Encrypted Authorize Response

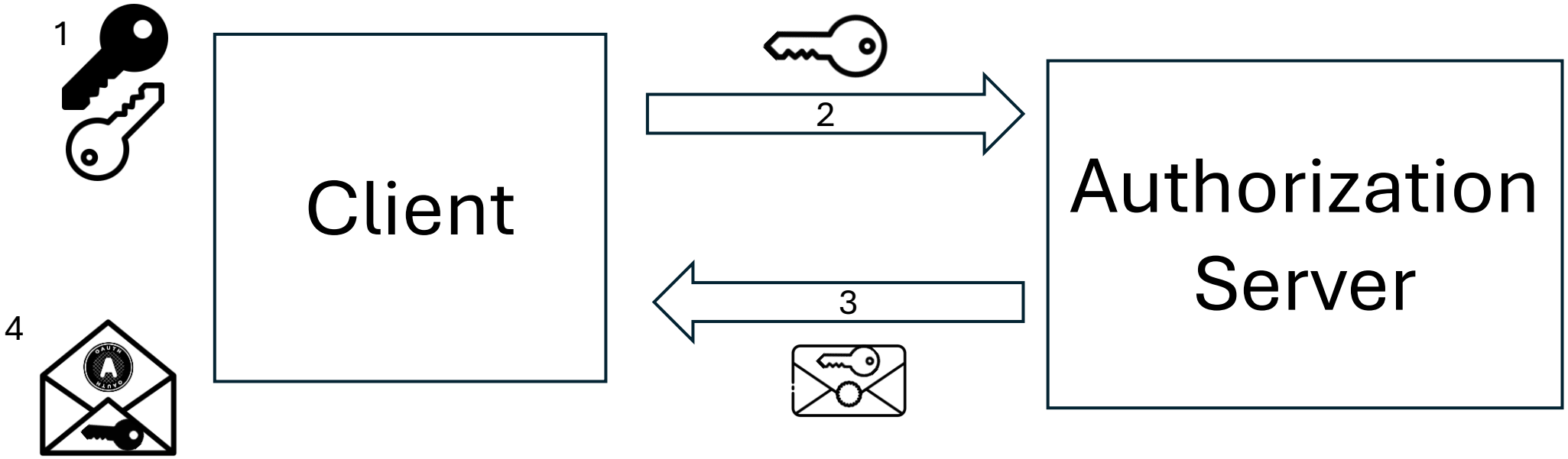
The challenge

- Authorization Code Flow with PKCE is great!
  - Avoids challenges with Implicit Grant Flow
  - Prevents User Agents from access Access Tokens
  - PKCE protects against Authorization Code interception and CSRF attacks

- But….
  - It requires an extra roundtrip
  - Impacts cost (at scale) and user experience (latency and page load times)

What if... (the idea)

1. Client generates ephemeral key material
2. Client sends key material to the Authroization Server as part of the authorizations request
3. Authorization Server returns an encrypted response to client (including the Access Token)
4. Client decrypts the authorization response using ephemeral key to obtain the Access Token



Some Security Questions

- Credential Leakage via Referrer Headers
  - Similar level of protection to authorization code flow?
- Credential Leakage via Browser History
  - Similar level of protection to authorization code flow?
- Access Token Injection
  - Is this fully prevented by encrypting the authorization response?
- Sender Constrained Tokens
  - Could DPoP be adopted/used with such a flow?
  - Can the ephemeral key be used as a binding key?
- Redirect URI Validation
  - Can open redirection attacks be mitigated?
- Refresh Tokens
  - Should Refresh Tokens be returned
  - Should they be time bound similar to an Access Code?

What Next?

- Has this been proposed before?
- Has there been any formal analysis of this approach?
- Is anyone else doing this, or have an interest in doing this?
- Individual Draft?