draft-fett-oauth-dpop-03

OAuth 2.0 **D**emonstration of **P**roof-**o**f-**P**ossession at the Application Layer (**DPoP**)

IETF 106
Singapore
Nov 2019

Brian Campbell (presenter, co-author, workation photographer)
Torsten Lodderstedt (co-author)
Daniel Fett (co-author)

John Bradley (co-author of sorts)
Michael Jones (co-author)
David Waite (co-author)

Executive Summary

DPoP is a draft proposal for a new[ish], simple and concise approach to proof-of-possession for OAuth access and refresh tokens using application-level constructs and leveraging existing library support

-00 was published during IETF 105 in Prague thereby justifying the use of this photo

Prior proof-of-possession efforts in OAuth:

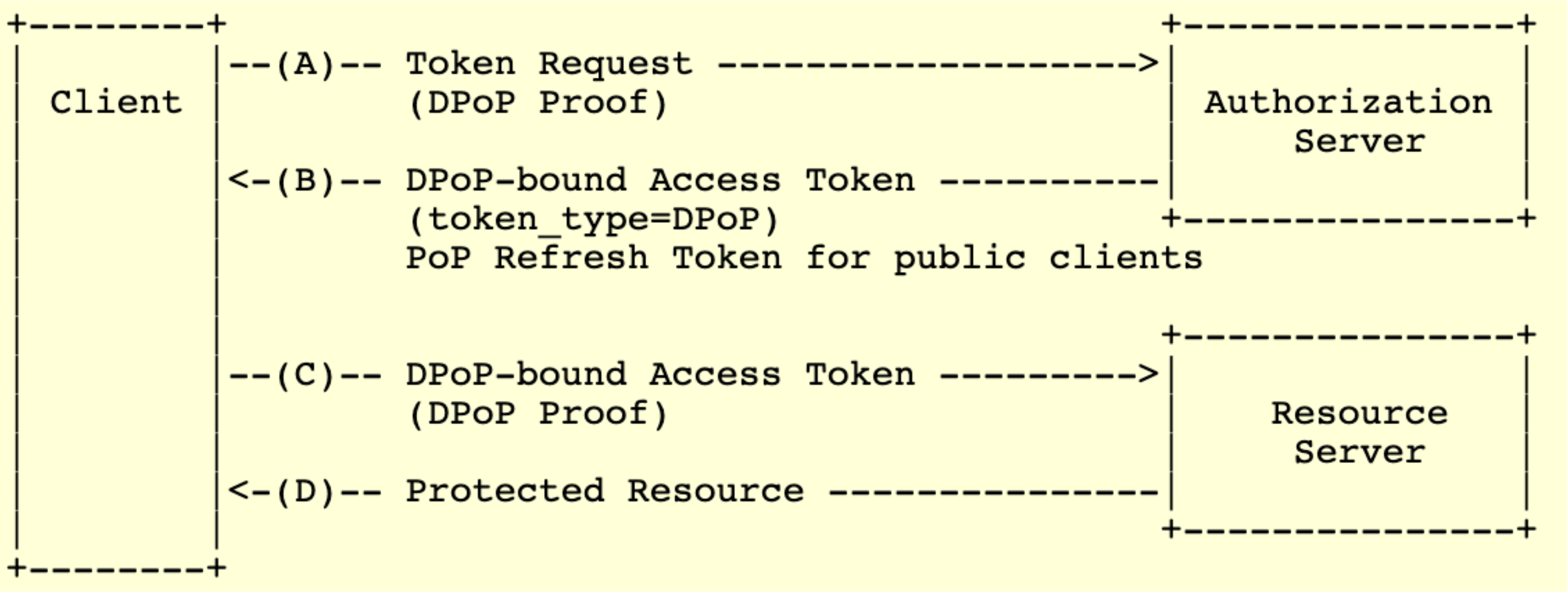
The road to now is littered with [to varying degrees] failures

