Workgroup: ACE Working Group
Internet-Draft:
draft-selander-ace-coap-est-oscore-04
Published: 2 November 2020
Intended Status: Standards Track
Expires: 6 May 2021
Authors: G. Selander S. Raza M. Furuhed M. Vucinic
Ericsson AB RISE Nexus INRIA
T. Claeys
INRIA

Protecting EST Payloads with OSCORE

Abstract

This document specifies public-key certificate enrollment procedures protected with lightweight application-layer security protocols suitable for Internet of Things (IoT) deployments. The protocols leverage payload formats defined in Enrollment over Secure Transport (EST) and existing IoT standards including the Constrained Application Protocol (CoAP), Concise Binary Object Representation (CBOR) and the CBOR Object Signing and Encryption (COSE) format.

Table of Contents

1. Introduction

One of the challenges with deploying a Public Key Infrastructure (PKI) for the Internet of Things (IoT) is certificate enrollment, because existing enrollment protocols are not optimized for constrained environments [RFC7228].

One optimization of certificate enrollment targeting IoT deployments is specified in EST-coaps ([I-D.ietf-ace-coap-est]), which defines a version of Enrollment over Secure Transport [RFC7030] for transporting EST payloads over CoAP [RFC7252] and DTLS [RFC6347], instead of secured HTTP.

This document describes a method for protecting EST payloads over CoAP or HTTP with OSCORE [RFC8613]. OSCORE specifies an extension to CoAP which protects the application layer message and can be applied independently of how CoAP messages are transported. OSCORE can also be applied to CoAP-mappable HTTP which enables end-to-end security for mixed CoAP and HTTP transfer of application layer data. Hence EST payloads can be protected end-to-end independent of underlying transport and through proxies translating between between CoAP and HTTP.

OSCORE is designed for constrained environments, building on IoT standards such as CoAP, CBOR [RFC7049] and COSE [RFC8152], and has in particular gained traction in settings where message sizes and the number of exchanged messages needs to be kept at a minimum, such as 6TiSCH [I-D.ietf-6tisch-minimal-security], or for securing multicast CoAP messages [I-D.ietf-core-oscore-groupcomm]. Where OSCORE is implemented and used for communication security, the reuse of OSCORE for other purposes, such as enrollment, reduces the code footprint.

In order to protect certificate enrollment with OSCORE, the necessary keying material (notably, the OSCORE Master Secret, see [RFC8613]) needs to be established between EST-oscore client and EST-oscore server. For this purpose we assume the use of the lightweight authenticated key exchange protocol EDHOC [I-D.ietf-lake-edhoc]. Other methods for key establishment are described in Appendix A.

Other ways to optimize the performance of certificate enrollment and certificate based authentication described in this draft include the use of:

* Compact representations of X.509 certificates (see [I-D.mattsson-cose-cbor-cert-compress])

* Certificates by reference (see [I-D.ietf-cose-x509])

* Compact representations of EST payloads (see Appendix B)

1.1. Operational Differences with EST-coaps

The protection of EST payloads defined in this document builds on EST-coaps [I-D.ietf-ace-coap-est] but transport layer security is replaced, or complemented, by protection of the transfer- and

application layer data (i.e., CoAP message fields and payload). This specification deviates from EST-coaps in the following respects:

- The DTLS record layer is replaced, or complemented, with OSCORE.

- The DTLS handshake is replaced, or complemented, with the lightweight authenticated key exchange protocol EDHOC [I-D.ietf-lake-edhoc], and makes use of the following features:

  - Authentication based on certificates is complemented with authentication based on raw public keys.

  - Authentication based on signature keys is complemented with authentication based on static Diffie-Hellman keys, for certificates/raw public keys.

  - Authentication based on certificate by value is complemented with authentication based on certificate/raw public keys by reference.

- One new EST function, /rpks, is defined for installation of compact explicit TAs in the EST client.

- The EST payloads protected by OSCORE can be proxied between constrained networks supporting CoAP/CoAPs and non-constrained networks supporting HTTP/HTTPs with a CoAP-HTTP proxy protection without any security processing in the proxy (see Section 5). The concept "Registrar" and its required trust relation with EST server as described in Section 6 of [I-D.ietf-ace-coap-est] is therefore redundant.

