Workgroup: Network Working Group
Internet-Draft:
draft-selander-ace-ake-authz-01
Published: 9 March 2020
Intended Status: Informational
Expires: 10 September 2020
Authors: G. Selander   J. Mattsson   M. Vucinic
Ericsson AB   Ericsson AB   INRIA
M. Richardson
Sandelman Software Works
A. Schellenbaum
Institute of Embedded Systems, ZHAW

Lightweight Authorization for Authenticated Key Exchange.

Abstract

This document describes a procedure for augmenting an authenticated Diffie-Hellman key exchange with third party assisted authorization targeting constrained IoT deployments (RFC 7228).

Note to Readers

Source for this draft and an issue tracker can be found at https://github.com/EricssonResearch/ace-ake-authz.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

For constrained IoT deployments [RFC7228] the overhead contributed by security protocols may be significant which motivates the specification of lightweight protocols that are optimizing, in particular, message overhead (see [I-D.ietf-lake-reqs]). This document describes a lightweight procedure for augmenting an authenticated Diffie-Hellman key exchange with third party assisted authorization.

The procedure involves a device, a domain authenticator and an authorization server. The device and authenticator perform mutual authentication and authorization, assisted by the authorization server which provides relevant authorization information to the device (a "voucher") and the authenticator.

The protocol specified in this document optimizes the message count by performing authorization and enrollment in parallel with authentication, instead of in sequence which is common for network access. It further reuses protocol elements from the authentication protocol leading to reduced message sizes on constrained links.

The specification assumes a lightweight AKE protocol [I-D.ietf-lake-reqs] between device and authenticator, and defines the integration of a lightweight authorization procedure. This enables a secure target interaction in few message exchanges. In this document we consider the target interaction to be "enrollment", for example certificate enrollment (such as [I-D.ietf-ace-coap-est]) or joining a network for the first time (e.g. [I-D.ietf-6tisch-minimal-security]), but it can be applied to authorize other target interactions.

This protocol is applicable to a wide variety of settings, and can be mapped to different authorization architectures. This document specifies a profile of the ACE framework [I-D.ietf-ace-oauth-authz]. Other settings such as EAP [RFC3748] are out of scope for this specification.

1.1. Terminology

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.

2. Problem Description

The (potentially constrained) device wants to enroll into a domain over a constrained link. The device authenticates and enforces authorization of the (non-constrained) domain authenticator with the help of a voucher, and makes the enrollment request. The domain authenticator authenticates the device and authorizes its enrollment. Authentication between device and domain authenticator is made with a lightweight authenticated Diffie-Hellman key exchange protocol (LAKE, [I-D.ietf-lake-reqs]). The procedure is assisted by a (non-constrained) authorization server located in a non-constrained network behind the domain authenticator providing information to the device and to the domain authenticator.

The objective of this document is to specify such a protocol which is lightweight over the constrained link and reuses elements of the LAKE. See illustration in Figure 1.

```
                     Voucher
                LAKE  Info
  +----------+  |    |   +----------------+ Voucher  +----------------+
  |          |  |    |   |                | Request  |                |
  |  Device  |--|----o-->|    Domain      |--------->| Authorization  |
  |          |<-|---o----| Authenticator  |<---------|     Server     |
  |    (U)   |--|---|--->|      (V)       | Voucher  |       (W)      |
  |          |      |    |                | Response |                |
  +----------+      |    +----------------+          +----------------+
                    Voucher
```

Figure 1: Overview and example of message content. Voucher Info and Voucher are sent together with LAKE messages.

3. Assumptions

3.1. Device

The device is pre-provisioned with an identity ID and asymmetric key credentials: a private key, a public key (PK_U), and optionally a public key certificate Cert(PK_U), issued by a trusted third party such as e.g. the device manufacturer, used to authenticate to the domain authenticator. The ID may be a reference or pointer to the certificate.

The device is also provisioned with information about its authorization server:

* At least one static public DH key of the authorization server (G_W) used to ensure secure communication with the device (see Section 4.1).

* Location information about the authorization server (LOC_W), e.g. its domain name. This information may be available in the device certificate Cert(PK_U).

3.2. Domain Authenticator

The domain authenticator has a private key and a corresponding public key PK_V used to authenticate to the device.

The domain authenticator needs to be able to locate the authorization server of the device for which the LOC_W is expected to be sufficient. The communication between domain authenticator and authorization server is mutually authenticated and protected. Authentication credentials and communication security used with the domain authenticator is out of scope, except for as specified below in this section.

