PIKA: Proof of Issuer Key Authority

OAuth interim meeting
June 4, 2024

R. Barnes
Cisco

S. Goldberg
BastionZero, Inc
(acquired by
Cloudflare)

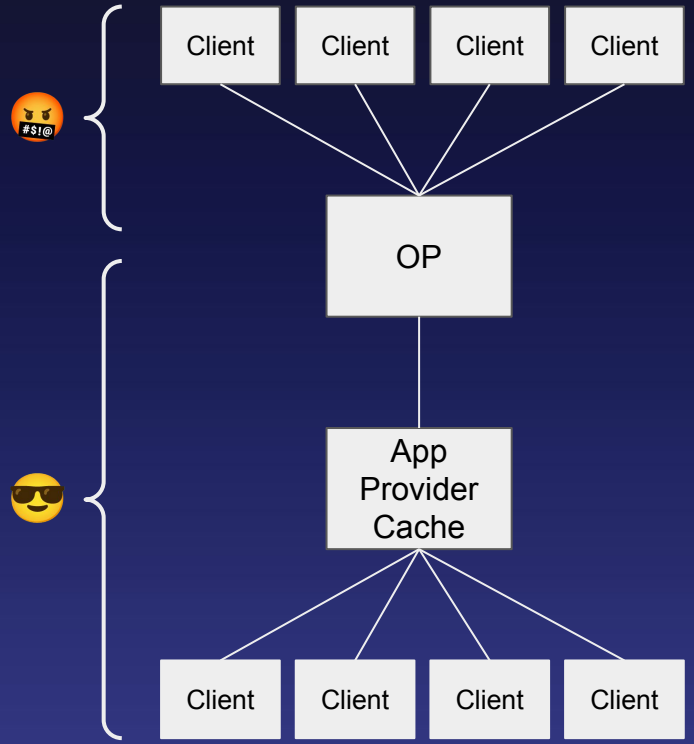
Today's JWT lifecycle can sometimes be problematic

What if JWK Set is unavailable?

- Key has been rotated
- Server is down

Use case: Applications using Verifiable Credentials

- Application software directly verifying signatures from OP when joining a session
  - No real caching of OP keys
  - Highly sensitive to OP downtime or latency
  - Many hits to JWK set endpoint from distributed relying parties
- Goal: Have app server fetch OP keys once and redistribute them to all relying parties
  - … without trusting the app server!



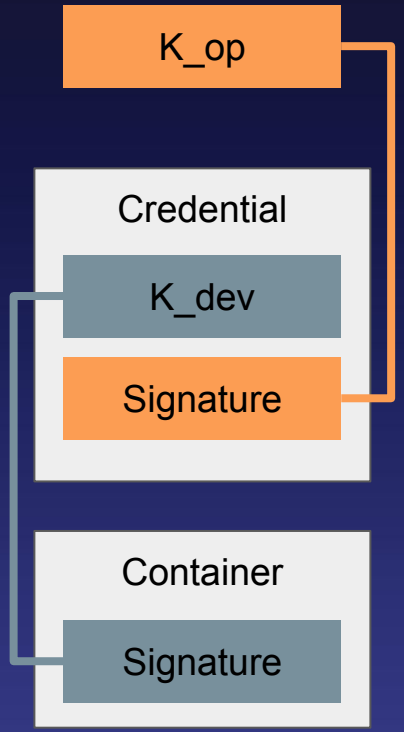
Use case: Durable Signing (e.g. of Containers)

Goal: App wants to know that at some past time T:

- Container was signed by Developer with K_dev
- OP signed a credential using K_op associating K_dev with Developer
  - E.g., VC or OpenPubkey PK Token

Solution (partial):

- Assume a time-stamping authority TSA attesting “X existed at time T”
- TSA can prove that K_op and credential *existed* at time T
- **How can we verify that the OP was using key K_op at time T? (*“issuer key authority”*)**



PIKA Chain of Trust

Given similar requirements to OpenID Federation, we re-use the Federation Historical Keys Response format as a base format for PIKAs.

Sample PIKA

```
JWT Header:
   {
     "alg": "ES256",
     "typ": "JWT",
     "x5c": ["MII..."]
   }

   JWT Payload:
   {
     "iss": "https://server.example.com",
     "iat": 123972394272,
     "exp": 124003930272,
     "keys":
       [
         {
           "kty": "EC",
           "crv": "P-256",
           "alg": "ES256"
           "x": "qiGKLwXRJmJR_AOQpWOHXLX5uYIfzvPwDurWvmZBwvw",
           "y": "ip8nyuLpJ5NpriZzCVKiG0TteqPMkrzfNOUQ8YzeGdk"
           "kid": "2HnoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs",
           "iat": 123972394872,
           "exp": 123974395972
         },
         {
           "kty": "RSA",
           "n": "ng5jr...",
           "e": "AQAB",
           "kid": "8KnoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMJJr",
           "iat": 123972394872,
           "exp": 123974394972
           "revoked": {
             "revoked_at": 123972495172,
             "reason": "keyCompromise",
             "reason_code": 1
           }
         }
       ]
   }

   JWT Signature:
   // Signature over JWT Header and Claims, as defined in RFC 7519
```

X.509 certificates chain to root of WebPKI

Validity interval of the PIKA

OP key

Validity interval for *signing* with OP's key

Another OP key

Signature using key in X.509 cert

Results of Adoption Call v1

Core value / interest in solving the problem: ✅

Issue 1: Use of PKI at the application layer (“x5c”)

- … as opposed to at the transport layer, for HTTPS
- PIKA bucks the long term trend away from X.509 at the application layer
  - JWK, OpenID Connect, OpenID Federation, etc.
- Need to make sure that systems using PIKA have a clear upgrade/interop path to alternatives to application-level certificates (e.g., OpenID Federation)

Issue 2: Security considerations using Web PKI certificates for PIKA signing

- Web servers might have different risks than PIKA-signing servers
- Compromise of a web server could allow the issuer to sign a PIKA
- Solution space: domain name prefix, lifetime limitation, …

Adoption Call v2?

Core value / interest in solving the problem: ✅

Do we have enough clarity to believe the issues are solvable?