So, while the same authentication scheme (Diffie-Hellman key exchange authenticated with transported certificates) and the same EST payloads as EST-coaps also apply to EST-oscore, the latter specifies other authentication schemes and a new matching EST function. The reason for these deviations is that a significant overhead can be removed in terms of message sizes and round trips by using a different handshake, public key type or transported credential, and those are independent of the actual enrollment procedure.

Appendix A discusses yet other authentication and secure communication methods.

2. 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.

This document uses terminology from [I-D.ietf-ace-coap-est] which in turn is based on [RFC7030] and, in turn, on [RFC5272].

The term "Trust Anchor" follows the terminology of [RFC6024]: "A trust anchor represents an authoritative entity via a public key and associated data. The public key is used to verify digital signatures, and the associated data is used to constrain the types of information for which the trust anchor is authoritative." One example of specifying more compact alternatives to X.509 certificates for exchanging trust anchor information is provided by the TrustAnchorInfo structure of [RFC5914], the mandatory parts of which essentially is the SubjectPublicKeyInfo structure [RFC5280], i.e., an algorithm identifier followed by a public key.

3. Authentication

This specification replaces the DTLS handshake in EST-coaps with the lightweight authenticated key exchange protocol EDHOC [I-D.ietf-lake-edhoc]. During initial enrollment the EST-oscore client and server run EDHOC [I-D.ietf-lake-edhoc] to authenticate and establish the OSCORE security context with which the EST payloads are protected.

EST-oscore clients and servers MUST perform mutual authentication. The EST server and EST client are responsible for ensuring that an acceptable cipher suite is negotiated. The client MUST authenticate the server before accepting any server response. The server MUST authenticate the client and provide relevant information to the CA for decision about issuing a certificate.

3.1. EDHOC

EDHOC supports authentication with certificates/raw public keys (referred to as "credentials"), and the credentials may either be transported in the protocol, or referenced. This is determined by the identifier of the credential of the endpoint, ID_CRED_x for x= Initiator/Responder, which is transported in an EDHOC message. This identifier may be the credential itself (in which case the credential is transported), or a pointer such as a URI to the credential (e.g., x5t, see [I-D.ietf-cose-x509]) or some other identifier which enables the receiving endpoint to retrieve the credential.

3.2. Certificate-based Authentication

EST-oscore, like EST-coaps, supports certificate-based authentication between EST client and server. In this case the

client MUST be configured with an Implicit or Explicit Trust Anchor (TA) [RFC7030] database, enabling the client to authenticate the server. During the initial enrollment the client SHOULD populate its Explicit TA database and use it for subsequent authentications.

The EST client certificate SHOULD conform to [RFC7925]. The EST client and/or EST server certificate MAY be a (natively signed) CBOR certificate [I-D.mattsson-cose-cbor-cert-compress].

3.3. Channel Binding

The [RFC5272] specification describes proof-of-possession as the ability of a client to prove its possession of a private key which is linked to a certified public key. In case of signature key, a proof-of-possession is generated by the client when it signs the PKCS#10 Request during the enrollment phase. Connection-based proof-of-possession is OPTIONAL for EST-oscore clients and servers.

When desired the client can use the EDHOC-Exporter API to extract channel-binding information and provide a connection-based proof-of possession. Channel-binding information is obtained as follows

edhoc-unique = EDHOC-Exporter("EDHOC Unique", length),

where length equals the desired length of the edhoc-unique byte string. The client then adds the edhoc-unique byte string as a challengePassword (see Section 5.4.1 of [RFC2985]) in the attributes section of the PKCS#10 Request to prove to the server that the authenticated EDHOC client is in possession of the private key associated with the certification request, and signed the certification request after the EDHOC session was established.

3.4. Optimizations

* The last message of the EDHOC protocol, message_3, MAY be combined with an OSCORE request, enabling authenticated Diffie-Hellman key exchange and a protected CoAP request/response (which may contain an enrolment request and response) in two round trips [I-D.palombini-core-oscore-edhoc].