The domain authenticator may in principle use differents credentials for authenticating to the authorization server and to the device, for which PK_V is used. However, the domain authenticator MUST prove possession of private key of PK_V to the authorization server since the authorization server is asserting (by means of the voucher to the device) that this credential belongs to the domain authenticator.

In this version of the draft it is assumed that the domain authenticator authenticates to the authorization server with PK_V using some authentication protocol providing proof of possession of the private key, for example TLS 1.3 [RFC8446]. A future version of this draft may specify explicit proof of possession of the private key of PK_V in the voucher request, e.g., by including a signature of the voucher request with the private key of PK_V.

3.3. Authorization Server

The authorization server has a private DH key corresponding to G_W, which is used to secure the communication with the device (see Section 4.1).

Authentication credentials and communication security used with the domain authenticator is out of scope, except for the need to verify the possession of the private key of PK_V as specified in Section 3.2.

The authorization server provides to the device the authorization decision for enrollment with the domain authenticator in the form of a voucher. The authorization server provides information to the domain authenticator about the device, such as the the device's certificate Cert(PK_U).

The authorization server needs to be available during the execution of the protocol.

3.4. Lightweight AKE

We assume a Diffie-Hellman key exchange protocol complying with the LAKE requirements [I-D.ietf-lake-reqs]. Specifically we assume for the LAKE:

* Three messages
* CBOR encoding
* The ephemeral public Diffie-Hellman key of the device, G_X, is sent in message 1. G_X is also used as ephemeral key and nonce in an ECIES scheme between device and authorization server.
* The public authentication key of the domain authenticator, PK_V, is sent in message 2.
* Support for Auxilliary Data AD1-3 in messages 1-3 as specified in section 2.5 of [I-D.ietf-lake-reqs].
* Cipher suite negotiation where the device can propose ECDH curves restricted by its available public keys of the authorization server.

4. The Protocol

Three security sessions are going on in parallel (see Figure 2):

* Between device (U) and (domain) authenticator (V),
* between authenticator and authorization server (W), and
* between device and authorization server mediated by the authenticator.

The content of the LAKE messages (see Section 3.4) is highlighted with brackets in the figure below (Figure 2) using the notation of EDHOC [I-D.selander-lake-edhoc]. The content includes:

* G_X: the x-coordinate of the ephemeral public Diffie-Hellman key of party U

* ID_CRED_V: data enabling the party U to obtain the credentials containing the public authentication key of V

* Sig(V;): a signature made with the private authentication key of V

* Sig(U;): a signature made with the private authentication key of U

We study each security session in turn, starting with the last.

```
U                                    V                              W
|                (G_X)               |                              |
|  AD1=(LOC_W, CC, AEAD(K_1; ID_U))  |                              |
+----------------------------------->|                              |
|           LAKE message 1           |G_X, PK_V, CC, AEAD(K_1; ID_U)|
|                                    +----------------------------->|
|                                    |    Voucher Request (VREQ)    |
|                                    |                              |
|                                    |      CERT_PK_U, Voucher      |
|                                    |<-----------------------------+
|        (ID_CRED_V, Sig(V;))        |    Voucher Response (VRES)   |
|             AD2=Voucher            |                              |
|<-----------------------------------+                              |
|           LAKE message 2           |                              |
|                                    |                              |
|              (Sig(U;))             |                              |
+----------------------------------->|                              |
|           LAKE message 3           |                              |

where Voucher = AEAD(K_2; V_TYPE, PK_V, G_X, ID_U)
```

Figure 2: W-assisted authorization of AKE between U and V. Relevant content from the LAKE protocol between U and V with auxiliary data AD1 and AD2. The Voucher Request/Response Protocol between V and W.

4.1. Device <-> Authorization Server

The communication between device and authorization server is carried out via the authenticator protected between the endpoints (protocol between U and W in Figure 2) using an ECIES hybrid encryption scheme (see [I-D.irtf-cfrg-hpke]): The device uses the private key corresponding to its ephemeral DH key G_X generated for LAKE message 1 (see Section 4.2) together with the static public DH key of the authorization server G_W to generate a shared secret G_XW. The shared secret is used to derive AEAD encryption keys to protect data between device and authorization server. The data is carried in AD1 and AD2 (between device and authenticator) and in Voucher Request/ Response (between authenticator and authorization server).

TODO: Reference relevant ECIES scheme in [I-D.irtf-cfrg-hpke].

