ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 2, 2018

S. Aragon
M. Tiloca
S. Raza
RISE SICS AB

October 29, 2017

IPsec profile of ACE
draft-aragon-ace-ipsec-profile-01

Abstract

This document defines a profile of the ACE framework for authentication and authorization. It uses the IPsec protocol suite and the IKEv2 protocol to ensure secure communication, server authentication and proof-of-possession for a key bound to an OAuth 2.0 access token.

Table of Contents

1. Introduction

The IPsec protocol suite [RFC4301] allows communications based on the Constrained Application Protocol (CoAP) [RFC7252] to fulfill a number of security goals at the network layer, i.e. integrity and IP spoofing protection, confidentiality of traffic flows, and message replay protection. In several resource-constrained platforms, this can leverage security operations directly provided by hardware

crypto-modules, including mandatory-to-implement cipher suites defined in [RFC4835].

This document defines a profile of the ACE framework for authentication and authorization [I-D.ietf-ace-oauth-authz], where a client (C) and a resource server (RS) communicate using CoAP [RFC7252] over IPsec [RFC4301]. In particular, C uses an Access Token released by an Authorization Server (AS) and bound to a key (proof-of-possession key) to authorize its access to RS and its protected resources.

The establishment of an IPsec channel between C and RS provides secure communication, proof-of-possession as well as RS and C mutual authentication. Furthermore, this profile preserves the flexibility of IPsec as to the selection of specific security protocols, i.e. Encapsulating Security Payload (ESP) [RFC4303] and IP Authentication Header (AH) [RFC4302], key management, and modes of operations, i.e. tunnel or transport. Those parameters are specified in the IPsec Security Association (SA) pair established between C and RS. Optionally, the client and the resource server may also use CoAP and IPsec to communicate with the Authorization Server.

This specification supports different key management methods for setting up SA pairs, namely direct provisioning of SA pairs and establishment of SA pairs based on symmetric or asymmetric key authentication. The latter approach relies on the Internet Key Exchange Protocol version 2 (IKEv2) [RFC7296].

1.1. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here. These keywords indicate requirement levels for compliant CoAP-IPsec profile implementations.

Readers are expected to be familiar with terminology such as client (C), resource server (RS), authentication server (AS), and endpoint which are defined in [RFC6749] and [I-D.ietf-ace-actors]. It is assumed in this document that a given resource on a specific RS is associated to a unique AS.

The concept of IPsec Security Association (Section 4.1. of [RFC4301]) plays a key role, and this profile uses it extensively. An SA indicates how to secure a one-way communication between two parties. Hence, two SAs are required to be created and coordinated, in order

to secure a two-way communication channel. This document refers to a SA pair as the two IPsec SAs used to protect the two-way communication channel between two IPsec peers.

The SA parameters described in section 4.4.2.1 of [RFC4301] are divided into the following two sets.

- Network Parameters: the parameters defining the network properties of the IPsec channel, e.g. DSCP filtering;

- Security Parameters: the parameters defining the security properties of the IPsec channel.

This document refers to SA-C as the SA for securing communication from C to RS, and to SA-RS as the SA for securing communication from RS to C. Thus, a SA pair consists of an SA-C and an SA-RS.

2. Methods for Setting Up SA Pairs

The following key management methods are supported for setting up a SA pair between C and RS.

1. Direct Provisioning (DP). The SA pair is pre-defined by the AS. Then, SA-RS and SA-C are specified in the Access Token Response and in the Access Token issued by the AS.

2. Establishment with symmetric key authentication. A symmetric Pre-Shared Key (PSK) is used to authenticate both parties during the SA pair establishment and is bound to the Access Token as proof-of-possession key. If C is interacting for the first time with the RS, then the AS MUST include a PSK and a unique key identifier in the Access Token Response. Otherwise, C MUST include the unique key identifier pointing at the previously established PSK in the Access Token Request.

3. Establishment with asymmetric key authentication. An asymmetric Raw Public Key (RPK) or Certificate-based Public Key (CPK) is used to authenticate both parties during the SA pair establishment and is bound to the Access Token as proof-of-possession key. If the AS does not know C's asymmetric authentication information, then C MUST include its RPK or CPK in the Access Token Request. Otherwise, C MUST include a key identifier linked to its own RPK or CPK available at the AS.

