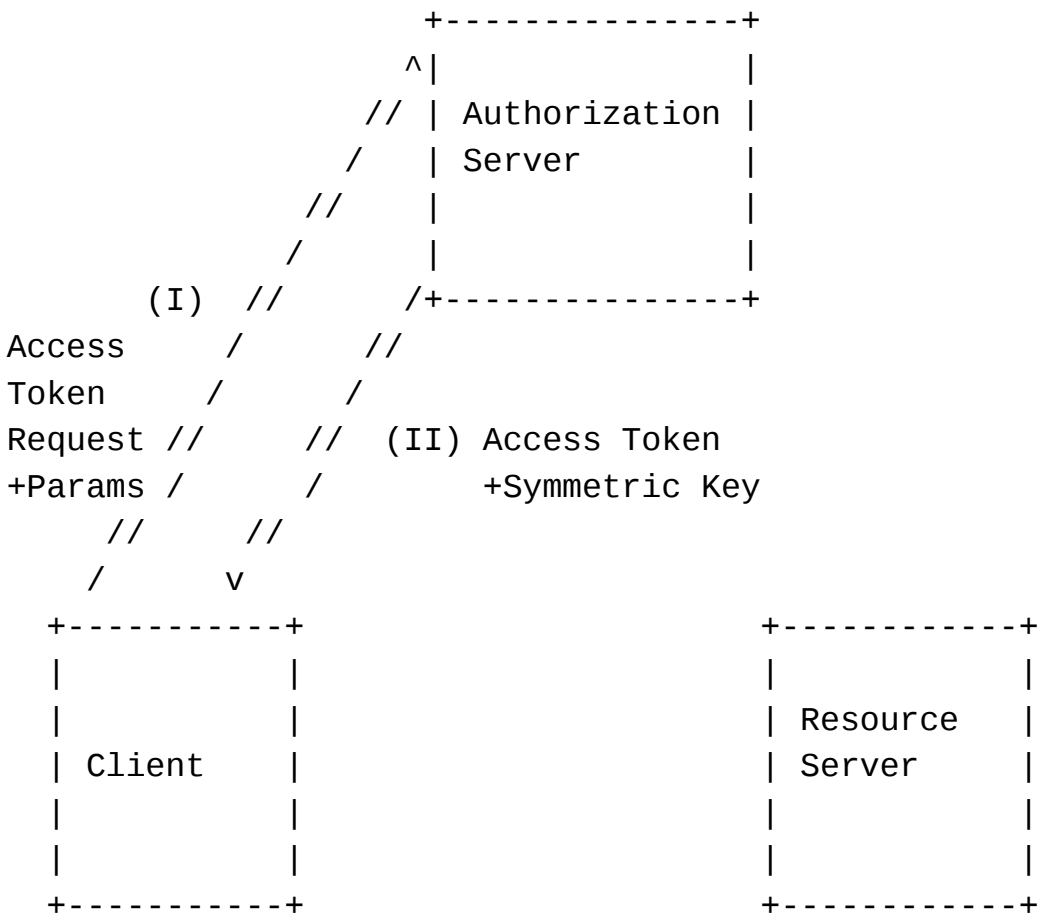
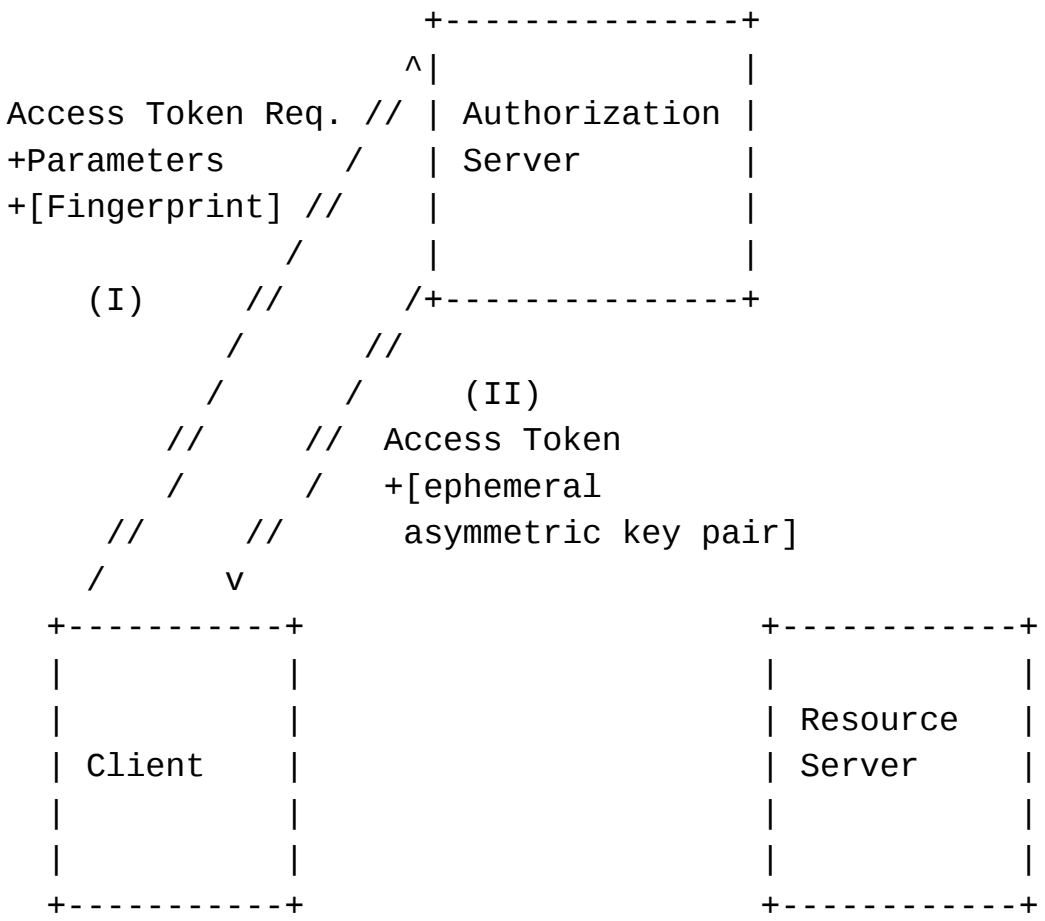
OAuth PoP Tokens

