Financial-grade API: JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)

Torsten Lodderstedt, Brian Campbell

IETF-103
Nov 5 2018

Introduction

- In some use cases there is a need to sign authorization requests and responses
- Although we have JWT Secured Authorization Request (JAR), we don't have signatures for authorization responses (yet)
- In the wild: hybrid response types and ID tokens as detached signature
  - Requires OpenID Connect stack to perform pure API authorization
  - Makes "real" OpenID Connect harder to implement
- OpenID Foundation's FAPI Working Group came up with JARM

JWT secured authorization response mode (JARM)

- Authorization response parameters (+ extra data) are encoded in a JWT
- JWT is signed and (optionally) encrypted
- Response mode*, can be combined with any response type and w/ OIDC

Example for response type “code”

```
{
   "iss":"https://accounts.example.com",
   "aud":"s6BhdRkqt3",
   "exp":1311281970,
   "code":"PyyFaux2o7Q0YfXBU32jhw.5FXSQpvr8akv9CeRDSd0QA",
   "state":"S8NJ7uqk5fY4EjNvP_G_FtyJu6pUsvH9jsYni9dMAJw"
}
```

*https://openid.net/specs/oauth-v2-multiple-response-types-1_0.html

Request

```
GET /authorize?response_type=code&client_id=s6BhdRkqt3
    &state=S8NJ7uqk5fY4EjNvP_G_FtyJu6pUsvH9jsYni9dMAJw
    &redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
    &response_mode=jwt
HTTP/1.1
    Host: server.example.com
```

Response

```
HTTP/1.1 302 Found
Location: https://client.example.com/cb?
response=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJodHRwczovL2FjY291bnRzL
m
V4YW1wbGUuY29tIiwiYXVkIjoiczZCaGRSa3F0MyIsImV4cCI6MTMxMTI4MTk3MCwiY29kZSI6IlB5e
U
ZhdXgybzdRMFlmWEJVMzJqaHcuNUZYU1FwdnI4YWt2OUNlUkRTZDBRQSIsInN0YXRlIjoiUzhOSjd1c
W
s1Zlk0RWpOdlBfR19GdHlKdTZwVXN2SDlqc1luaTlkTUFKdyJ9.HkdJ_TYgwBBj10C-aWuNUiA062Am
q
2b0_oyuc5P0aMTQphAqC2o9WbGSkpfuHVBowlb-zJ15tBvXDIABL_t83q6ajvjtq_pqsByiRK2dLVdU
w
KhW3P_9wjvI0K20gdoTNbNlP9Z41mhart4BqraIoI8e-L_EfAHfhCG_DDDv7Yg
```

Processing

1. (Optional) Decrypt
2. Check state, iss, aud & exp
3. Check signature


```
{
   "iss":"https://accounts.example.com",
   "aud":"s6BhdRkqt3",
   "exp":1311281970,
   "code":"PyyFaux2o7Q0YfXBU32jhw.5FXSQpvr8akv9CeRDSd0QA",
   "state":"S8NJ7uqk5fY4EjNvP_G_FtyJu6pUsvH9jsYni9dMAJw"
}
```

JARM provides

- signing and encryption,
- sender authentication,
- audience restriction,
- protection from replay,
- protection from credential leakage, and
- protection from mix-up attacks

in a single mechanism!