Every SA MUST include the following Security Parameters.

- A Security Parameter Index (SPI);

- IPsec protocol mode: tunnel or transport;
- Security protocol: AH or ESP;
- "AH-authentication", "ESP-encryption", "ESP-integrity" or "ESP-combined" algorithm;
- Source and destination, if tunnel mode is selected;
- Cryptographic keys;
- SA lifetime.

As assumed in Section 5.5.2 of [I-D.ietf-ace-oauth-authz], the AS has knowledge of C's and RS's capabilities, and of RS's preferred communications settings. Therefore, the AS MUST set the values of Security Parameters and Network Parameters in the SA pair.

2.1. The "ipsec" Structure

This document defines the "ipsec" structure as a field of the "cnf" parameter of the Access Token and Access Token Response. This structure encodes the Network and Security Parameters of the SA pair as defined in Figure 1. The Network Parameters are not discussed in this specification.

```
ipsec{
        <Security Parameters>,
        <Network Parameters>
}
```

Figure 1: "ipsec" structure overview.

The AS builds the "ipsec" structure as follows:

- The Security Parameters MUST always include the set of parameters sec_A shown in Figure 2.
- The Security Parameters MUST include the set of parameters sec_B shown in Figure 3 if the AS uses the Direct Provisioning method.

```
sec_A{
        mode,
        protocol,
        life,
        IP_C,  (if mode == tunnel)
        IP_RS  (if mode == tunnel)
}
```

Figure 2: Set sec_A of Security Parameters

```
sec_B{
        SPI_SA_C,
        SPI_SA_RS,
        alg,
        seed
}
```

Figure 3: Set sec_B of Security Parameters

In sec_A, the IP_C field is the IP address of C, while IP_RS is the IP address of RS. In tunnel mode, the RS MUST use IP_C as the destination address and IP_RS as source address of outgoing IPsec messages. Similarly, C MUST use IP_RS as destination address and IP_C as source address of incoming IPsec messages.

In sec_B, the field "SPI_SA_C" is the SPI of SA-C. Similarly, "SPI_SA_RS" is the SPI of SA-RS. The field "alg" indicates the algorithm used for securing communications over IPsec. The "seed" field MUST reflect the SKEYSEED secret defined in Section 2.14 of [RFC7296]. Thus, C and RS MUST use the same key derivation techniques to generate the necessary SA keys from "seed".

Note that if the Direct Provisioning method is used, the AS cannot guarantee the uniqueness of the "SPI_SA_C" value at the RS and of the "SPI_SA_RS" value at C. In such a case, the AS MUST randomly generate the "SPI_SA_C" value and the "SPI_SA_RS" value, so that the probability of a collision to occur is negligible.

If RS receives an "SPI_SA_C" value which results in a collision, then RS MUST reply to C with en error response, and both C and RS MUST abort the set up of the IPsec channel. In order to overcome this issue, the AS can manage a pool of "SPI_SA_C" reserved values, intended only for use with the Direct Provisioning method. Then, in case of SA termination, the RS asks the AS to set back the identifier of that SA-C as available.

If C receives an "SPI_SA_RS" value which results in a collision, then C sends a second Token Request to the AS, asking for a Token Update.

This Token Request includes also an "ipsec" structure, which contains only the field "SPI_SA_RS" specifying an available value to use. Then, the AS replies with an Access Token and an Access Token Response both updated as to the "SPI_SA_RS" value only.

3. Protocol Description

This profile considers a client C that intends to access a protected resource hosted by a resource server RS. The resource access is authorized through an Access Token issued by the AS as specified in [I-D.ietf-ace-oauth-authz] and indicating that IPsec is used to secure communications between C and RS. In particular, this profile defines how C and RS set up a SA pair, using the key management methods introduced in Section 2.

The protocol is composed of three parts, as shown in Figure 4.

