Course by
Consortium GARR
and
Reti
Business & IT Consulting

JSON Web Token (JWT) overview

OIDC primer - a course on OpenID Connect

Credits: Roland Hedberg, Ioannis Kakavas

JWT in the context of OIDC

The **OpenID Connect** protocol, as we have seen, is a simple REST/JSON- based identity federation protocol layered on OAuth 2.0.

It uses the **JSON Web Token (JWT)** and **JSON Object Signing and Encryption (JOSE)** formats both to represent security tokens and to provide security for other protocol messages.

In particular, to guarantee security:

- JWT is used to **represent information** (i.g. to describe a user identity)

JSON Web Token (JWT)

JSON Web Token (JWT) is a **transport format** to represent a set of claims as JSON object.

It has the characteristic of being:

- **Versatile**, you can represent all information you need
- **Compact**, the representation does not add significant overhead
- **URL-safe**, the string obtained can be passed in URLs or HTTP headers

JWT Representation

JWT token is a **sequence of URL-safe parts** separated by period ‘.’ characters.

The main parts commonly are:

- Header
- Payload
- Digital signature

```
eyJhbGciOiJSUzI1NiIsImtpZCI6Imkwd25uIn0.eyJzdWIiOiJqb2UiLCJhdWQiOiJ
pbV9vaWNfY2xpZW50IiwianRpIjoidWY5MFNLNHdzY0ZoY3RVVDZEdHZiMiIs
ImlzcyI6Imh0dHBzOlwvXC9sb2NhbGhvc3Q6OTAzMSIsImlhdCI6MTM5NDA2
MDg1MywiZXhwIjoxMzk0MDYxMTUzLCJub25jZSI6ImU5NTdmZmJhLTlhNzg
tNGVhOS04ZWNhLWFlOGM0ZWY5Yzg1NiIsImF0X2hhc2giOiJ3Zmd2bUU5V
nhqQXVkc2w5bGM2VHFBIn0.lr4L-oT7DJi7Re0eSZDstAdOKHwSvjZfR-OpdW
SOmsrw0QVeI7oaIcehyKUFpPFDXDR0-RsEzqno0yek-_U-Ui5EM-yv0PiaUOm
JK1U-ws_C-fCplUFSE7SK-TrCwaOow4_7FN5L4i4NAa_WqgOjZPloT8o3kKyTk
BL7GdITL8rEe4BDK8L6mLqHJrFX4SsEduPk0CyHJSykRqzYS2MEJlncocBBI4up
5Y5g2BNEb0aV4VZwYjmrv9oOUC_yC1Fb4Js5Ry1t6P4Q8q_2ka5OcArlo188X
H7lMgPA2GnwSFGHBhccjpxhN7S46ubGPXRBNsnrPx6RuoR2cI46d9ARQ
```

JWT Representation - Header

Value	Value Decoded

eyJhbGciOiJSUzI1NiIsImtpZCI6Imkwd25uIn0	{
	"alg": "RS256",
	"kid": "i0wnn"
	}

JWT Representation - Payload

Value	Value Decoded

eyJzdWIiOiJqb2UiLCJhdWQiOiJpbV9vaWNfY2xpZW50IiwianRpIjoidWY5MFNLNHdzY0ZoY3RVVDZEdHZiMiIsImlzcyI6Imh0dHBzOlwvXC9sb2NhbGhvc3Q6OTAzMSIsImlhdCI6MTM5NDA2MDg1MywiZXhwIjoxMzk0MDYxMTUzLCJub25jZSI6ImU5NTdmZmJhLTlhNzgtNGVhOS04ZWNhLWFlOGM0ZWY5Yzg1NiIsImF0X2hhc2giOiJ3Zmd2bUU5VnhqQXVkc2w5bGM2VHFBIn0	{
	"sub": "joe",
	"aud": "im_oic_client",
	"jti": "uf90SK4wscFhctUT6Dtvb2",
	"iss": "https:\/\/localhost:9031",
	"iat": 1394060853,
	"exp": 1394061153,
	"nonce": "e957ffba-9a78-4ea9-ae8c4ef9c856",
	"at_hash": "wfgvmE9VxjAudsl9lc6TqA"
	}

JWT Representation - Digital signature

Value	Value Decoded
lr4L-oT7DJi7Re0eSZDstAdOKHwSvjZfR-OpdWSOmsrw0QVeI7oaIcehyKUFpPFDXDR0-RsEzqno0yek-_U-Ui5EM-yv0PiaUOmJK1U-ws_C-fCplUFSE7SK-TrCwaOow4_7FN5L4i-4NAa_WqgOjZPloT8o3kKyTkBL7GdITL8rEe4BDK8L6mLqHJrFX4SsEduPk0CyHJSykRqzYS2MEJlncocBBI4up5Y5g2BNEb0aV4VZwYjmrv9oOUC_yC1Fb4Js5Ry1t6P4Q8q_2ka5OcArlo188XH7lMgPA2GnwSFGHBhccjphN7S46ubGPXRBNsnrPx6RuoR2cI46d9ARQ	N/A *(signature represented as specified by JOSE)*

JSON Signing and Encryption (JOSE)

JOSE is a framework intended to provide a method to **securely transfer claims** (such as authorization information) between parties. The JOSE framework provides a collection of specifications to serve this purpose.

As by its name, JOSE deals with:

- Digital **signature** of claims
- **Encryption** of claims

JOSE Signature

JOSE permit to **describe the algorithms** used for signing as defined in the JSON Web Algorithm (JWA) specification.

In the “alg” field JOSE specifies the signature method used:

- **None**: no digital signature
- **HS256**: HMAC w/ SHA-256 hash
- **RS256**: RSA PKCS v1.5 w/ SHA-256 hash
- **ES256**: ECDSA w/ P-256 curve and SHA-256 hash
- ...

JOSE Encryption

JOSE permit to **describe** also **the algorithms** used for encryption as defined in the JSON Web Algorithm (JWA) specification.

In the “alg” field JOSE specifies the encryption method used:

- **None**: no digital signature
- **RSA1_5**: RSA 1.5
- **RSA-OAEP-256**: RSA Optimal Asymmetric Encryption Padding 256 bit
- **A256KW**: AES Keywrap w/ 256 key
- **dir**: direct encryption
- **ECDH-ES+A256KW**: EC Diffie Hellman Ephemeral+Static key agreement w/ AES256 key
- ...

JSON Web Key (JWK)

A JSON Web Key (JWK) is a JSON data structure that represents a **cryptographic key**.

Using a JWK rather than one or more parameters allows for a generalized key as input that can be applied to a number of different algorithms that may expect a different number of inputs.

Summary

The JW* standards permit to represent relevant information in a versatile, concise and URL safe manner.

1. **JWT** *(JSON Web Token)* is used to represent user information and ID tokens
2. **JWS** *(JSON Web Signature)* is used to sign the message
3. **JWE** *(JSON Web Encryption)* is used to describe encryption used for the message
4. **JWA** *(JSON Web Algorithms)* is used to describe the security algorithm used
5. **JWK** *(JSON Web Key)* is used to describe the key used by security algorithm

Q&A

Thanks for your attention!