TODO: Define derivation of encryption keys (K_1, K_2) and nonces (N_1, N_2) for the both directions

AD1 SHALL be the following CBOR sequence containing voucher information:

```
AD1 = (
    LOC_W:           tstr,
    CC:              bstr,
    CIPHERTEXT_RQ:   bstr
)
```

where

* LOC_W is location information about the authorization server

* CC is a crypto context identifier for the security context between the device and the authorization server

* 'CIPHERTEXT_RQ' is the authenticated encrypted identity of the device with CC as Additional Data, more specifically:

'CIPHERTEXT_RQ' is 'ciphertext' of COSE_Encrypt0 (Section 5.2-5.3 of [RFC8152]) computed from the following:

* the secret key K_1

* the nonce N_1

* 'protected' is a byte string of size 0

* 'plaintext and 'external_aad' as below:

```
plaintext = (
    ID:              bstr
 )

external_aad = (
    CC:              bstr
 )
```

where

* ID is the identity of the device, for example a reference or pointer to the device certificate

* CC is defined above.

AD2 SHALL be the Voucher, defined in the next section.

```
AD2 = (
    Voucher:        bstr
)
```

4.1.1. Voucher

The Voucher is essentially a Message Authentication Code binding the identity of the authenticator to the first message sent from the device in the LAKE protocol.

More specifically 'Voucher' is the 'ciphertext' of COSE_Encrypt0 (Section 5.2 of [RFC8152]) computed from the following:

* the secret key K_2

* the nonce N_2

* 'protected' is a byte string of size 0

* 'plaintext' is empty (plaintext = nil)

* 'external_aad' as below:

```
external_aad = bstr .cbor external_aad_array

external_aad_array = [
    voucher_type:  int,
    PK_V:          bstr,
    G_X:           bstr,
    CC:            bstr,
    ID:            bstr
]
```

where

* 'voucher-type' indicates the kind of voucher used

* PK_V is a COSE_Key containing the public authentication key of the authenticator. The public key must be an Elliptic Curve Diffie-Hellman key, COSE key type 'kty' = 'EC2' or 'OKP'.

  - COSE_Keys of type OKP SHALL only include the parameters 1 (kty), -1 (crv), and -2 (x-coordinate). COSE_Keys of type EC2 SHALL only include the parameters 1 (kty), -1 (crv), -2 (x-coordinate), and -3 (y-coordinate). The parameters SHALL be encoded in decreasing order.

* G_X is the ephemeral key of the device sent in the first LAKE message

* CC and ID are defined in Section 4.1

All parameters, except 'voucher-type', are as received in the voucher request (see Section 4.3).

TODO: Consider making the voucher a CBOR Map to indicate type of voucher, to indicate the feature (cf. Section 4.3)

4.2. Device <-> Authenticator

The device and authenticator run the LAKE protocol authenticated with public keys (PK_U and PK_V) of the device and the authenticator, see protocol between U and V in Figure 2. The normal processing of the LAKE is omitted here.

4.2.1. Message 1

4.2.1.1. Device processing

The device selects a cipher suite with an ECDH curve satisfying the static public DH key G_W of the authorization server. As part of the normal LAKE processing, the device generates the ephemeral public key G_X to be sent in LAKE message 1. A new G_X MUST be generated for each execution of the protocol. The ephemeral key G_X is reused in the ECIES scheme, see Section 4.1.

The device sends LAKE message 1 with AD1 as specified in Section 4.1.

4.2.1.2. Authenticator processing

The authenticator receives LAKE message 1 from the device, which triggers the exchange of voucher related data with the authorization server as described in Section 4.3.

4.2.2. Message 2

4.2.2.1. Authenticator processing

The authenticator sends LAKE message 2 to the device with the voucher (see Section 4.1) in AD2. The public key PK_V is encoded in the way public keys are encoded in the LAKE protocol.

4.2.2.2. Device processing

The device MUST verify the Voucher using its ephemeral key G_X sent in message 1 and PK_V received in message 2. If the Voucher does not verify, the device MUST discontinue the protocol.

4.2.3. Message 3

4.2.3.1. Device processing

The device sends message 3. AD3 depends on the kind of enrollment the device is requesting. It may e.g. be a CBOR encoded Certificate Signing Request, see [I-D.raza-ace-cbor-certificates].

4.2.3.2. Authenticator processing

The authenticator MUST NOT verify the signature Sig(U;) (see Figure 2) in LAKE message 3 with the PK_U included in message 3. Instead, the signature MUST be verified with the public key included in Cert(PK_U) (see Section 4.3.2) received from the authorization server. This way, the authenticator can make sure that message 3 is signed by the right entity trusted by the authorization server.