* The certificates MAY be compressed, e.g. using the CBOR encoding defined in [I-D.mattsson-cose-cbor-cert-compress].

* The certificate MAY be referenced instead of transported [I-D.ietf-cose-x509]. The EST-oscore server MAY use information in the credential identifier field of the EDHOC message (ID_CRED_x) to access the EST-oscore client certificate, e.g., in a directory or database provided by the issuer. In this case the certificate may not need to be transported over a constrained link between EST client and server.

*Conversely, the response to the PKCS#10 request MAY be a reference to the enrolled certificate rather than the certificate itself. The EST-oscore server MAY in the enrolment response to the EST-oscore client include a pointer to a directory or database where the certificate can be retrieved.

3.5. RPK-based Trust Anchors

A trust anchor is commonly a self-signed certificate of the CA public key. In order to reduce transport overhead, the trust anchor could be just the CA public key and associated data (see Section 2), e.g., the SubjectPublicKeyInfo, or a public key certificate without the signature. In either case they can be compactly encoded, e.g. using CBOR encoding [I-D.mattsson-cose-cbor-cert-compress]. A client MAY request an unsigned trust anchors using the /rpks function (see Section 4.2).

Client authentication can be performed with long-lived RPKs installed by the manufacturer. Re-enrollment requests can be authenticated through a valid certificate issued previously by the EST-oscore server or by using the key material available in the Implicit TA database.

TODO: Sanity check this. Review the use of Implicit TA vs. Explicit TA.

4. Protocol Design and Layering

EST-oscore uses CoAP [RFC7252] and Block-Wise [RFC7959] to transfer EST messages in the same way as [I-D.ietf-ace-coap-est]. Instead of DTLS record layer, OSCORE [RFC8613] is used to protect the EST payloads. Figure 1 below shows the layered EST-oscore architecture.

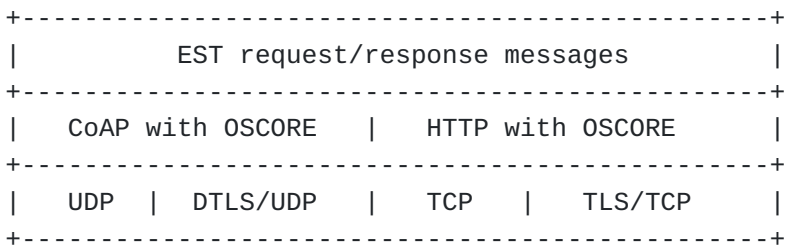

Figure 1: EST protected with OSCORE.

EST-oscore follows much of the EST-coaps and EST design.

4.1. Discovery and URI

The discovery of EST resources and the definition of the short EST-coaps URI paths specified in Section 5.1 of [I-D.ietf-ace-coap-est], as well as the new Resource Type defined in Section 9.1 of [I-D.ietf-ace-coap-est] apply to EST-oscore. Support for OSCORE is indicated by the "osc" attribute defined in Section 9 of [RFC8613], for example:

```
  REQ: GET /.well-known/core?rt=ace.est.sen

  RES: 2.05 Content
</est>; rt="ace.est";osc
```

4.2. Distribution of RPKs

The EST client can request a copy of the current CA public keys.

TODO: Map relevant parts of section 4.1 of RFC 7030 and other EST function related content from RFC7030 and EST-coaps.

RATIONALE: EST-coaps provides the /crts operation. A successful request from the client to this resource will be answered with a bag of certificates which is subsequently installed in the Explicit TA. Motivated by the specification of more compact trust anchors (see Section 2) we define here the new EST function /rpks which returns a set of RPKs to be installed in the Explicit TA database.

4.3. Mandatory/optional EST Functions

The EST-oscore specification has the same set of required-to-implement functions as EST-coaps. The content of Table 1 is adapted from Section 5.2 in [I-D.ietf-ace-coap-est] and uses the updated URI paths (see Section 4.1).

