OIDC FEDERATION

GOALS

- Bootstrap trust by using a trusted 3rd party.
- SSL security not the only protection.
    - Signed ’metadata’
- Allow for semi-dynamic behavior
    - Software statements

RELYING PARTY OPERATOR (RPO)

- Create key pair (A)
- Collect information to be in the software statement (SS)
  - includes pub_A and *redirect_uris*
- Send SS proposal to Federation operator (FO)
- FO verifies the SS and possibly adds extra FO info
- FO signs SS using $\mathrm{priv}_{\mathrm{FO}}$ returns it to the Owner

OPENID CONNECT PROVIDER OPERATOR

- OpenID Connect Provider Operator (OPO), creates a long lived signing key pair; call it *B*
- OPO submits registration data to Federation Operator (FO). The registration data MUST include *issuer* and pub_B
- FO returns a signed (with priv_{FO}) software statement, SS_{OP}, containing the submitted registration data, and any applied policy restrictions (*response_types*, signing/encryption algorithms …).

KEY INITIALIZATION

- To allow for key rotation in multiple steps, an intermediate key is used for signing. The keys in the JWKS could be rotated on a timescale of once every 24 hours, while the intermediate key could be rotated on timescale of once every month (the long-lived key can't be rotated at all).

A -- sign --> *JWK(pub(An))* -- sign --> JWKS

RELAYING PARTY

- Create a JSON Web Key Set (JWKS) and publish it at a URL specified by *jwks_uri* in the client metadata sent in the Registration Request.
- Create a new intermediate signing key pair, call it A_n and sign the JWK representation of $pub(A_n)$ with A.
- Sign the JWKS with priv$_{An}$.
- The URL specified by *signed_jwks_uri* contains a signed (JWS) version of the JWKS found at *jwks_uri*

OPENID CONNECT PROVIDER

- Create a JSON Web Key Set (JWKS) and publish it at a URL specified by *jwks_uri* in the provider metadata sent in the response to a discovery request.
- Create a new intermediate signing key pair, call it B_n and sign the JWK representation of $pub(B_n)$ with B.
- Sign the JWKS with priv_{Bn}.
- The URL specified by *signed_jwks_uri* contains a signed (JWS) version of the JWKS found at *jwks_uri*

DISCOVERY

The OP responds with its provider configuration and the following additional metadata parameters:

- *Software statements*: a list of software statements from all federations the OP is part of.
- *signed_metadata*: a JWS containing all published metadata, except signed_metadata.
- *signed_jwks_uri*: a URI to the location where the OP publishes the signed JWKS, SHOULD return the Content-Type 'application/jose' to indicate that the JWKS is in the form of a JWS using the JWS Compact Serialization.
- *signing_key*: a JWK containing the OP's intermediate public key pub(Bn).

REGISTRATION

The RP makes a standard client registration request that includes the following extra parameters:

- *software statements*: a list of software statements from all federations the RP is part of.
- *signing_key*: a JWK containing the RP's intermediate public key pub_{An}.
- *signed_jwks_uri*: a URI to the location where the RP publishes the signed JWKS, SHOULD return the Content-Type 'application/jose' to indicate that the JWKS is in the form of a JWS using the JWS Compact Serialization.