- **“OAuth 1.0a”** - RFC 5849
- **“OAuth 2.0 Message Authentication Code (MAC) Tokens”** - draft-ietf-oauth-v2-http-mac
- **“Proof-of-Possession Key Semantics for JSON Web Tokens”** – RFC 7800
- **“OAuth 2.0 Proof-of-Possession (PoP) Security Architecture”** - draft-ietf-oauth-pop-architecture
- **“OAuth 2.0 Proof-of-Possession: Authorization Server to Client Key Distribution”** - draft-ietf-oauth-pop-key-distribution
- **“A Method for Signing HTTP Requests for Oauth”** – draft-ietf-oauth-signed-http-request
- **“OAuth 2.0 Token Binding”** - draft-ietf-oauth-token-binding
- **“OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens”** - draft-ietf-oauth-mtls

Motivations for this new effort

- Be better than bearer (be best…?)
- OAuth 2.0 Security BCP recommends use of sender-constrained tokens (somewhat aspirational)
  - To prevent token replay at a different endpoint/resource (among other benefits)
- Yet OAuth lacks suitable and widely-applicable PoP mechanism
- Especially true for Single Page Applications (SPA)
  - MTLS for OAuth 2.0 would have major UX issues with SPAs
  - Status of Token Binding is uncertain
- Proof-of-possession bound refresh tokens for public clients

Basic DPoP flow in ASCII



```
+--------+                                          +---------------+
|        |--(A)-- Token Request ------------------->|               |
| Client |        (DPoP Proof)                      | Authorization |
|        |                                          |     Server    |
|        |<-(B)-- DPoP-bound Access Token ----------|               |
|        |        (token_type=DPoP)                 +---------------+
|        |        PoP Refresh Token for public clients
|        |
|        |                                          +---------------+
|        |--(C)-- DPoP-bound Access Token --------->|               |
|        |        (DPoP Proof)                      |    Resource   |
|        |                                          |     Server    |
|        |<-(D)-- Protected Resource ---------------|               |
|        |                                          +---------------+
+--------+
```

Anatomy of a DPoP Proof JWT

```
{
 "typ":"dpop+jwt",
 "alg":"ES256",
 "jwk":
 {
   "kty":"EC", "crv":"P-256"
   "x":"l8tFrhx-34tV3hRICRDY9zCkDlpBhF42UQUfWVAWBFs",
   "y":"9VE4jf_Ok_o64zbTTlcuNJajHmt6v9TDVrU0CdvGRDA"
 }
}
{
 "jti":"-BwC3ESc6acc2lTc",
 "htm":"POST",
 "htu":"https://server.example.com/token",
 "iat":1562262616
}
```

Explicitly typed

Asymmetric signature algorithms only

The public key for which proof-of-possession is being demonstrated

Minimal info about the HTTP request (method & URI)

Unique identifier for replay checking

Only valid for a limited time window relative to creation time

Other stuff *could* go here

Access Token Request

```
POST /token HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
DPoP: eyJ0eXAiOiJkcG9wK2p3dCIsImFsZyI6IkVTMjU2IiwiandrIjp7Imt0eSI6Ik
 VDIiwieCI6Imw4dEZyaHgtMzR0VjNoUklDUkRZOXpDa0RscEJoRjQyVVFVZldWQVdCR
 nMiLCJ5IjoiOVZFNGpmX09rX282NHpiVFRsY3VOSmFqSG10NnY5VERWclUwQ2R2R1JE
 QSIsImNydiI6IlAtMjU2In19.eyJqdGkiOiItQndDM0VTYzZhY2MybFRjIiwiaHRtIj
 oiUE9TVCIsImh0dSI6Imh0dHBzOi8vc2VydmVyLmV4YW1wbGUuY29tL3Rva2VuIiwia
 WF0IjoxNTYyMjYyNjE2fQ.2-GxA6T8lP4vfrg8v-FdWP0A0zdrj8igiMLvqRMUvwnQg
 4PtFLbdLXiOSsX0x7NVY-FNyJK70nfbV37xRZT3Lg
grant_type=authorization_code
&code=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
&code_verifier=bEaL42izcC-o-xBk0K2vuJ6U-y1p9r_wW2dFWIWgjz-
```