EST functions	EST-oscore implementation
/crts	MUST
/sen	MUST
/sren	MUST
/skg	OPTIONAL
/skc	OPTIONAL
/att	OPTIONAL

Table 1: Mandatory and optional EST-oscore functions

TODO: Add /rpks OPTIONAL

4.4. Payload formats

Similar to EST-coaps, EST-oscore allows transport of the ASN.1 structure of a given Media-Type in binary format. In addition, EST-oscore uses the same CoAP Content-Format Options to transport EST requests and responses . Table 2 summarizes the information from Section 5.3 in [I-D.ietf-ace-coap-est].

URI	Content-Format	#IANA
/crts	N/A (req)	-
	application/pkix-cert (res)	287
	application/pkcs-7-mime;smime-type=certs-only (res)	281
/sen	application/pkcs10 (req)	286
	application/pkix-cert (res)	287
	application/pkcs-7-mime;smime-type=certs-only (res)	281
/sren	application/pkcs10 (req)	286
	application/pkix-cert (res)	287
	application/pkcs-7-mime;smime-type=certs-only (res)	281
/skg	application/pkcs10 (req)	286
	application/multipart-core (res)	62
/skc	application/pkcs10 (req)	286
	application/multipart-core (res)	62
/att	N/A (req)	-
	application/csrattrs (res)	285

Table 2: EST functions and there associated Media-Type and IANA numbers

NOTE: CBOR is becoming a de facto encoding scheme in IoT settings. There is already work in progress on CBOR encoding of X.509 certificates [I-D.mattsson-cose-cbor-cert-compress], and this can be extended to other EST messages, see Appendix B.

4.5. Message Bindings

The EST-oscore message characteristics are identical to those specified in Section 5.4 of [I-D.ietf-ace-coap-est]. It is RECOMMENDED that

- The EST-oscore endpoints support delayed responses
- The endpoints supports the following CoAP options: OSCORE, Uri-Host, Uri-Path, Uri-Port, Content-Format, Block1, Block2, and Accept.
- The EST URLs based on https:// are translated to coap://, but with mandatory use of the CoAP OSCORE option.

4.6. CoAP response codes

See Section 5.5 in [I-D.ietf-ace-coap-est].

4.7. Message fragmentation

The EDHOC key exchange is optimized for message overhead, in particular the use of static DH keys instead of signature keys for authentication (e.g., method 3 of [I-D.ietf-lake-edhoc]). Together with various measures listed in this document such as CBOR payloads (Appendix B), CBOR certificates [I-D.mattsson-cose-cbor-cert-compress], certificates by reference (Section 3.4), and trust anchors without signature (Section 3.5), a significant reduction of message sizes can be achieved.

Nevertheless, depending on application, the protocol messages may become larger than available frame size resulting in fragmentation and, in resource constrained networks such as IEEE 802.15.4 where throughput is limited, fragment loss can trigger costly retransmissions.

It is RECOMMENDED to prevent IP fragmentation, since it involves an error-prone datagram reconstitution. To limit the size of the CoAP payload, this specification mandates the implementation of CoAP option Block1 and Block2 fragmentation mechanism [RFC7959] as described in Section 5.6 of [I-D.ietf-ace-coap-est].

4.8. Delayed Responses

See Section 5.7 in [I-D.ietf-ace-coap-est].

5. HTTP-CoAP Proxy

As noted in Section 6 of [I-D.ietf-ace-coap-est], in real-world deployments, the EST server will not always reside within the CoAP boundary. The EST-server can exist outside the constrained network in a non-constrained network that supports HTTP but not CoAP, thus requiring an intermediary CoAP-to-HTTP proxy.

Since OSCORE is applicable to CoAP-mappable HTTP (see Section 11 of [RFC8613]) the EST payloads can be protected end-to-end between EST client and EST server independent of transport protocol or potential transport layer security which may need to be terminated in the proxy, see Figure 2. Therefore the concept "Registrar" and its required trust relation with EST server as described in Section 6 of [I-D.ietf-ace-coap-est] is redundant.

The mappings between CoAP and HTTP referred to in Section 9.1 of [I-D.ietf-ace-coap-est] apply, and additional mappings resulting from the use of OSCORE are specified in Section 11 of [RFC8613].

OSCORE provides end-to-end security between EST Server and EST Client. The use of TLS and DTLS is optional.