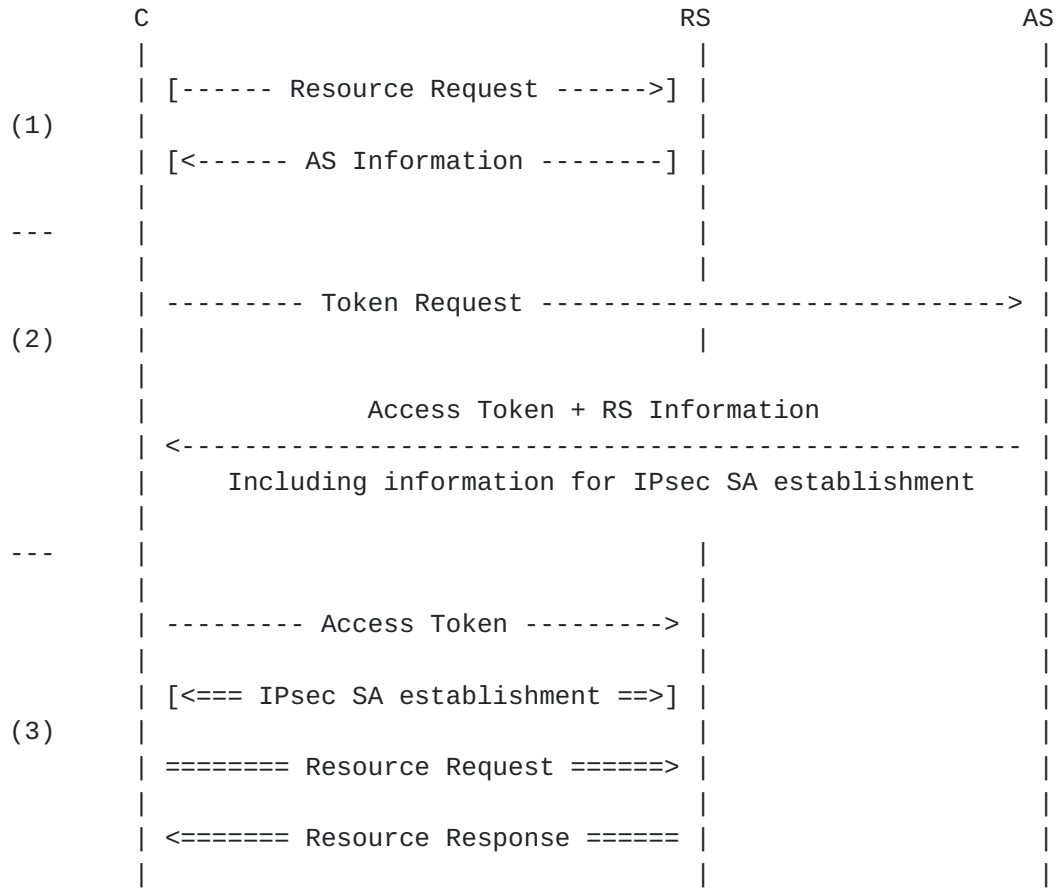

Figure 4: Protocol Overview

3.1. Unauthorized Client to RS

Phase (1) in Figure 4 is OPTIONAL and aims at providing C with the necessary information to contact the AS, in case C does not know AS's address. Through an unauthorized request to RS, C determines which AS is responsible for granting authorization to that particular RS. When doing so, C learns to which address the Access Token Request has to be addressed. The unauthorized request is denied by RS, which sends back to C a response containing the information to contact the AS.

3.2. Client to AS

Phase (2) in Figure 4 starts with C sending the Access Token Request to the /token endpoint at the AS, as specified in Section 5.5.1 of [I-D.ietf-ace-oauth-authz]. Figure 2 and Figure 3 of [I-D.ietf-ace-oauth-authz] provide examples of such request.

If the AS successfully verifies the Access Token Request and C is authorized to access the resource specified in the Token Request, then the AS issues the corresponding Access Token and includes it in a CoAP response with code 2.01 (Created) as specified in Section 5.5.2 of [I-D.ietf-ace-oauth-authz]. The AS can signal that IPsec is REQUIRED to secure communications between C and RS by including the "profile" parameter with the value "coap_ipsec" in the Access Token Response. Together with authorization information, the Access Token also includes the same information for the set up of the IPsec channel included in the Access Token Response. The error response procedures defined in Section 5.5.3 of [I-D.ietf-ace-oauth-authz] are unchanged by this profile.