DPoP proof JWT
in HTTP header

Access Token Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store

{
   "access_token":"eyJhbGciOiJFUzI1NiIsImtpZCI6IkJlQUxrYiJ9.eyJzdWIiOi
     Jzb21lb25lQGV4YW1wbGUuY29tIiwiaXNzIjoiaHR0cHM6Ly9zZXJ2ZXIuZXhhbXB
     sZS5jb20iLCJhdWQiOiJodHRwczovL3Jlc291cmNlLmV4YW1wbGUub3JnIiwibmJm
     IjoxNTYyMjYyNjExLCJleHAiOjE1NjIyNjYyMTYsImNuZiI6eyJqa3QiOiIwWmNPQ
     09SWk5ZeS1EV3BxcTMwalp5SkdIVE4wZDJIZ2xCVjN1aWd1QTRJIn19.vsFiVqHCy
     IkBYu50c69bmPJsj8qYlsXfuC6nZcLl8YYRNOhqMuRXu6oSZHe2dGZY0ODNaGg1cg
     -kVigzYhF1MQ",
   "token_type":"DPoP",
   "expires_in":3600,
   "refresh_token":"4LTC8lb0acc6Oy4esc1Nk9BWC0imAwH7kic16BDC2",
}
```

Token type indicates that the access token is bound to the DPoP public key

DPoP Bound Access Token

JWT & Introspection Response

```
{
   "sub":"someone@example.com",
   "iss":"https://server.example.com",
   "aud":"https://resource.example.org",
   "nbf":1562262611,
   "exp":1562266216,
   "cnf":
   {
     "jkt":"0ZcOCORZNYy-DWpqq30jZyJGHTN0d2HglBV3uiguA4I"
   }
 }
```

Confirmation claim carries the SHA-256 JWK Thumbprint of the DPoP public key to which the access token is bound

Protected Resource Request

