

JWT CWT Status List

A simple and scalable credential revocation/status mechanism

Tobias Looker, Paul Bastian, Christian Bormann



The Problem

How to enable the issuer of a token (e.g CWT or JWT) to communicate dynamic status information about a token after it is issued and before it expires.

Example - A Verifiable Credential where the Issuer would like to communicate whether the credential is revoked or not.



Key Requirements

- Scalable -> Must scale to millions (100's millions) of credentials
- Issuer Herd Privacy -> Able to protect verifiers and token subjects from issuer knowing where a given token is being verified/used
- Work with common formats -> Support JOSE/COSE based tokens/credentials
- Caching Support -> Enable verifying parties to cache status lists for offline verification



Proposed Solution

- Byte array based status list (for large amounts of credentials)
- Status is indicated by the value of a specific index in the status list
- Status List is Gzip-compressed and the outcome base64 encoded
- Signed and delivered as JWT/CWT



Example: Referenced JWT

```
{
  "alg": "ES256",
  "kid": "11"
}
.
{
  "iss": "https://example.com",
  ...
  "status": {
    "uri": "https://example.com/statuslists/1",
    "idx": 5
  }
}
```

URI of the status list token

Index in the status list



Example: Status List JWT

```
eyJhbGciOiJIUzI1NiIsImtpZCI6IjEyIiwidHlwIjoiYm9ja3RhdHVzIGlzdCtqd3QifQ.eyJleHAiOjE2MDc1MTc3NzAsImVhdCI6MTY0NjkxMjk3MCwiaXNzIjoiaHR0cHM6Ly9leGFtcGxlLmNvbSIsInN0YXR1c19saXN0Ijp7ImJpdHMiOjIsImxzdCI6Ikg0c0lBTW9faKdRQ196dnA4aE1BWkxSTE1RTUFBQUEifSwic3ViIjoiaHR0cHM6Ly9leGFtcGxlLmNvbS9zdGF0dXNsaXN0cy8xIn0.8uaUXshaJdG  
WGjvwPwaa2Gtt0M7-M7dG09rXaz3x99LCdG5tKb-ARL1ezqguLT  
s63VeudYWqpdg4HpN-D2h0kg
```

```
{  
  "alg": "ES256",  
  "kid": "12",  
  "typ": "statuslist+jwt"  
}  
.  
{  
  "exp": 1687517770,  
  "iat": 1686912970,  
  "iss": "https://example.com",  
  ... //other claims  
  "status_list": {  
    "bits": 1,  
    "lst": "H4sIAMo_jGQC_zvp8hMAZLRLMQMAAAA"  
  },  
  "sub": "https://example.com/statuslists/1"  
}
```

Example: How it fits together

```
"status": {  
  "idx": 5  
  "uri": "https://example.com/statuslists/1",  
}
```

```
"sub": "https://example.com/statuslists/1"  
"status_list": {  
  "bits": 1,  
  "lst": "H4sIAMo_jGQC_zvp8hMAZLRLMQMAAAA"  
}
```

0x0 = VALID
0x1 = INVALID

1	0	0	1	0	1	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---

Deflate gzip



Usecases

- Any digital/verifiable credential
 - SD-JWT (eIDAS 2.0)
 - ISO mdoc (mdl)
- Any other long-lived token



Why not other approaches

- Current Accumulator/ZKP-based approaches provide better privacy characteristics but don't scale well
- X.509 Certificate Revocation Lists don't scale well
- OCSP/Validity credentials reveal information directly to the Issuer



Work in Progress

- CWT presentation is still in progress
- Security & Privacy considerations in progress
- Testing the current specification with implementations
- And much more...



Questions?



Links

- Current Editors Copy ->
<https://vcstuff.github.io/draft-looker-oauth-jwt-cwt-status-list/draft-looker-oauth-jwt-cwt-status-list.html>
- Git Repository ->
<https://github.com/vcstuff/draft-looker-oauth-attestation-based-client-auth>