The information exchanged between C and the AS depends on the specific method used to set up the SA pair (see Section 3.2.1, Section 3.2.2 and Section 3.2.3). Note that, unless Direct Provisioning of SAs is used, C and RS are required to finalize the SA pair set up by running a Key Management Protocol such as IKEv2 (see Section 3.3.2). The AS indicates to use IKEv2 for establishing a SA pair by setting the "kmp" field to "ikev2" in the "cnf" parameter in the Access Token Response.

As specified in Section 5.5 of [I-D.ietf-ace-oauth-authz], the Client and the AS can also use CoAP instead of HTTP to communicate via the /token endpoint. This communication channel MUST be secured.

This section specifies how to use IPsec [RFC4301] to protect the channel between the Client and the AS. The use of IPsec for this communication channel is OPTIONAL in this profile, and other security

protocols MAY be used instead, such as DTLS [RFC6347] and OSCORE [I-D.ietf-core-object-security].

The Client and the AS are either expected to have pre-established a pair of IPsec SA or to have pre-established credentials to authenticate an IKEv2 key exchange. How these credentials are established is out of scope for this profile.

3.2.1. Direct Provisioning of SA pairs

If the AS selects this key management method, it encodes the SA pair in the Access Token and in the Access Token Response as an "ipsec" structure in the "cnf" parameter.

Figure 5 shows an example of an Access Token Response, signaling C to set up an IPsec channel with RS based on the ESP protocol in transport mode.

```
Header: Created (Code=2.01)
Content-Type: "application/cose+cbor"
Payload : {
    "access_token" : b64'YiksuH&=1GFfg ...
    (remainder of Access Token omitted for brevity)',
    "profile" : "coap_ipsec",
    "expires_in" : "3600",
    "cnf" : {
       "ipsec" : {

             "mode"      : "transport",
             "protocol"  : "ESP",
             "life"      : "3600",
             "SPI_SA_C"  : "87615",
             "SPI_SA_RS" : "87616",
             "seed"      : b64'+a+Dg2jjU+eIiOFCa9lObw',
             "alg"       : "AES-CCM-16-64-128",
             ... (the Network Parameters are omitted for brevity),
       }
    }
}
```

Figure 5: Example of Access Token Response with DP of SA pair

3.2.2. SA Establishment Based on Symmetric Keys

If the AS selects this key management method, it specifies the following pieces of information in the Access Token Response and in the Access Token:

- a symmetric key to be used as proof-of-possession key;
- a key identifier associated to the symmetric key;
- SA pair's Network Parameters and Security Parameters, as an "ipsec" structure in the "cnf" parameter (see Section 2.1).

If C has previously received a PSK from the AS, then C MUST provide a key identifier of that PSK either directly in the "kid" field of "cnf" parameter or in the "kid" field of the "COSE_Key" object of the Access Token Request. In this case, the AS omits the PSK and its identifier in the Access Token Response.

The AS indicates the use of symmetric cryptography for the key management message exchange in the "kty" field of the "COSE_Key" object, including also the PSK in the "k" field as well as its key identifier in the "kid" field, as shown in Figure 6.

```
Header: Created (Code=2.01)
Content-Type: "application/cose+cbor"
Payload:
  {
    "access_token" : b64'YiksuH&=1GFfg ...
    (remainder of Access Token omitted for brevity)',
    "profile" : "coap_ipsec",
    "expires_in" : "3600",
    "cnf" : {
      "COSE_Key" : {
      "kty" : "Symmetric",
      "kid" : b64'6kwi42ec',
      "k"   : b64'+pAd48jU+eIiOF23gd=',
      }
      "kmp": "ikev2",
      "ipsec" : {
           "mode"     : "tunnel",
           "protocol" : "ESP",
           "life"     : "1800",
           "IP_C"     : "a.b.c.d2",
           "IP_RS"    : "a.b.c.d1",
           ... (the Network Parameters are omitted for brevity),
      }
    }
  }
```

Figure 6: Example of Access Token Response with a symmetric key as proof-of-possession key.

3.2.3. SA Establishment Based on Asymmetric Keys

C MUST include its own public key in the Access Token Request, as shown in Figure 7. As an alternative, C MUST provide the key identifier of its own public key, previously shared with the AS.

The AS specifies in the Access Token and in the Access Token Response the SA pair's Network Parameters and Security Parameters, as an "ipsec" structure in the "cnf" parameter (see Section 2.1).