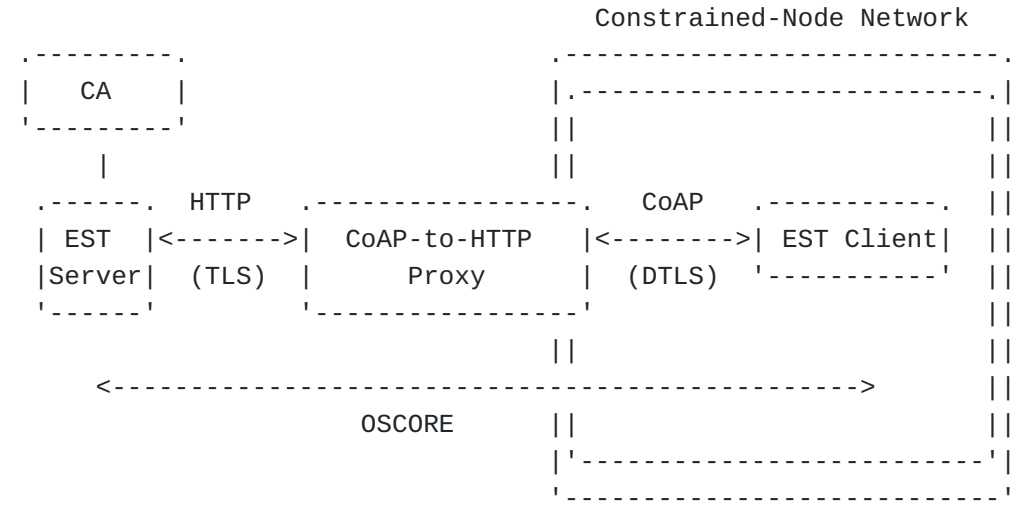

Figure 2: CoAP-to-HTTP proxy at the CoAP boundary.

6. Security Considerations

TBD

7. Privacy Considerations

TBD

8. IANA Considerations

9. Acknowledgments

10. References

10.1. Normative 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>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

[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>.

[RFC7925]
Tschofenig, H., Ed. and T. Fossati, "Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things", RFC 7925, DOI 10.17487/RFC7925, July 2016, <https://www.rfc-editor.org/info/rfc7925>.

[RFC7959] Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in the Constrained Application Protocol (CoAP)", RFC 7959, DOI 10.17487/RFC7959, August 2016, <https://www.rfc-editor.org/info/rfc7959>.

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

[RFC8613] Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019, <https://www.rfc-editor.org/info/rfc8613>.

[I-D.ietf-lake-edhoc]
Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", Work in Progress, Internet-Draft, draft-ietf-lake-edhoc-01, 2 August 2020, <http://www.ietf.org/internet-drafts/draft-ietf-lake-edhoc-01.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>.

10.2. Informative References

[RFC2985] Nystrom, M. and B. Kaliski, "PKCS #9: Selected Object Classes and Attribute Types Version 2.0", RFC 2985, DOI 10.17487/RFC2985, November 2000, <https://www.rfc-editor.org/info/rfc2985>.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, DOI 10.17487/RFC2986, November 2000, <https://www.rfc-editor.org/info/rfc2986>.

[RFC5272] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC)", RFC 5272, DOI 10.17487/RFC5272, June 2008, <https://www.rfc-editor.org/info/rfc5272>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5914] Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Format", RFC 5914, DOI 10.17487/RFC5914, June 2010, <https://www.rfc-editor.org/info/rfc5914>.

[RFC6024] Reddy, R. and C. Wallace, "Trust Anchor Management Requirements", RFC 6024, DOI 10.17487/RFC6024, October 2010, <https://www.rfc-editor.org/info/rfc6024>.

[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>.

[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>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/info/rfc7030>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

[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.ietf-ace-oscore-profile] Palombini, F., Seitz, L., Selander, G., and M. Gunnarsson, "OSCORE Profile of the Authentication and Authorization for Constrained Environments Framework", Work in Progress, Internet-Draft, draft-ietf-ace-oscore-profile-13, 27 October 2020, <http://www.ietf.org/internet-drafts/draft-ietf-ace-oscore-profile-13.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-35, 24 June 2020, <http://www.ietf.org/internet-drafts/draft-ietf-ace-oauth-authz-35.txt>.