4.3. Authenticator <-> Authorization Server

The authenticator and authorization server are assumed to have secure communication, for example TLS 1.3 authenticated with certificates, protecting the Voucher Request/Response Protocol (see protocol between V and W in Figure 2).

4.3.1. Voucher Request

The authenticator sends the voucher request to the authorization server. The Voucher_Request SHALL be a CBOR array as defined below:

```
Voucher_Request = [
    PK_V:            bstr,
    G_X:             bstr,
    CC:              bstr,
    CIPHERTEXT_RQ:   bstr
]
```

where the parameters are defined in Section 4.1.

4.3.2. Voucher Response

The authorization server decrypts the identity of the device and looks up its certificate, Cert(PK_U). The authorization server sends the voucher response to the authenticator. The Voucher_Response SHALL be a CBOR array as defined below:

```
Voucher_Response = [
    CERT_PK_U:      bstr,
    Voucher:        bstr
]
```

where

* CERT_PK_U is the device certificate of the public key PK_U, Cert(PK_U), issued by a trusted third party, intended to be verified by the authenticator. The format of this certificate is out of scope.

* The voucher is defined in Section 4.1

TODO: The voucher response may contain a "Voucher-info" field as an alternative to make the Voucher a CBOR Map (see Section 4.1)

5. ACE Profile

This section defines the profile of the ACE framework (see Appendix C of [I-D.ietf-ace-oauth-authz]).

U plays the role of the ACE Resource Server (RS). V plays the role of the ACE Client (C). W plays the role of the ACE Authorization Server (AS).

C and RS use the Auxiliary Data in the LAKE protocol to communicate. C and RS use the LAKE protocol to protect their communication. LAKE also provides mutual authentication of C and RS, assisted by the AS.

5.1. Protocol Overview

```
   RS                                C                      AS
   |         LAKE Message 1          |                      |
   |  AD1=AS Request Creation Hints  |                      |
   |-------------------------------->|     POST /token      |
   |                                 |--------------------->|
   |                                 |                      |
   |                                 | Access Token +       |
   |          LAKE Message 2         |  Access Information  |
   |         AD2=Access Token        |<---------------------|
   |<--------------------------------|                      |
   |          LAKE Message 3         |                      |
   |-------------------------------->|                      |
```

Figure 3: Overview of the protocol mapping to ACE

RS proactively sends the AS Request Creation Hints message to C to signal the information on where C can reach the AS. RS piggybacks the AS Request Creation Hints message using Auxiliary Data of the LAKE message 1. Before continuing the LAKE handshake, based on the AS Request Creation Hints information, C sends a POST request to the token endpoint at the AS requesting the access token. The AS issues an assertion to C that is cryptographically protected based on the secret shared between the AS and RS. In this profile, the assertion is encoded as a Bearer Token. C presents this token to RS in the Auxiliary Data of the LAKE message 2. RS verifies the token based on the possession of the shared secret with the AS and authenticates C.

5.2. AS Request Creation Hints

Parameters that can appear in the AS Request Creation Hints message are specified in Section 5.1.2. of [I-D.ietf-ace-oauth-authz]. RS MUST use the "AS" parameter to transport LOC_W, i.e. an absolute URI where C can reach the AS. RS MUST use the "audience" parameter to transport the CBOR sequence consisting of two elements: CC, the crypto context; CIPHERTEXT_RQ, the authenticated encrypted identity of the RS. The "cnonce" parameter MUST be implied to G^X, i.e. the ephemeral public key of the RS in the underlying LAKE exchange. The "cnonce" parameter is not carried in the AS Request Creation Hints message for byte saving reasons. AS Request Creation Hints MUST be carried within Auxiliary Data of the LAKE message 1 (AD1).

An example AD1 value in CBOR diagnostic notation is shown below:

```
AD1:
{
    "AS" : "coaps://as.example.com/token",
    "audience": << h'73',h'737570657273...' >>
}
```

5.3. Client-to-AS Request

The protocol that provides the secure channel between C and the AS is out-of-scope. This can, for example, be TLS or DTLS. What is important is that the two peers are mutually authenticated, and that the secure channel provides message integrity, confidentiality and freshness. It is also necessary for the AS to be able to extract the public key of C used in the underlying security handshake.

C sends the POST request to the token endpoint at the AS following Section 5.6.1. of [I-D.ietf-ace-oauth-authz]. C MUST set the "audience" parameter to the value received in AS Request Creation Hints. C MUST set the "cnonce" parameter to G^X, the ephemeral public key of RS in the LAKE handshake.

