When last we met
(in person at #106 in Singapore)
after the interim is this other interim
[Some] Motivations for [D]PoP

- Do something that’s better than bearer
- OAuth 2.0 Security BCP (somewhat aspirationally) recommends use of “sender-constrained” tokens as do various FAPI profiles
  - To prevent token (re)play at a different endpoint/resource (among other benefits)
- Proof-of-possession bound refresh tokens for public clients (also per Security BCP)
- Yet OAuth lacks suitable and widely-applicable PoP mechanism
  - MTLS is “Virtually undeployable [for] general purpose applications” – a WG participant
  - What else is there really?
- Especially lacking for Single Page Applications (SPA)
  - MTLS for OAuth 2.0 would have major UX issues with SPAs
  - Token Binding is dead in the water & needed fetch() API changes anyway
Some existing PoP efforts:

- OAuth 1.0a - RFC 5849
- The OAuth 2.0 Authorization Framework – RFC 6749
- 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
- The OAuth 2.0 Authorization Framework: JWT Pop Token Usage - draft-sakimura-oauth-jpop
- OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens – RFC 8705
- OAuth 2.0 Demonstration of Proof-of-Possession at the Application Layer (DPoP) - draft-fett-oauth-dpop
- “a tentative suggestion for an alternative (to/in DPoP) design” – Neil Madden email
- Proof-of-Possession Tokens for OAuth Using JWS HTTP Signatures - draft-richanna-oauth-http-signature-pop
- Signing HTTP Requests via JSON Web Signatures - draft-richanna-http-jwt-signature
- Signing HTTP Messages - draft-richanna-http-message-signatures formerly draft-cavage-http-signatures
Criticisms of DPoP (paraphrased)

- It’s not draft-ietf-oauth-pop-key-distribution
- An asymmetric crypto operation on every single HTTP request is too expensive
- Tracking `jti` is prohibitive at scale
- Bit of a Rorschach Test even amongst its supporters
Where to now?

- Stay the course
  - Something between doing nothing and -pop-key-distribution + some HTTP signing

- Push forward and adopt and tweak DPoP
  - “… for us mere mortals, DPoP is fine as-is”
  - “we need to sender constrain refresh tokens issued to SPAs yesterday.”

- Work toward an approach that’s similar(ish) to DPoP using asymmetric keys but with ECDH to amortize the cost of asymmetric crypto over *many* requests (riffing on Neil’s idea)
  - allowing for the aggregated/derived key (unique to client/RS or client/AS) to be non-exportable

- ? -> Profit
See all y’alls @ IETF #107 in Vancouver

So… Now What?

Gratuitous closing slide featuring the city where will meet together next *

* Some of us anyway pending governmental pandemic response intervention

IETF 107 @ Vancouver Hyatt Regency