In addition, the AS specifies the RS's public key in the Access Token Response, and the C's public key to be used as proof-of-possession key in the Access Token.

The AS indicates the use of asymmetric cryptography for the key management message exchange in the "kty" field of the "COSE_Key" object, which includes also the RS's public key in the Access Token Response and the C's public key in the Access Token.

```
Header: POST (Code=0.02)
Uri-Host: "server.example.com"
Uri-Path: "token"
Content-Type: "application/cose+cbor"
Payload:
{
   "grant_type" : "client_credentials",
   "cnf" : {
     "COSE_Key" : {
            "kty" : "EC",
            "crv" : "P-256",
            "x"   : b64'CaFadPPavdtjRH3YqaTqm0FrFtNV0',
            "y"   : b64'ehekJBwciJdeT6cKieycnk6kg4pHC'
     }
   }
 }
```

Figure 7: Example of Access Token Request with an asymmetric key as proof-of-possession key.

3.3. Client to RS

Phase (3) in Figure 4 starts with C posting the Access Token by means of a POST CoAP message to the /authz-info endpoint at RS, as specified in Section 5.7 of [I-D.ietf-ace-oauth-authz]. The processing details of this request, as well as the handling of invalid Access Tokens at RS, are defined in Section 5.7.1 of [I-D.ietf-ace-oauth-authz] and in the rest of this section. The Access Token and Access Token Response specify one of the SA setup

methods defined in Section 2. In particular, C and RS determine the specific SA setup method as follows:

- In case of Direct Provisioning, the "ipsec" structure is present, while the "COSE_Key" object is not present.
- If the SA pair set up based on Symmetric Keys through IKEv2 is used, then:
  - the "COSE_Key" object is present with the "kty" field set to "Symmetric"; and
  - the "kmp" parameter is set to "ikev2".
- If the SA pair set up based on Asymmetric Keys through IKEv2 is used, then:
  - the "COSE_Key" object is present with the "kty" field set to a value that indicates the use of an asymmetric key, e.g. "EC"; and
  - the "kmp" parameter is set to "ikev2".

If the Direct Provisioning method is used, then C and RS do not perform the SA establishment shown in Figure 4. Otherwise, C and RS perform the key management protocol indicated by the "kmp" parameter (such as IKEv2), in the authentication mode indicated by the "kty" field of the "COSE_key" object.

Regardless the chosen SA setup method and the successful establishment of the IPsec channel, if C holds a valid Access Token but this does not grant access to the requested protected resource, RS MUST send a 4.03 (Forbidden) response. Similarly, if the Access Token does not cover the intended action, RS MUST send a 4.05 (Method Not Allowed) response.

3.3.1. SA Direct Provisioning

Once received a positive Access Token Response from the AS, C derives the necessary IPsec key material from the "seed" field of the "ipsec" structure in the Access Token Response, as discussed in Section 2.1. Similarly, RS performs the same key derivation process upon receiving and successfully verifying the Access Token. After that, RS replies to C with a 2.01 (Created) response, using the IPsec channel specified by the SA pair. Thereafter, Resource Requests and Responses are also sent using the IPsec channel.

3.3.2. Authenticated SA Establishment

If an Authenticated Key Management method is used (see Section 3.2.2 and Section 3.2.3), C and RS MUST run a Key Management Protocol to finalize the establishment of the SA pair and the IPsec channel, i.e. the required keys and algorithms. As shown in Figure 8, the first message IKE_SA_INIT of the IKEv2 protocol is used to acknowledge the Access Token submission. Depending on the used authentication method, i.e. symmetric or asymmetric, the proof-of-possession key MUST be used accordingly to authenticate the IKEv2 message exchange as defined in Section 2.15 of [RFC7296]. The rest of the IKEv2 protocol MUST be executed between C and RS as described in Section 2 of [RFC7296], with no further modifications. If IKEv2 is successfully completed, C and RS agree on keys and algorithms to use, and thus the IPsec channel between C and RS is ready to be used.

Figure 8: IKEv2 used as Key Management Protocol.

3.4. RS to AS

As specified in Section 5.6 of [I-D.ietf-ace-oauth-authz], the RS and the AS can also use CoAP instead of HTTP to communicate via the /introspect endpoint. This communication channel MUST be secured.