REFRESHER

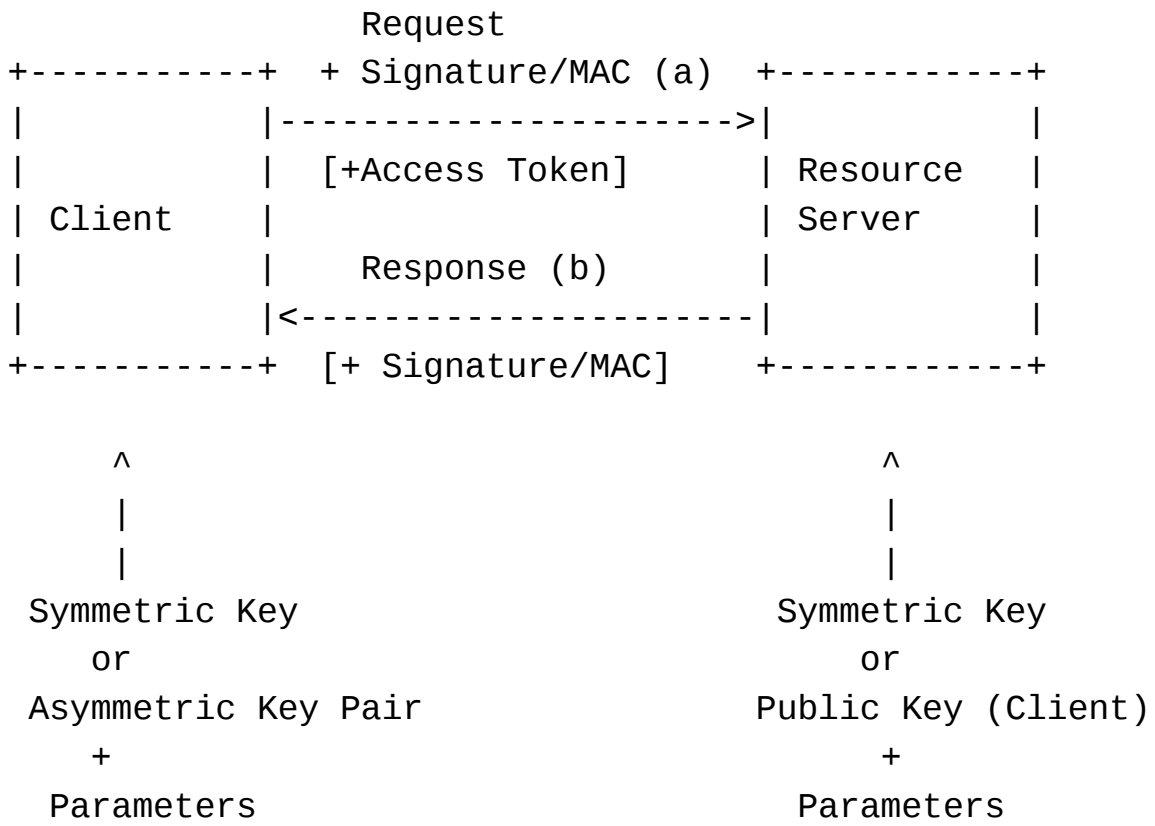
Interaction between the Client and the Authorization Server (Symmetric Keys)