```
GET /protectedresource HTTP/1.1
Host: resource.example.org
Authorization: DPoP eyJhbGciOiJFUzI1NiIsImtpZCI6IkJlQUxrYiJ9.eyJzdWI
 iOiJzb21lb25lQGV4YW1wbGUuY29tIiwiaXNzIjoiaHR0cHM6Ly9zZXJ2ZXIuZXhhbX
 BsZS5jb20iLCJhdWQiOiJodHRwczovL3Jlc291cmNlLmV4YW1wbGUub3JnIiwibmJmI
 joxNTYyMjYyNjExLCJleHAiOjE1NjIyNjYyMTYsImNuZiI6eyJqa3QiOiIwWmNPQ09S
 Wk5ZeS1EV3BxcTMwalp5SkdIVE4wZDJIZ2xCVjN1aWd1QTRJIn19.vsFiVqHCyIkBYu
 50c69bmPJsj8qYlsXfuC6nZcLl8YYRNOhqMuRXu6oSZHe2dGZY0ODNaGg1cg-kVigzY
 hF1MQ
DPoP: eyJ0eXAiOiJkcG9wK2p3dCIsImFsZyI6IkVTMjU2IiwiandrIjp7Imt0eSI6Ik
 VDIiwieCI6Imw4dEZyaHgtMzR0VjNoUklDUkRZOXpDa0RscEJoRjQyVVFVZldWQVdCR
 nMiLCJ5IjoiOVZFNGpmX09rX282NHpiVFRsY3VOSmFqSG10NnY5VERWclUwQ2R2R1JE
 QSIsImNydiI6IlAtMjU2In19.eyJqdGkiOiJlMWozV19iS2ljOC1MQUVCIiwiaHRtIj
 oiR0VUIiwiaHR1IjoiaHR0cHM6Ly9yZXNvdXJjZS5leGFtcGxlLm9yZy9wcm90ZWN0Z
 WRyZXNvdXJjZSIsImlhdCI6MTU2MjI2MjYxOH0.lNhmpAX1WwmpBvwhok4E74kWCiGB
 NdavjLAeevGy32H3dbF0Jbri69Nm2ukkwb-uyUI4AUg1JSskfWIyo4UCbQ
```

DPoP public key bound access token

DPoP proof

Document History and Status

(and workation slideshow)

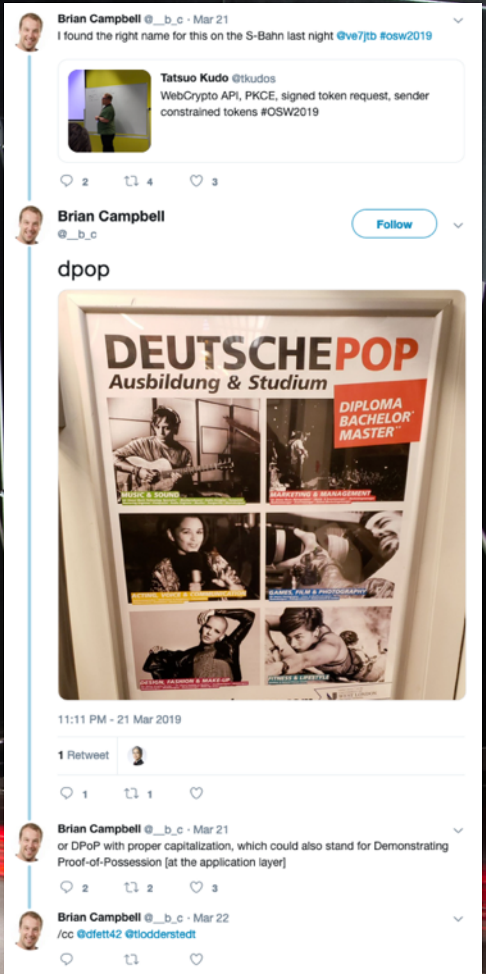
They'll tell the story of tonight

OAuth Security Workshop
Stuttgart*
March 2019

* Took the train from Frankfurt

backstory on the "shiny name"*

Near Darmstadt on the eve of the 2015 OAuth Security Workshop



Brian Campbell @__b_c · Mar 21
I found the right name for this on the S-Bahn last night @ve7jtb #osw2019

Tatsuo Kudo @tkudos
WebCrypto API, PKCE, signed token request, sender constrained tokens #OSW2019

Brian Campbell
@__b_c

dpop

11:11 PM - 21 Mar 2019

1 Retweet

Brian Campbell @__b_c · Mar 21
or DPoP with proper capitalization, which could also stand for Demonstrating Proof-of-Possession [at the application layer]

Brian Campbell @__b_c · Mar 22
/cc @dfett42 @tlodderstedt

2019 OAuth Security Workshop

*Hannes https://youtu.be/tUmT5qqlKik?t=4178

IETF
#104

We’ll always have Prague

- -00 quickly published & presented
- some interest expressed
- just an individual draft (with all the authority thereby bestowed upon it*)

* https://tools.ietf.org/html/draft-abr-twitter-reply-00

- -01/**-02** published & presented
- interest again expressed
- yet remains an individual draft

- "… and running code."
  - Node AS - https://github.com/panva/node-oidc-provider
  - Go library - https://github.com/pquerna/dpop
  - Running demo - https://murmuring-journey-60982.herokuapp.com
  - Java JWT library API enhancements - https://bitbucket.org/b_c/jose4j

Vive la Canada!

IETF #105

Montreal

IETF #106 Singapore

- -03 of the individual draft published
  - smaller tokens via “htm”, “htu”, and “jkt” rather than “http_method”, “http_uri”, and “jkt#S256” respectively
  - clarify/fix “jti” uniqueness requirements in DPoP proof

You are ¼ mile over this way

Advance praise for DPoP

"I have a client that is very keen on binding tokens but not so keen on MTLS [… and …] is pushing me quite hard for DPoP"
– anonymous consultant

"lightweight... application level only... existing libraries"
– unnamed speaker at Vancouver Identity Meetup

"interesting work... lot of potential"
–unspecified Identiverse keynote speaker pictured here

"what's your take on it? To me it seems simple and very sensible... how soon do you think it might actually turn into something real?"
– anonymous colleague

"very simple, very concise"
– unnamed co-author

"very enthusiastic about the new proposal [… that …] represents a significant advance in OAuth 2.0"
– unnamed mailing list participant

opportunities for further discussion

- Asymmetric cryptography is not super fast
- Threat model and stated objectives are a bit loose
- Specific claims
- ‘jti’ tracking isn’t always as easy as it seems
- Error code(s) and/or metadata
- MTI and/or algorithm discovery/negotiation

Next Steps

Before IETF #107 in Vancouver

Humbly request that the WG consider a call for adoption!