This section specifies how to use IPsec to protect the channel between the RS and the AS. The use of IPsec for this communication channel is OPTIONAL in this profile, and other security protocols MAY be used instead, such as DTLS [RFC6347] and OSCORE [I-D.ietf-core-object-security].

The RS and the AS are either expected to have pre-established a pair of IPsec SA or to have pre-established credentials to authenticate an IKEv2 key exchange. How these credentials are established is out of scope for this profile.

4. Security Considerations

This document inherits the security considerations of [RFC4301], [RFC4302] and [RFC4303]. Furthermore, if IKEv2 is used as key establishment method (see Section 3.3.2), the same considerations discussed in [RFC7296] hold.

4.1. Privacy Considerations

The message exchange in Phase (1) of Figure 4 is unprotected and MAY disclose the relation between the AS, RS and C, as well as network related information, such as IP addresses. Thus RS SHOULD only include the necessary information to contact the AS in the unprotected response.

5. IANA Considerations

This document requires the following IANA considerations:

name	label	CBOR type	value	description
kmp	TBD	bstr	ikev2	Indicates the key management protocol to be used to establish a SA pair
ipsec	TBD	struct		Contains Security and Network Parameters of an SA pair

5.1. CoAP-IPsec Profile registration

- Profile name: CoAP-IPsec
- Profile description: ACE Framework profile
- Profile ID: coap_ipsec
- Change Controller: IESG
- Specification Document: This document

5.2. Confirmation Methods registration

5.2.1. IPsec field

- Confirmation Method Name: "ipsec"
- Confirmation Method Value: TBD
- Confirmation Method Description: A structure containing the corresponding information of an IPsec Security Association Pair.
- Change Controller: IESG
- Specification Document: This document

5.2.2. Key Management Protocol field

- Confirmation Method Name: "kmp"
- Confirmation Method Value: TBD
- Confirmation Method Description: Key management protocol.
- Change Controller: IESG
- Specification Document: This document

5.3. Key Management Protocol Methods Registry

This specification establishes the IANA "Key Management Protocol Methods" registry for the "kmp" member values. The registry records the confirmation method member and a reference to the spec that defines it.

5.3.1. Registration Template

Key Management Protocol Method Name:

> The name requested (e.g. "ikev2"). This name is intended to be human readable and be used for debugging purposes. It is case sensitive. Names may not match other registered names in a case-insensitive manner unless the Designated Experts state that there is a compelling reason to allow an exception.

Key Management Protocol Method Value:

Integer representation for the confirmation method value.
Intended for use to uniquely identify the confirmation method.
The value MUST be an integer in the range of 1 to 65536.

Key Management Protocol Method Description:

Brief description of the confirmation method (e.g. "Key Identifier").

Change Controller:

For Standards Track RFCs, list the "IESG". For others, give the name of the responsible party. Other details (e.g. postal address, email address, home page URI) may also be included.

Specification Document(s):

Reference to the document or documents that specify the parameter, preferably including URIs that can be used to retrieve copies of the documents. An indication of the relevant sections may also be included but is not required.

5.3.2. Initial Registry Contents

- Key Management Protocol Method Name: "ikev2"
- Key Management Protocol Method Value: TBD
- Key Management Protocol Method Description: Defines IKEv2 as key management protocol.
- Change Controller: IESG
- Specification Document: this document

6. Acknowledgments

The authors sincerely thank Max Maass for his comments and feedback.

The authors gratefully acknowledge the EIT-Digital Master School for partially funding this work.

7. References

7.1. Normative References

[I-D.ietf-ace-oauth-authz]
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE)", draft-ietf-ace-oauth-authz-08 (work in progress), October 2017.

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

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC4835] Manral, V., "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4835, DOI 10.17487/RFC4835, April 2007, <https://www.rfc-editor.org/info/rfc4835>.

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

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

7.2. Informative References

[I-D.ietf-ace-actors]
Gerdes, S., Seitz, L., Selander, G., and C. Bormann, "An architecture for authorization in constrained environments", draft-ietf-ace-actors-05 (work in progress), March 2017.

[I-D.ietf-core-object-security]
Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", draft-ietf-core-object-security-06 (work in progress), October 2017.

