Introduction to OpenID Connect Core Ch.7 – Self Issued Identity Provider

2018-05-15

Nat Sakimura(@_nat_en)

OpenID® Foundation

Chairman of the board

NRI

Research Fellow

*Unless otherwise noted, all the photos and vector images are licensed by GraphicStocks.

OpenID® is a registered trademark of the OpenID Foundation.
Have you read the Chapter 7 of OpenID Connect?
0.2.4. request_uri rationale

6.3. Validating JWT-Based Requests
   6.3.1. Encrypted Request Object
   6.3.2. Signed Request Object
   6.3.3. Request Parameter Assembly and Validation

7. Self-Issued OpenID Provider
   7.1. Self-Issued OpenID Provider Discovery
   7.2. Self-Issued OpenID Provider Registration
      7.2.1. Providing Information with the "registration" Request Parameter
   7.3. Self-Issued OpenID Provider Request
   7.4. Self-Issued OpenID Provider Response
   7.5. Self-Issued ID Token Validation

8. Subject Identifier Types
   8.1. Pairwise Identifier Algorithm

9. Client Authentication

10. Signatures and Encryption
   10.1. Signing
It is an IdP on your local machine

- I am the issuer of my “identity” therefor it will not be taken away

- Sounds a lot like “Self Sovereign Identity”, is it not?

- It does not need Blockchain, and does not leak information like current proposals that uses Blockchain.

- Wire-protocol-wise, it is OpenID Connect with a little twist.

- It can obviously use the platform supported Authenticator,
  - e.g. FIDO/WebAuthen supporting TEE through biometric unlocking.
Aggregated Claims

- Signed Claims
  - Data Source
  - Data Source
  - Self Issued IdP
  - Relying Party

Typically On your phone
Distributed Claims

Data Source

Data Source

Data Source

Signed Claims

Self Issued IdP

Permission

Relying Party

Typically On your phone
When Self Issued IdP is Supported by the client/RP, The RP should show an icon For it, e.g., a phone icon.

Self Issued Provider

Tap on it.
Since it is using Custom Scheme, I get asked if I want to open it.

Tap Open.
Use touch ID to unlock the private key in the keychain.

On Android, we can utilize TEE (Trusted Execution Environment).
And, you get a regular ID Token, which when decoded will be as on the right.

Hash of the public key that Belongs to the subject

Public key that belongs to the subject

```json
{
  "gender": "M",
  "iat": "2018-05-15T19:49:37.000Z",
  "family_name": "Sakimura",
  "nonce": "12p29on",
  "sub": "vrv-X0e69uJD3jvFtAFKgn-tF1fmSdqJknN5v34AJkI",
  "sub_jwk": {
    "kty": "RSA",
    "n": "AN8Yh9JyU1AnHpx01TKsv6AEqlyxyjHdH-ve13j-p-YfNvBw7az7zyAlftX_3l380HGNaHqfpysAUMIK8AAu5p843BM4i35d8mJ BartonGbnS0PpD2a66cYRrvTttS4gBjgbf1wA4wJo82r7MRliL1rwp
    "e": "AQAB"
  },
  "aud": "http://connect.openid4.us/eshop/sicallback.html",
  "exp": "2018-05-15T19:54:37.000Z",
  "updated_at": 1526413732,
  "iss": "https://self-issued.me"
}
```