Network Working Group G. Selander
Internet-Draft J. Mattsson
Intended status: Informational Ericsson AB
Expires: August 9, 2020 M. Vucinic

INRIA

M. Richardson

Sandelman Software Works

February 06, 2020

Lightweight Authorization for Authenticated Key Exchange. draft-selander-ace-ake-authz-00

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).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 9, 2020.

Copyright Notice

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

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 a AAA server. The device performs mutual authentication and authorization of the authenticator, assisted by the AAA server which provides

relevant authorization information to the device in the form of a "voucher".

The protocol specified in this document optimizes the message count by performing authorization and enrolment 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 "enrolment", for example certificate enrolment or joining a network for the first time, but it can be applied to authorize other target interactions.

This protocol is applicable in a wide variety of settings, e.g. an enterprise network using EAP [RFC3748].

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 enrol 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 enrolment request. The domain authenticator authenticates the device and authorizes its enrolment. 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) AAA 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.

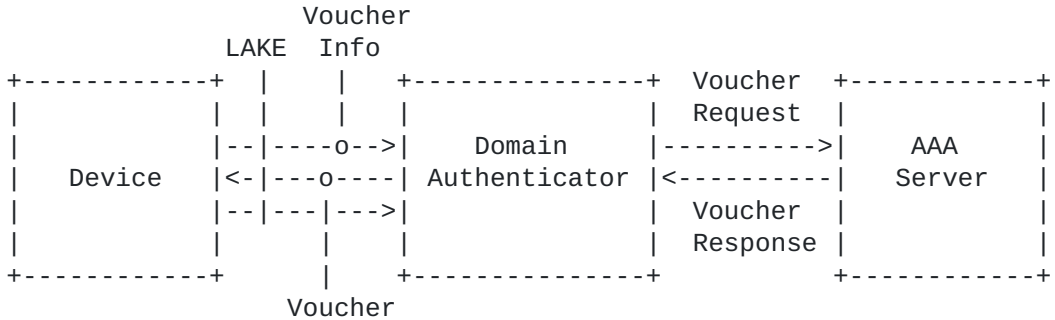

Figure 1: Overview and example of message content. Voucher Info and Voucher are sent together with LAKE messages.

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_D), and optionally a public key certificate Cert(PK_D) 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 AAA server:

- At least one static public DH key of the AAA server (G_S) used to ensure secure communication with the device (see Section 4.1).

- Location information about the AAA server (LOC_S), e.g. its domain name. This information may be available in the device certificate Cert(PK_D).

3.2. Domain Authenticator

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

The domain authenticator needs to be able to locate the AAA server of the device for which the LOC_S is expected to be sufficient. The communication between domain authenticator and AAA server is mutually authenticated and protected. Authentication credentials used with the AAA server is out of scope. How this communication is established and secured (typically TLS) is out of scope.

3.3. AAA Server

The AAA server has a private DH key corresponding to G_S, 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.

The AAA server provides the authorization decision for enrolment to the device in the form of a CBOR encoded voucher. The AAA server provides information to the domain authenticator about the device, such as the the device's certificate Cert(PK_D).

The AAA 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 the ECIES scheme between device and AAA server.
- The static public key of the domain authenticator, PK_A, 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 AAA server.

4. The Protocol

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

- Between device and (domain) authenticator,
- between authenticator and AAA server, and
- between device and AAA server mediated by the authenticator.

We study each in turn, starting with the last.

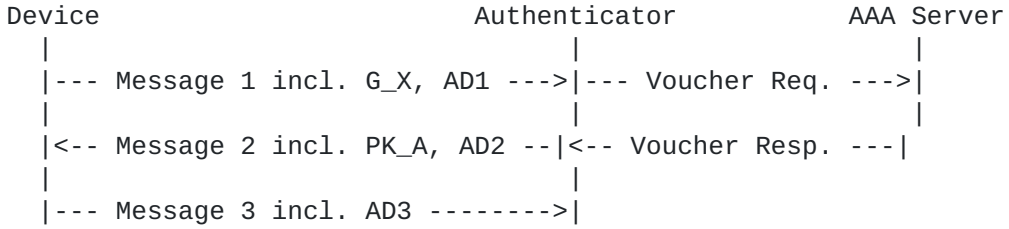

Figure 2: The Protocol

4.1. Device <-> AAA Server

The communication between device and AAA server is carried out via the authenticator protected between the endpoints using an ECIES hybrid encryption scheme (see [I-D.irtf-cfrg-hpke]): The device uses the private key of its ephemeral DH key G_X generated for LAKE message 1 (see Section 4.2) together with the static public DH key of the AAA server G_S to generate a shared secret G_XS. The shared secret is used to derive AEAD encryption keys to protect data between device and AAA server. The data is carried in AD1 and AD2 (between device and authenticator) and in voucher request/response (between authenticator and AAA server).

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

TODO: Define derivation of encryption keys (k_rq, k_rs) and nonces (n_rq, n_rs) for both directions

AD1 SHALL be the following CBOR sequence containing voucher information:

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

where

- LOC_S is location information about the AAA server
- CC is a crypto context identifier for the security context between the device and the AAA 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_rq
- the nonce n_rq
- '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 a CBOR sequence of one item, 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 the following:

- the secret key k_rs

- the nonce n_rs

- 'protected' is a byte string of size 0

- 'plaintext' is empty (plaintext = nil)

- 'external_aad' as below:

```
external_aad = bstr .cbor external_aad_arr
```

```
external_aad_arr = [
    voucher_type:  int,
    PK_A:          bstr,
    G_X:           bstr,
    CC:            bstr,
    ID:            bstr
]
```

where

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

- PK_A 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_D and PK_A) of the device and the authenticator. 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_S of the AAA 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 same ephemeral key is used 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 AAA 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_A 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_A 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 enrolment 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 receives message 3.

4.3. Authenticator <-> AAA Server

The authenticator and AAA server are assumed to have secure communication, for example based on TLS authenticated with certificates.

4.3.1. Voucher Request

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

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

where the parameters are defined in Section 4.1.

4.3.2. Voucher Response

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

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

where

- CERT_PK_D is the device certificate of the public key PK_D, issued by a trusted third party, intended to be verified by the authenticator. The format of this certificate is out of scope.

- 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. Security Considerations

TODO: Identity protection of device

TODO: How can the AAA server attest the received PK_A?

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

TODO: Remote attestation

6. IANA Considerations

TODO: CC registry

TODO: Voucher type registry

7. Informative References

[I-D.ietf-lake-reqs]
Vucinic, M., Selander, G., Mattsson, J., and D. Garcia-Carillo, "Requirements for a Lightweight AKE for OSCORE", draft-ietf-lake-reqs-00 (work in progress), December 2019.

[I-D.irtf-cfrg-hpke]
Barnes, R. and K. Bhargavan, "Hybrid Public Key Encryption", draft-irtf-cfrg-hpke-02 (work in progress), November 2019.

[I-D.raza-ace-cbor-certificates]
Raza, S., Hoglund, J., Selander, G., Mattsson, J., and M. Furuhed, "CBOR Profile of X.509 Certificates", draft-raza-ace-cbor-certificates-03 (work in progress), December 2019.

[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>.

Authors' Addresses

Goeran Selander
Ericsson AB

Email: goran.selander@ericsson.com

John 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