An example exchange using CoAP and CBOR diagnostic notation is shown below:

```
Header: POST (Code=0.02)
Uri-Host: "as.example.com"
Uri-Path: "token"
Content-Format: "application/ace+cbor"
Payload:
{
    "audience" : << h'73',h'737570657273...' >>
    "cnonce" : h'756E73686172...'
}
```

5.4. AS-to-Client Response

Given successful authorization of C at the AS, the AS responds by issuing a Bearer token and retrieves the certificate of RS on behalf of C. The access token and the certificate are passed back to C, who uses it to complete the LAKE handshake. This document extends the ACE framework by registering a new Access Information parameter:

rsp_ad: OPTIONAL. Carries additional information from the AS to C associated with the access token.

When responding to C, the AS MUST set the "ace_profile" parameter to "lake". The AS MUST set the "token_type" parameter to "Bearer". The access token MUST be formatted as specified in Section 4.1.1. The AS MUST set the "rsp_ad" parameter to the certificate of RS. To be able to do so, AS first needs to decrypt the audience value, and based on it retrieve the corresponding RS certificate.

An example AS response to C is shown below:

```
2.01 Created
Content-Format: application/ace+cbor
Max-Age: 3600
Payload:
{
    "ace_profile" : "lake",
    "token_type" : "Bearer",
    "access_token" : h'666F72657175617274657 2...',
    "rsp_ad" : h'61726973746F64656D6F637261746963616C...'
}
```

6. Security Considerations

TODO: Identity protection of device

TODO: Use of G_X as ephemeral key between device and authenticator, and between device and authorization server

TODO: Remote attestation

7. IANA Considerations

TODO: CC registry

TODO: Voucher type registry

TODO: register rsp_ad ACE parameter

8. Informative References

[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>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <https://www.rfc-editor.org/info/rfc3748>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/

RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

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

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[I-D.ietf-lake-reqs]
Vucinic, M., Selander, G., Mattsson, J., and D. Garcia-Carillo, "Requirements for a Lightweight AKE for OSCORE", Work in Progress, Internet-Draft, draft-ietf-lake-reqs-01, 19 February 2020, <http://www.ietf.org/internet-drafts/draft-ietf-lake-reqs-01.txt>.

[I-D.ietf-ace-oauth-authz]
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth)", Work in Progress, Internet-Draft, draft-ietf-ace-oauth-authz-33, 6 February 2020, <http://www.ietf.org/internet-drafts/draft-ietf-ace-oauth-authz-33.txt>.

[I-D.raza-ace-cbor-certificates]
Raza, S., Hoglund, J., Selander, G., Mattsson, J., and M. Furuhed, "CBOR Profile of X.509 Certificates", Work in Progress, Internet-Draft, draft-raza-ace-cbor-certificates-03, 20 December 2019, <http://www.ietf.org/internet-drafts/draft-raza-ace-cbor-certificates-03.txt>.

[I-D.irtf-cfrg-hpke] Barnes, R. and K. Bhargavan, "Hybrid Public Key Encryption", Work in Progress, Internet-Draft, draft-irtf-cfrg-hpke-02, 4 November 2019, <http://www.ietf.org/internet-drafts/draft-irtf-cfrg-hpke-02.txt>.

[I-D.ietf-ace-coap-est] Stok, P., Kampanakis, P., Richardson, M., and S. Raza, "EST over secure CoAP (EST-coaps)", Work in Progress, Internet-Draft, draft-ietf-ace-coap-est-18, 6

January 2020, <http://www.ietf.org/internet-drafts/draft-ietf-ace-coap-est-18.txt>.

[I-D.ietf-6tisch-minimal-security]
Vucinic, M., Simon, J., Pister, K., and M. Richardson, "Constrained Join Protocol (CoJP) for 6TiSCH", Work in Progress, Internet-Draft, draft-ietf-6tisch-minimal-security-15, 10 December 2019, <http://www.ietf.org/internet-drafts/draft-ietf-6tisch-minimal-security-15.txt>.

[I-D.selander-lake-edhoc]
Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", Work in Progress, Internet-Draft, draft-selander-lake-edhoc-00, 4 November 2019, <http://www.ietf.org/internet-drafts/draft-selander-lake-edhoc-00.txt>.

Authors' Addresses

Goeran Selander
Ericsson AB

Email: goran.selander@ericsson.com

John Preuss Mattsson
Ericsson AB

Email: john.mattsson@ericsson.com

Malisa Vucinic
INRIA

Email: malisa.vucinic@inria.fr

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

Aurelio Schellenbaum
Institute of Embedded Systems, ZHAW

Email: aureliorubendario.schellenbaum@zhaw.ch