[I-D.seitz-ace-oscoap-profile]
Seitz, L., Palombini, F., and M. Gunnarsson, "OSCORE profile of the Authentication and Authorization for Constrained Environments Framework", draft-seitz-ace-oscoap-profile-06 (work in progress), October 2017.

[I-D.selander-ace-cose-ecdhe]
Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", draft-selander-ace-cose-ecdhe-07 (work in progress), July 2017.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

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

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

Appendix A. Coexistence of OSCORE and IPsec

Object Security of Constrained RESTful Environments (OSCORE) [I-D.ietf-core-object-security] is a data object based security protocol that protects CoAP messages end-to-end while allowing proxy operations. It encloses unprotected CoAP messages, and selected CoAP options and headers fields into a CBOR Object Signing and Encryption (COSE) object [RFC8152]. This section describes a scenario where communications between C and RS are secured by means of OSCORE and IPsec. Figure 9 depicts a scenario where a Client needs to access a

Resource Server which is behind an untrusted CoAP-Proxy. This scenario requires that:

1. the Proxy has access to the selected CoAP options to perform management and support operations;

2. the integrity of messages and their IP headers can be verified by the Resource Server;

3. the confidentiality of the Resource Server address and CoAP request has to be guaranteed between the Client and the Proxy.

The first requirement is addressed by means of an OSCORE channel between the Client and the Resource Server established as described in [I-D.seitz-ace-oscoap-profile]), by marking as Class E the sensitive fields of the CoAP payload as defined in [I-D.ietf-core-object-security].

To address the second requirement, a SA pair between the Client and the Resource Server is established, as specified in Section 3, by using the IPsec AH protocol in transport mode. Finally, the third requirement is fulfilled by means of a SA pair between the Client and the CoAP-Proxy, as specified in Section 3, by using the IPsec ESP protocol in tunnel mode.

This profile can be used to establish the necessary SA pairs. After that, C can request a token update to the AS, in order to establish an OSCORE security context with RS, as specified in Section 2.2 of [I-D.seitz-ace-oscoap-profile].

Figure 9 overviews the involved secure communication channels. Logical links such as the SA pair shared between the Client and the Proxy are represented by dotted lines. IPsec traffic is depicted with double-dashed lines, and an example of the packets going through these links is represented with numbers, e.g. (1). The destination address included in the IP headers is also specified, e.g. "IP:P" indicates the Proxy's address as destination address. The source address of the IP header is omitted, since all the IP packets have the Client's address as source address.

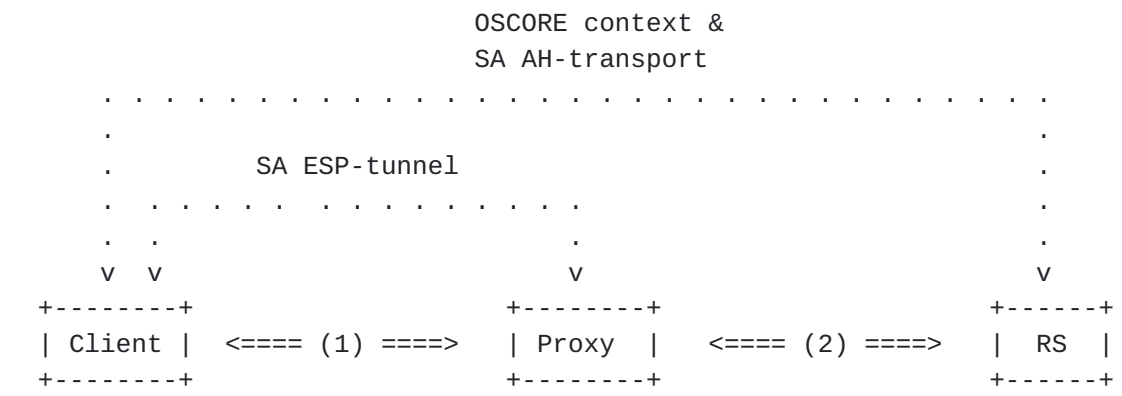

Figure 9: OSCORE and IPsec - Scenario overview

Appendix B. SA Establishment with EDHOC

As discussed in Appendix A, securing communications between C and RS with both OSCORE and IPsec makes it possible to fulfill a number of additional security requirements. An OSCORE security context between C and RS can be established using Ephemeral Diffie-Hellman Over COSE (EDHOC) as defined in Appendix C.2 of [I-D.selander-ace-cose-ecdhe] and according to [I-D.seitz-ace-oscoap-profile]. This section proposes a method to establish also IPsec SA pairs by means of EDHOC. This makes it possible for constrained devices running the scenario described in Appendix A to rely solely on EDHOC for establishing both OSCORE contexts and IPsec SA pairs, thus avoiding to include the implementation of IKEv2 as further key management protocol.

