ACE Working Group   L. Seitz
SICS Swedish ICT

Internet-Draft   July 18, 2016
Intended Status: Standards Track
Expires: January 19, 2017

OSCOAP profile of ACE
draft-seitz-ace-oscoap-profile-00

Abstract

This memo specifies a profile for the ACE framework for Authentication and Authorization. It utilizes Object Security of CoAP (OSCOAP) and Ephemeral Diffie-Hellman over COSE (EDHOC) to provide communication security, server authentication, and proof-of-possession for a key owned by the client and bound to an OAuth 2.0 access token.

Status of this Memo

Copyright and License Notice

Table of Contents

1. Introduction

This memo specifies a profile of the ACE framework [I-D.ietf-ace-oauth-authz]. In this profile, a client and a resource server use CoAP to communicate. The client uses an access token, bound to a key (the proof-of-possession key) to authorize its access to the resource server. In order to provide communication security, proof of possession, and server authentication they use Object Security of CoAP (OSCOAP) [I-D.selander-ace-object-security] and Ephemeral Diffie-Hellman Over COSE (EDHOC) [I-D.selander-ace-cose-ecdhe]. Optionally the client and the resource server may also use CoAP and OSCOAP to communicate with the authorization server. The use of EDHOC in this profile in addition to OSCOAP, provides perfect forward secrecy (PFS) and the initial proof-of-possession, which ties the proof-of-possession key to an OSCOAP security context.

OSCOAP specifies how to use CBOR Object Signing and Encryption (COSE) [I-D.ietf-cose-msg] to secure CoAP messages. In order to provide replay and reordering protection OSCOAP also introduces sequence numbers that are used together with COSE. EDHOC specifies an authenticated Diffie-Hellman protocol that allows two parties to use COSE in order to establish a shared secret key with perfect forward secrecy.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. These words may also appear in this document in lowercase, absent their normative meanings.

Certain security-related terms such as "authentication", "authorization", "confidentiality", "(data) integrity", "message authentication code", and "verify" are taken from [RFC4949].

Since we describe exchanges as RESTful protocol interactions HTTP [RFC7231] offers useful terminology.

Terminology for entities in the architecture is defined in OAuth 2.0 [RFC6749] and [I-D.ietf-ace-actors], such as client (C), resource server (RS), and authorization server (AS).

Note that the term "endpoint" is used here following its OAuth definition, which is to denote resources such as /token and /introspect at the AS and /authz-info at the RS. The CoAP [RFC7252] definition, which is "An entity participating in the CoAP protocol" is not used in this memo.

2. Client to Resource Server

The use of OSCOAP for arbitrary CoAP messages is specified in [I-D.selander-ace-object-security]. This section defines the specific uses and their purpose for securing the communication between a client and a resource server, and the parameters for to negotiating the use of this profile with the token endpoint at the authorization server as specified in section 6 of the ACE framework [I-D.ietf-ace-oauth-authz].

2.1. Signalling the use of OSCOAP

A client requesting a token at an AS via the /token endpoint MAY signal a preference for using OSCOAP by including the "profile" parameter with the value "coap_oscoap" in it's access token request. This follows the message formats specified in section 6.1 of the ACE framework.

The AS responding to a successful access token request as defined in section 6.2 of the ACE framework can signal that the use of OSCOAP is REQUIRED for a specific access token by including the "profile" parameter with the value "coap_oscoap" in the access token response. This means that the client MUST use OSCOAP towards all resource servers for which this access token is valid.

The error response procedures defined in section 6.3 of the ACE framework are unchanged by this profile.

Note the the client and the authorization server MAY OPTIONALLY use OSCOAP to protect the interaction via the /token endpoint. See section 3 for details.

2.2. Key establishment for OSCOAP

Section 3.2 of OSCOAP [I-D.selander-ace-object-security] defines how to derive a security context based on a pre-shared secret established between client and server. If the proof-of-possession key is a symmetric key, it MAY be directly used as shared secret with OSCOAP.

However to provide forward secrecy and mutual authentication in the case of pre-established raw public keys or with X.509 certificates it is RECOMMENDED to use EDHOC [I-D.selander-ace-cose-ecdhe] to generate the initial shared key. EDHOC MUST be used as follows:

When the client sends the access token to the RS using the /authz-info endpoint as specified in section 8.1 of the ACE framework, this message MUST carry message_1 of the EDHOC protocol in the CoAP payload, and the access token MUST be included in the COSE

unprotected header of message_1 as a CBOR map with the key 'access_token'.

When the RS responds to this token submission request, if the access token was valid the payload of the CoAP response MUST contain message_2 of the EDHOC protocol. If the token was not valid, the error response defined in the ACE framework is not modified. If the EDHOC message_1 was not valid the RS MUST respond with error code 4.01 (Unauthorized).

In the case of EDHOC being used with symmetric pop-keys, the protocol in section 3.4 of [I-D.selander-ace-cose-ecdhe] MUST be used. If the pop-key is asymmetric, the RS MUST also use an asymmetric key for authentication. This key is known to the client through the access token response (see section 6.2 of the ACE framework). In this case the protocol in section 3.5 of [I-D.selander-ace-cose-ecdhe] MUST be used.

Note that if the OSCOAP profile is used, the /authz-info endpoint at the Resource Server MUST be prepared to process and generate the protocol messages of the EDHOC protocol as specified above. Hence the use of EDHOC does not add any additional roundtrips to the ACE message exchange.