[I-D.ietf-core-oscore-groupcomm]
Tiloca, M., Selander, G., Palombini, F., and J. Park, "Group OSCORE - Secure Group Communication for CoAP", Work in Progress, Internet-Draft, draft-ietf-core-oscore-groupcomm-09, 23 June 2020, <http://www.ietf.org/internet-drafts/draft-ietf-core-oscore-groupcomm-09.txt>.

[I-D.ietf-cose-x509] Schaad, J., "CBOR Object Signing and Encryption (COSE): Header parameters for carrying and referencing X.509 certificates", Work in Progress, Internet-Draft, draft-ietf-cose-x509-07, 17 September 2020, <http://www.ietf.org/internet-drafts/draft-ietf-cose-x509-07.txt>.

[I-D.mattsson-cose-cbor-cert-compress]
Raza, S., Hoglund, J., Selander, G., Mattsson, J., and M. Furuhed, "CBOR Profile of X.509 Certificates", Work in Progress, Internet-Draft, draft-mattsson-cose-cbor-cert-compress-01, 13 July 2020, <http://www.ietf.org/internet-drafts/draft-mattsson-cose-cbor-cert-compress-01.txt>.

[I-D.palombini-core-oscore-edhoc]
Palombini, F., Tiloca, M., Hoeglund, R., Hristozov, S., and G. Selander, "Combining EDHOC and OSCORE", Work in Progress, Internet-Draft, draft-palombini-core-oscore-edhoc-00, 13 July 2020, <http://www.ietf.org/internet-drafts/draft-palombini-core-oscore-edhoc-00.txt>.

Appendix A. Other Authentication Methods

In order to protect certificate enrollment with OSCORE, the necessary keying material (notably, the OSCORE Master Secret, see [RFC8613]) needs to be established between EST-oscore client and EST-oscore server. In this appendix we analyse alternatives to EDHOC, which was assumed in the body of this specification.

A.1. TTP Assisted Authentication

Trusted third party (TTP) based provisioning, such as the OSCORE profile of ACE [I-D.ietf-ace-oscore-profile] assumes existing security associations between the client and the TTP, and between

the server and the TTP. This setup allows for reduced message overhead and round trips compared to the full-fledged EDHOC key exchange. Following the ACE terminology the TTP plays the role of the Authorization Server (AS), the EST-oscore client corresponds to the ACE client and the EST-oscore server is the ACE Resource Server (RS).

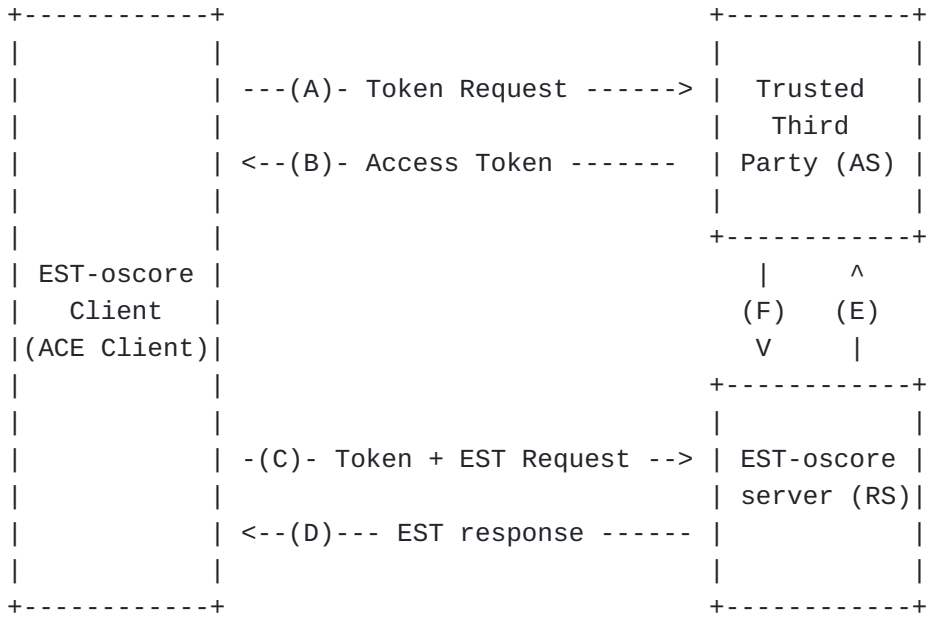

Figure 3: Accessing EST services using a TTP for authenticated key establishment and authorization.

