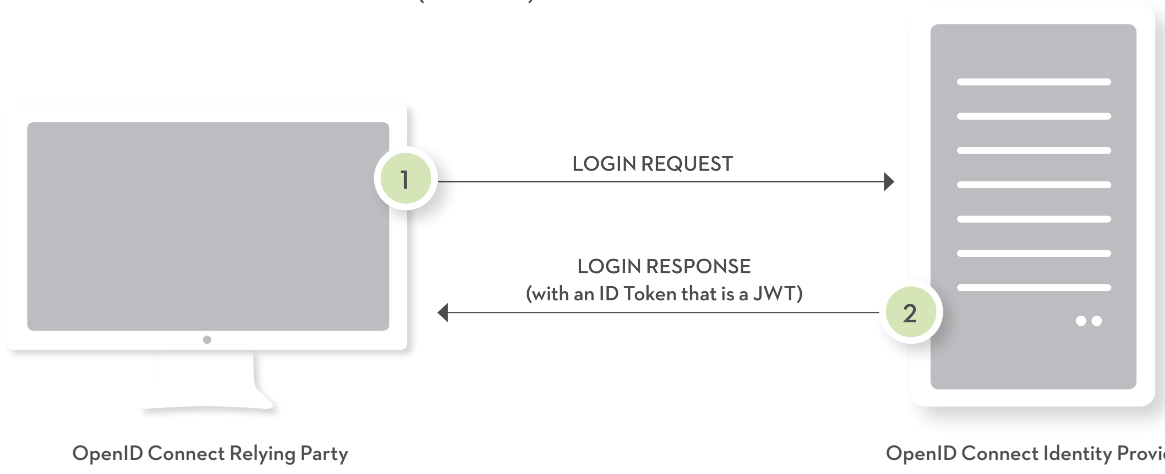
This specification was developed collaboratively based upon inputs from a number of independently developed precursor JSON token, signing, and encryption specifications. Over a dozen independent and interoperable implementations of the JWS specification are known to exist at this point, many of them in production use. The IETF JSON Object Signing and Encryption (JOSE) working group[17] has requested publication of JWS as an RFC—an IETF standard.

JSON Web Encryption (JWE)

JSON Web Encryption (JWE) is a means of representing encrypted content using JSON data structures. This specification complements the signature capabilities described in the closely related JSON Web Signature (JWS) specification. Encryption enables participants to pass confidential messages between themselves.

Several independent and interoperable implementations of the JWE specification are known to exist at this point, many of them in production use. Like JWS, publication of JWE has been requested as an RFC.

JSON WEB TOKEN (JWT)


1
LOGIN REQUEST
LOGIN RESPONSE
(with an ID Token that is a JWT)
2
OpenID Connect Relying Party
OpenID Connect Identity Provider

JSON Web Key (JWK)

A JSON Web Key (JWK)[18] is a JSON data structure that represents a set of cryptographic keys. The JWK format is used to represent bare keys; representing certificate chains is an explicit non-goal of this specification. For instance, sets of JWKs are used by OpenID Connect to publish public keys and enable key rotation. In this use case, the signature on a JWT issued by the identity provider about the user having logged in is verified using keys published by the identity provider as JWKs.

Like the other specifications in this family, over a dozen independent and interoperable implementations of the JWK specification are known to exist at this point, many of them in production use. Like JWS, publication of JWK has been requested as an RFC.

JSON Web Algorithms (JWA)

The JSON Web Algorithms (JWA)[19] specification defines algorithms for use by JWS, JWE, and JWK (and therefore also algorithms used by JWT). Like the other specifications in this family, publication of JWA has been requested as an RFC.

WebFinger

WebFinger[20] defines an HTTPS GET based mechanism to discover the location of a given type of service for a given principal starting only with a domain name. These identifiers are URNs, which could be e-mail addresses, account identifiers, URLs, or other identifiers. For instance, OpenID Connect uses WebFinger to look up the identity provider for a user, given an identifier for the user such as an e-mail address.

OAuth 2.0 Specifications

The OAuth 2.0 family of specifications enables scoped authorization of third-party applications to HTTP-based services to occur without releasing end-user credentials to those applications. This scoped authorization process enables client applications to gain limited access to online resources with permission of the resource owner. See the photo sharing example in the next section for an example. The OAuth specifications use JSON data structures to represent structured data.

The OAuth 2.0 Authorization Framework

The OAuth 2.0 Authorization Framework[21] enables third-party applications to be granted limited access to an HTTP service on behalf of an end user by orchestrating an approval interaction between the end user and the HTTP service. This means, for instance, that I don't have to give an application my password on my photo site for it to be able to access my photos there for me and I don't have to give it the ability to change my photos just to read them. This specification is widely deployed on the Web and mobile devices today. Whenever you install an application on your phone and give it permission to access resources on your behalf, you're actually using OAuth. Likewise, both OpenID Connect and Facebook Connect[22] are built using OAuth 2.0.

A rich suite of complementary and interoperable digital identity standards using JSON data structures and RESTful communication patterns has emerged and is in increasingly widespread use. These protocols retain much of the semantic richness of previous standards, while being easier to use across a broad range of Web development tools and platforms.

The OAuth 2.0 Authorization Framework: Bearer Token Usage

OAuth 2.0 Authorization Framework: Bearer Token Usage[23] enables clients to access protected resources by obtaining an access token, rather than using the resource owner's credentials. Access tokens are issued to clients by an authorization server with the approval of the resource owner. The client uses the access token to access the protected resources hosted by the resource server. This specification describes how to make protected resource requests when the OAuth 2.0 access token is a bearer token. A bearer token is usable by any party in possession of it.