Interaction between the Client and the Authorization Server (Asymmetric Keys)



Client Demonstrates PoP Token



STATUS

Status

- ACE-OAuth uses PoP token functionality in context with CWTs. It enhances the OAuth model with several new protocols, most notably CoAP.
- One use case allows regular OAuth to be used between the Client and the AS while the RS is constrained.
- Parameters for CoAP and HTTP are currently defined in draft-ietf-ace-oauth-authz.
- Additionally, the WebRTC community is also using OAuth in their STUN/TURN exchange.
- Details explained in https://www.ietf.org/mail-archive/web/ace/current/msg02907.html

Open Issues

- Where should the HTTP-based parameter definitions go?
- “alg” vs. “profile” parameter
- How should the key transport be encoded?

ACE-OAUTH PARAMETERS

ACE-OAuth defined Parameters

- **Audience**: This parameter specifies for which audience the client is requesting a token.
- **Confirmation**:
  - The "**cnf**" parameter identifies or provides the key used for proof-of- possession.
  - The "**rs_cnf**" parameter provides the raw public key of the RS.
- **Profile**: This parameter specifies the security protocol and transport the client must use with the RS.
- *These parameters are also defined for use with token introspection.*

PROFILES

What is an ACE Profile?

- The client or RS may be limited in the encodings or protocols it supports.
- To support a variety of different deployment settings, specific interactions between client and RS are defined in an ACE profile.
- In ACE framework the AS is expected to manage the matching of compatible profile choices between a client and an RS.
- The AS informs the client of the selected profile using the "profile" parameter in the token response.
- Example: coap_dtls

The “alg” Parameter

- To allow clients to indicate support for specific token types and respective algorithms they need to interact with authorization servers.
- The value in the 'alg' parameter together with value from the 'token_type' parameter allow the client to indicate the supported algorithms for a given token type.
- The token type refers to the specification used by the client to interact with the resource server to demonstrate possession of the key. The 'alg' parameter provides further information about the algorithm, such as whether a symmetric or an asymmetric crypto-system is used.

What is needed?

- To successfully establish access to access to a resource by a client it needs to pick an appropriate
  - protocol (e.g., CoAP, HTTP, MQTT)
  - token type (e.g., CWT, JWT, TLV)
  - Security protocol (e.g., DTLS vs. OSCORE)
  - credential type (e.g., PSK, RPK, certificate, …)
  - algorithm + parameter

Scope

- The ACE-OAuth spec assumes that these parameter settings can be “fixed” at the time of specification writing.
- The PoP key distribution draft assumed that the algorithms may vary.
- Answer depends also a bit on who has the relevant information and whether the protocol used between the client and the RS is able to perform parameter negotiation.

KEY TRANSPORT DESIGN

Key Transport

- The AS needs to send information about the key it included in the token to the Client.
- The idea has been to re-use the format of this key container.
- ACE-OAuth suggests to re-use the "cnf" parameter from <draft-ietf-ace-cwt-proof-of-possession-03>.
- This is great when CoAP and CBOR is used.
- For HTTP, however, the “cnf” coding from RFC 7800 would be more appropriate.
- For STUN/TURN/RTCWeb and their token format (as defined in RFC 7635) the story may again be different.
- Making the value carried in the cnf parameterdependent on the protocol and token type being used would be an option.