During initial enrollment the EST-oscore client uses its existing security association with the TTP, which replaces the Implicit TA database, to establish an authenticated secure channel. The [I-D.ietf-ace-oscore-profile] ACE profile RECOMMENDS the use of OSCORE between client and TTP (AS), but TLS or DTLS MAY be used additionally or instead. The client requests an access token at the TTP corresponding the EST service it wants to access. If the client request was invalid, or not authorized according to the local EST policy, the AS returns an error response as described in Section 5.6.3 of [I-D.ietf-ace-oauth-authz]. In its responses the TTP (AS) SHOULD signal that the use of OSCORE is REQUIRED for a specific access token as indicated in section 4.3 of [I-D.ietf-ace-oscore-profile]. This means that the EST-oscore client MUST use OSCORE towards all EST-oscore servers (RS) for which this access token is valid, and follow Section 4.3 in [I-D.ietf-ace-oscore-profile] to derive the security context to run OSCORE. The ACE OSCORE profile RECOMMENDS the use of CBOR web token (CWT) as specified in [RFC8392]. The TTP (AS) MUST also provision an OSCORE security

context to the EST-oscore client and EST-oscore server (RS), which is then used to secure the subsequent messages between the client and the server. The details on how to transfer the OSCORE contexts are described in section 3.2 of [I-D.ietf-ace-oscore-profile].

Once the client has retrieved the access token it follows the steps in [I-D.ietf-ace-oscore-profile] to install the OSCORE security context and presents the token to the EST-oscore server. The EST-oscore server installs the corresponding OSCORE context and can either verify the validity of the token locally or request a token introspection at the TTP. In either case EST policy decisions, e.g., which client can request enrollment or reenrollment, can be implemented at the TTP. Finally the EST-oscore client receives a response from the EST-oscore server.

A.2. PSK Based Authentication

Another method to bootstrap EST services requires a pre-shared OSCORE security context between the EST-oscore client and EST-oscore server. Authentication using the Implicit TA is no longer required since the shared security context authenticates both parties. The EST-oscore client and EST-oscore server need access to the same OSCORE Master Secret as well as the OSCORE identifiers (Sender ID and Recipient ID) from which an OSCORE security context can be derived, see [RFC8613]. Some optional parameters may be provisioned if different from the default value:

* an ID context distinguishing between different OSCORE security contexts to use,
* an AEAD algorithm,
* an HKDF algorithm,
* a master salt, and
* a replay window size.

Appendix B. CBOR Encoding of EST Payloads

Current EST based specifications transport messages using the ASN.1 data type declaration. It would be favorable to use a more compact representation better suitable for constrained device implementations. In this appendix we list CBOR encodings of requests and responses of the mandatory EST functions (see Section 4.3).

B.1. Distribution of CA Certificates (/crts)

The EST client can request a copy of the current CA certificates. In EST-coaps and EST-oscore this is done using a GET request to /crts

(with empty payload). The response contains a chain of certificates used to establish an Explicit Trust Anchor database for subsequent authentication of the EST server.

CBOR encoding of X.509 certificates is specified in [I-D.mattsson-cose-cbor-cert-compress]. CBOR encoding of certificate chains is specified below. This allows for certificates encoded using the CBOR certificate format, or as binary X.509 data wrapped as a CBOR byte string.

CDDL:

```
certificate chain = (
   + certificate : bstr
)
certificate = x509_certificate / cbor_certificate
```

B.2. Enrollment/Re-enrollment of Clients (/sen, /sren)

Existing EST standards specify the enrollment request to be a PKCS#10 formated message [RFC2986]. The essential information fields for the CA to verify are the following:

- Information about the subject, here condensed to the subject common name,
- subject public key, and
- signature made by the subject private key.