JWT Profile for OAuth 2.0 Client Authentication and Authorization Grants

JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants[24] defines the use of a JWT bearer token as a means of requesting an OAuth 2.0 access token. It also defines how to use a JWT to authenticate an OAuth 2.0 client. For instance, this specification is used by OpenID Connect.

CONTINUED »

OpenID Connect Specifications

The OpenID Connect specifications enable Facebook Connect-like functionality from an open set of identity providers while also addressing some of the limitations of the OpenID 2.0 specifications. Put another way, it enables you to log into a relying party using a digital identity at an identity provider of your choice. These specifications build upon OAuth 2.0, JWT, JWS, JWE, JWK, JWA, and WebFinger. An explicit design point for the OpenID Connect protocols is enabling agents working on users' behalf, including browsers and mobile applications, to mediate users' identity interactions.

The OpenID Connect specifications were completed in February 2014. They are in production use by many organizations, including Google, Microsoft, Yahoo! Japan, Deutsche Telekom, Ping Identity, and Salesforce. For instance, when you're signing into Google+ or using Azure Active Directory, you're actually using OpenID Connect.

CONCLUSIONS

A rich suite of complementary and interoperable digital identity standards using JSON data structures and RESTful communication patterns has emerged and is in increasingly widespread use. These protocols retain much of the semantic richness of previous standards, while being easier to use across a broad range of Web development tools and platforms.

These protocols are being designed with an explicit awareness of the capabilities of modern browsers and Web development tools, including JSON support. Indeed, the designers believe that the already widespread adoption of these JSON-based digital identity standards demonstrates their usefulness for providing practical interoperable digital identity solutions. | SP | doi: 10.3789/isqv26no3.2014.05

MICHAEL B. JONES (mbj@microsoft.com) is an Identity Standards Architect at Microsoft and the author of the blog *Self-Issued: Musings on Digital Identity* (http://self-issued.info/).

REFERENCES

1. Neuman, B. Clifford, and Theodore Ts'o. "Kerberos: An Authentication Service for Computer Networks." *IEEE Communications Magazine*, September 1994, 32 (9): 33–38. http://dx.doi.org/10.1109/35.312841
2. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280. Internet Engineering Task Force, March 2, 2013. http://www.rfc-editor.org/rfc/rfc5280.txt
3. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 15, 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
4. *Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)*. OASIS Standard 200401, March 2004. http://docs.oasis open.org/wss/2004/01/oasis 200401-wss-soap-message-security-1.0.pdf
5. *WS-Trust 1.4*. OASIS Standard, February 2009. http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.html
6. *WS-SecurityPolicy 1.3*. OASIS Standard, February 2009. http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.3/os/ws-securitypolicy-1.3-spec-os.html
7. *OpenID Authentication 2.0*. OpenID Final Specification, December 5, 2007. http://openid.net/specs/openid-authentication-2_0.html
8. *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*. ITU-T X.690. International Telecommunication Union, July 2002. http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf
9. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C Recommendation, November 26, 2008. http://www.w3.org/TR/2008/REC-xml-20081126/
10. *The JavaScript Object Notation (JSON) Data Interchange Format*. IETF RFC 7159, March 2014. http://www.rfc-editor.org/rfc/rfc7159.txt
11. Fielding, Roy Thomas. "Representational State Transfer (REST)." In: *Architectural Styles and the Design of Network-based Software Architectures*, Chapter 5. Ph.D. Dissertation. University of California, Irvine, 2000. http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
12. *JSON Web Token (JWT)*. IETF Internet-Draft, October 24, 2014. http://www.ietf.org/id/draft-ietf-oauth-json-web-token-30.txt
13. *JSON Web Signature (JWS)*. IETF Internet-Draft, October 24, 2014. http://www.ietf.org/id/draft-ietf-jose-json-web-signature-36.txt
14. *JSON Web Encryption (JWE)*. IETF Internet-Draft, October 24, 2014. http://www.ietf.org/id/draft-ietf-jose-json-web-encryption-36.txt
15. *OpenID Connect Core 1.0*. OpenID Final Specification, February 25, 2014. http://openid.net/specs/openid-connect-core-1_0.html
16. Web Authorization Protocol (oauth) Working Group [webpage]. http://datatracker.ietf.org/wg/oauth/charter/
17. Javascript Object Signing and Encryption (jose) Working Group [webpage]. https://datatracker.ietf.org/wg/jose/charter/
18. *JSON Web Key (JWK)*. IETF Internet-Draft, October 24, 2014. http://www.ietf.org/id/draft-ietf-jose-json-web-key-36.txt
19. *JSON Web Algorithms (JWA)*. IETF Internet-Draft, October 24, 2014. http://www.ietf.org/id/draft-ietf-jose-json-web-algorithms-36.txt
20. *WebFinger*. IETF RFC 7033, September 2013. http://www.rfc-editor.org/rfc/rfc7033.txt
21. *The OAuth 2.0 Authorization Framework*. IETF RFC 6749, October 2012. http://www.rfc-editor.org/rfc/rfc6749.txt
22. Morin, Dave. *Announcing Facebook Connect*. Facebook Developers Blog, May 9, 2008. https://developers.facebook.com/blog/post/2008/05/09/announcing-facebook-connect/
23. *The OAuth 2.0 Authorization Framework: Bearer Token Usage*. IETF RFC 6750, October 2012. http://www.rfc-editor.org/rfc/rfc6750.txt
24. *JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants*. IETF Internet-Draft, October 21, 2014. http://www.ietf.org/id/draft-ietf-oauth-jwt-bearer-11.txt