OAuth2 PoP

Introduction to the basic principles of OAuth2, the components and inherent weaknesses of bearer tokens

- Components of typical OAuth2 architecture - authorization service, resource server, client, user/device, tokens
- Overview of stateless versus stateful access/refresh tokens
- Security of bearer tokens - resource server is only validating that the token, not who is presenting the token
- Stateful token validation generally looks like:
    - Call from RS to the AS
    - AS looks up opaque token reference in local store
    - checks the exp (expiration time) is valid
    - checks the nbf (not before time if set)
    - returns human readable form of token to show scopes and who access_token is issued to
    - RS can use the scope information to make informed authorization decision
- Stateless token validation generally looks like:
    - RS validates signature of the access_token
    - To check signature needs to know key used to sign (either shared secret for HMAC or JWK for RS etc)
    - Checks exp, nbf, scopes etc
- Proof of Possession introduces signature access_tokens
    - request for access_token also includes the cnf_key attribute
    - this is the base64 encoded public key in JWK (JSON Web Key) format
    - AS mints access_token with this cnf_key embedded
    - when token is presented to RS, they can retrieve the cnf_key (either locally if stateless token, or call back to AS if stateful)
    - once RS knows the public key of the requesting client, they can generate a challenge response to the requesting party
    - this CR protocol is open to implementation, but will generally be some sort of calculation request, using the public key to sign and/or encrypt the response
    - requiring the client response to prove they are in possession of the public keys corresponding private key, either to decrypt or sign a response
    - RS validates the response and if as expected access is granted
- Discussion on pro's and con's of PoP
    - requires key minting
    - requires extra hops on the RS/client interaction
    - challenge response protocol is not described in spec, so flexibility ↯ confusion of implementation