CDDL:

```
certificate request = (
   subject_common_name : bstr,
   public_key : bstr
   signature : bstr,
   ? ( signature_alg : int, public_key_info : int )
)
```

The response to the enrollment request is the subject certificate, for which CBOR encoding is specified in [I-D.mattsson-cose-cbor-cert-compress].

The same message content in request and response applies to re-enrollment.

TODO: PKCS#10 allows inclusion of attributes, which can be used to specify extension requests, see Section 5.4.2 of [RFC2985]. CBOR encoding of the challengePassword attribute needs to be defined (see Section 3.3). What other attributes are relevant?

B.2.1. CBOR Certificate Request Examples

Here is an example of CBOR encoding of certificate request as defined in the previous section.

114 bytes:

```
( h'0123456789ABCDF0',
h'61eb80d2abf7d7e4139c86b87e42466f1b4220d3f7ff9d6a1ae298fb9adbb464',
h'30440220064348b9e52ee0da9f9884d8dd41248c49804ab923330e208a168172dc
ae1
27a02206a06c05957f1db8c4e207437b9ab7739cb857aa6dd9486627b8961606a2b68ae' )
```

In the example above the signature is generated on an ASN.1 data structure. To validate this, the receiver needs to reconstruct the original data structure. Alternatively, in native mode, the signature is generated on the profiled data structure, in which case the overall overhead is further reduced.

B.2.2. ASN.1 Certificate Request Examples

A corresponding certificate request of the previous section using ASN.1 is shown in Figure 4.

```
SEQUENCE {
   SEQUENCE {
     INTEGER 0
     SEQUENCE {
       SET {
         SEQUENCE {
           OBJECT IDENTIFIER commonName (2 5 4 3)
           UTF8String '01-23-45-67-89-AB-CD-F0'
           }
         }
       }
     SEQUENCE {
       SEQUENCE {
         OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
         OBJECT IDENTIFIER prime256v1 (1 2 840 10045 3 1 7)
         }
       BIT STRING
         (65 byte public key)
       }
   SEQUENCE {
     OBJECT IDENTIFIER ecdsaWithSHA256 (1 2 840 10045 4 3 2)
     }
   BIT STRING
     (64 byte signature)
```

Figure 4: ASN.1 Structure.

In Base64, 375 bytes:

```
-----BEGIN CERTIFICATE REQUEST-----
MIHcMIGEAgEAMCIxIDAeBgNVBAMMFzAxLTIzLTQ1LTY3LTg5LUFCLUNELUYwMFkw
EwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEYeuA0qv31+QTnIa4fkJGbxtCINP3/51q
GuKY+5rbtGSeZn3l8rVbU0jVEBWvKhAd98JeqgsuauGHRNWt2FqJ1aAAMAoGCCqG
SM49BAMCA0cAMEQCIAZDSLnlLuDan5iE2N1BJIxJgEq5IzMOIIoWgXLcrhJ6AiBq
BsBZV/HbjE4gdDe5q3c5y4V6pt2UhmJ7iWFgaitorg==
-----END CERTIFICATE REQUEST-----
```

In hex, 221 bytes:

```
3081dc30818402010030223120301e06035504030c1730312d32332d34352d36
372d38392d41422d43442d46303059301306072a8648ce3d020106082a8648ce
3d03010703420004461eb80d2abf7d7e4139c86b87e42466f1b4220d3f7ff9d6a
1ae298fb9adbb4649e667de5f2b55b5348d51015af2a101df7c25eaa0b2e6ae1
8744d5add85a89d5a000300a06082a8648ce3d040302034700304402200643 48
b9e52ee0da9f9884d8dd41248c49804ab923330e208a168172dcae127a02206a
06c05957f1db8c4e207437b9ab7739cb857aa6dd9486627b8961606a2b68ae
```

Authors' Addresses

Goeran Selander
Ericsson AB

Email: goran.selander@ericsson.com

Shahid Raza
RISE

Email: shahid.raza@ri.se

Martin Furuhed
Nexus

Email: martin.furuhed@nexusgroup.com

Malisa Vucinic
INRIA

Email: malisa.vucinic@inria.fr

Timothy Claeys
INRIA

Email: timothy.claeys@inria.fr