Figure 1 illustrates the message exchanges for using EDHOC on the /authz-info endpoint (step C in figure 1 of [I-D.ietf-ace-oauth-authz]).

```
                  Resource
        Client    Server
          |         |
          |         |
      C:  +-------->| Header: POST (Code=0.02)
          | POST    | Uri-Path:"authz-info"
          |         | Content-Type: application/cose+cbor
          |         | Payload: EDHOC message_1 + access token
          |         |
          |<--------+ Header: 2.04 Changed
          |         | Content-Type: application/cose+cbor
          | 2.05    | Payload: EDHOC message_2
          |         |
```

Figure 1: Key establishment with EDHOC via the authz-info endpoint

Figure 2 shows an example of message_1 with an access token embedded in the unprotected header.

```
     997(
       [
```

```
             / protected / h'a201260444c150d41c',
                 / 'alg' : 'ES256', 'kid' : 'kid_c' /
             / unprotected / {'access_token' : h'4a5015df6864286979'},
             / payload / h'83381a0c582fa120a50102024103200121582098f
             50a4ff6c05861c8860d13a638ea56c3f5ad7590bbfbf054e1c7b4d9
             1d628022f5',
             / signature / h'eae868ecc1276883766c5dc5ba5b8dca25dab3c
             2e56a51ce5705b793914348e14eea4aee6e0c9f09db4ef3ddeca8f3
             506cd1a98a8fb64327be470355c9657ce0'
          ]
        )
```

Figure 2: EDHOC message_1 with an access token

2.3. Securing the Resource Request

When the client wishes to send a request to the RS, it uses the steps defined in section 6 of OSCOAP [I-D.selander-ace-object-security] to generate an OSCOAP message out of the unsecured CoAP message.

2.4. Securing the Resource Server Response

When a RS responds to a client's request, it uses the steps defined in section 6 of OSCOAP [I-D.selander-ace-object-security] to generate an OSCOAP message out of the unsecured CoAP message.

3. Client to Authorization Server

As specified in the ACE framework section 5 [I-D.ietf-ace-oauth-authz], the Client and AS can also use CoAP instead of HTTP to communicate via the token endpoint. This section specifies how to use OSCOAP between Client and AS together with CoAP. The use of OSCOAP for this communication is OPTIONAL in this profile, other security protocols (such as DTLS) MAY be used instead.

The client and the AS are expected to have pre-established credentials (e.g. raw public keys). How these credentials are established is out of scope for this profile. Furthermore the client and the AS communicate using CoAP through the token endpoint as specified in section 6 of [I-D.ietf-ace-oauth-authz]. At first point of contact, prior to making the token request and response, the client and the AS MUST perform an EDHOC exchange with the pre-established credentials to create forward secret keying material for use with OSCOAP. Subsequent requests and the responses MUST be protected with OSCOAP.

4. Resource Server to Authorization Server

As specified in the ACE framework section 5 [I-D.ietf-ace-oauth-authz], the RS and AS can also use CoAP instead of HTTP to communicate via the introspection endpoint. This section specifies how to use OSCOAP between RS and AS together with CoAP. The use of OSCOAP for this communication is OPTIONAL in this profile, other security protocols (such as DTLS) MAY be used instead.

The RS and the AS are expected to have pre-established credentials (e.g. symmetric keys). How these credentials are established is out of scope for this profile. Furthermore the RS and the AS communicate using CoAP through the introspection endpoint as specified in section 7 of [I-D.ietf-ace-oauth-authz]. At first point of contact, prior to making the introspection request and response, the RS and the AS MUST perform an EDHOC exchange with the pre-established credentials to create forward secret keying material for use with OSCOAP. Subsequent requests and the responses MUST be protected with OSCOAP.

5. Security Considerations

TBD.

6. Privacy Considerations

TBD.

7. IANA Considerations

FIXME: PoP alg: OSCOAP

8. Acknowledgements

The author wishes to thank Goeran Selander and Francesca Palombini for the input on this memo.

9. References

9.1 Normative References

[I-D.selander-ace-object-security] Selander, G., Mattsson J., Palombini F., and L. Seitz. "Object Security of CoAP (OSCOAP)", draft-selander-ace-object-security-04 (work in progress), March 2016.

[I-D.selander-ace-cose-ecdhe] Selander, G., Mattsson J., and F. Palombini. "Ephemeral Diffie-Hellman Over COSE (EDHOC)", draft-selander-ace-cose-ecdhe-02 (work in progress), June 2016.

[I-D.ietf-ace-oauth-authz] Seitz, L., Selander, G., Wahlstroem, E., Erdtmann, S., and H. Tschofenig. "Authentication and Authorization for Constrained Environments (ACE)", drart-ietf-ace-oauth-authz-02 (work in progress), June 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://www.rfc-editor.org/info/rfc7252>.

9.2 Informative References

[I-D.gerdes-ace-actors]
Gerdes, S., Seitz, L., G. Selander, and C. Bormann (ed). "An Arhitecture for Authorization in Constrained Environments", draft-ietf-ace-actors-03 (work in progress), March 2016.

[I-D.ietf-cose-msg] Schaad, J., "CBOR Object Signing and Encryption (COSE)", draft-ietf-cose-msg-14 (work in progress), June 2016.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <http://www.rfc-editor.org/info/rfc4949>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <http://www.rfc-editor.org/info/rfc6749>.

[RFC7231] Fielding, R., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <http://www.rfc-editor.org/info/rfc7231>.

Author's Address

Ludwig Seitz
SICS Swedish ICT AB
Scheelevagen 17
22370 Lund
SWEDEN
EMail: ludwig@sics.se