In particular, C and RS can refer to the SA Authenticated Establishment methods described in this specification, and then use EDHOC to finalize the SA pair, i.e. by deriving the encryption and authentication keys for the security protocols specified in the SA pair. This is possible thanks to IPsec's independence from specific key management protocols. In addition, the same security consideration discussed in [I-D.selander-ace-cose-ecdhe] hold.

The AS, C and RS refer to the same protocol shown in Figure 4, with the following changes.

B.1. Client to AS

The AS specifies the fields "alg", "SPI_SA_C" and "SPI_SA_RS" of the "ipsec" structure in the Access Token and in the Access Token Response, in addition to the pieces of information defined in

Section 3.2.2 or Section 3.2.3, in case the proof-of-possession key is symmetric or asymmetric, respectively.

The AS signals that EDHOC MUST be used, by setting the "kmp" field to "edhoc" in the Access Token and the Access Token Response. Then, C and RS MUST perform EDHOC as described in Section 4 or Section 5 of [I-D.selander-ace-cose-ecdhe], in case the proof-of-possession key is asymmetric or symmetric, respectively.

B.2. Client to RS

Figure 10 shows how EDHOC message_1 is sent through a POST Access Token Request to the /authz-info at the RS. The RS SHALL process the Access Token according to [I-D.ietf-ace-oauth-authz], and, if valid, continue with the EDHOC protocol as defined in Appendix C.1 of [I-D.selander-ace-cose-ecdhe]. Otherwise, RS aborts EDHOC and responds with an error code as specified in [I-D.ietf-ace-oauth-authz]. At the end of the EDHOC protocol, C and RS MUST derive an IPsec seed from the EDHOC shared secret. The seed is derived as specified in Section 3.2 of [I-D.selander-ace-cose-ecdhe], with other=exchange_hash, AlgorithmID="EDHOC IKE seed" and keyDataLength equal to the key length of the SKEYSEED secret defined in Section 2.14 of [RFC7296]. After that, the derived seed is written in the "seed" field of the "ipsec" structure, and accordingly used to derive IPsec key material as described in Section 2.1.

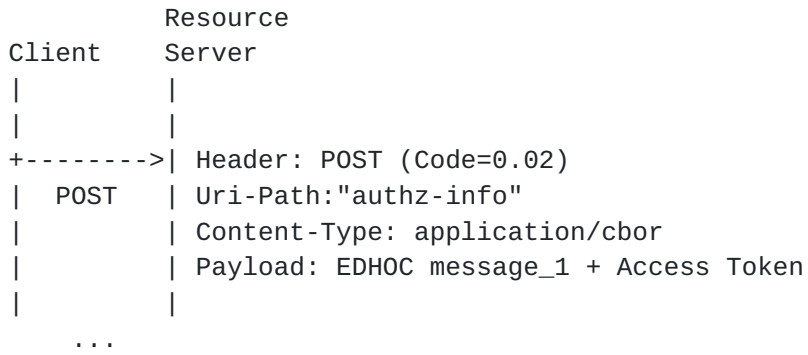

Figure 10: EDHOC used as Key Management Protocol

Authors' Addresses

Santiago Aragon
RISE SICS AB
Isafjordsgatan 22
Kista SE-164 29
Sweden

Email: santiago.aragon@stud.tu-darmstadt.de

Marco Tiloca
RISE SICS AB
Isafjordsgatan 22
Kista SE-164 29
Sweden

Phone: +46 70 604 65 01
Email: marco.tiloca@ri.se

Shahid Raza
RISE SICS AB
Isafjordsgatan 22
Kista SE-164 29
Sweden

Phone: +46 76 883 17 97
Email: shahid.raza@ri.se