Protecting EST Payloads with OSCORE

Göran Selander, Shahid Raza, Martin Furuhed, Mališa Vučinić, Timothy Claeys

Workgroup: ACE Working Group
Internet-Draft:
draft-ietf-ace-coap-est-oscore-06
Published: 21 October 2024
Intended Status: Standards Track
Expires: 24 April 2025
Authors: G. Selander (Ericsson AB)
S. Raza (RISE)
M. Furuhed (Nexus)
M. Vučinić (Inria)
T. Claeys

Protecting EST Payloads with OSCORE

Abstract

Enrollment over Secure Transport (EST) is a certificate provisioning protocol over HTTPS [RFC7030] or CoAPs [RFC9148]. This document specifies how to carry EST over the Constrained Application Protocol (CoAP) protected with Object Security for Constrained RESTful Environments (OSCORE). The specification builds on the EST-coaps [RFC9148] specification, but uses OSCORE and Ephemeral Diffie-Hellman over COSE (EDHOC) instead of DTLS. The specification also leverages the certificate structures defined in [I-D.ietf-cose-cbor-encoded-cert], which can be optionally used alongside X.509 certificates.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Authentication and Authorization for Constrained Environments Working Group mailing list (ace@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/ace/.

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

Status of This Memo

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 [RFC9148], which defines a version of Enrollment over Secure Transport [RFC7030] for transporting EST payloads over CoAP [RFC7252] and DTLS [RFC9147], instead of HTTP and TLS [RFC8446].

This document describes a method for protecting EST payloads over CoAP or HTTP with OSCORE [RFC8613]. OSCORE specifies an extension to CoAP which protects messages at the application layer 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 the underlying transport and through proxies translating between CoAP and HTTP.

OSCORE is designed for constrained environments, building on IoT standards such as CoAP, CBOR [RFC8949], and COSE [RFC9052] [RFC9053], and has in particular gained traction in settings where message sizes and the number of exchanged messages need to be kept at a minimum, such as 6TiSCH [RFC9031], or for securing CoAP group 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.

Prior to running EST-oscore, the protocol defined in this specification, there must exist a trust relation between the EST-oscore client and the EST-oscore server. This trust relation may be based on the pre-shared OSCORE security context, or based on the common root of trust. In case there is a pre-shared OSCORE security context, the CoAP exchange carrying EST payloads can occur immediately. In case there is a common root of trust, a security handshake based on the Ephemeral Diffie-Hellman over COSE (EDHOC, [RFC9528]) protocol needs to occur prior to running CoAP. How this trust relation is established is out of scope of this document.

How the EST-oscore server verifies the identity of the client prior to issuing a certificate is also out of scope of this specification.

EST-oscore defines a number of optimizations with respect to EST-coaps. The performance of certificate enrollment and certificate-based authentication described in this document includes the use of:

*Compact CBOR representations of X.509 certificates (see [I-D.ietf-cose-cbor-encoded-cert])

*Certificates by reference (see [RFC9360])

*Compact, CBOR representations of EST payloads (see [I-D.ietf-cose-cbor-encoded-cert])

1.1. Operational Differences with EST-coaps

The protection of EST payloads defined in this document builds on EST-coaps [RFC9148] 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 by, or complemented with, OSCORE.

* The DTLS handshake is replaced by, or complemented with, the lightweight authenticated key exchange protocol EDHOC [RFC9528], 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.

* 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 the EST server as described in Section 5 of [RFC9148] is therefore not applicable.

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. 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.

2. 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.

This document uses terminology from [RFC9148] 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."

Apart from enrolling signature keys, this document also specifies how to enroll static DH keys. Instead of signing, possession of the private static DH key may be proved by generating a MAC given the recipient's public DH key. Therefore this document extends the definition of the term "Trust Anchor": the corresponding public key can also be used for MAC generation for static DH proof of possession procedures.

3. Authentication

This specification replaces, or complements, the DTLS handshake in EST-coaps with the lightweight authenticated key exchange protocol EDHOC [RFC9528]. During initial enrollment, the EST-oscore client and server run EDHOC [RFC9528] to authenticate and establish the OSCORE Security Context used to protect the messages carrying EST payloads.

The EST-oscore client MUST play the role of the EDHOC Initiator. The EST-oscore server MUST play the role of the EDHOC Responder.

The 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. These requirements are fullfilled when using EDHOC [RFC9528].

The server must also provide relevant information to the CA to support its decision about issuing a certificate.

3.1. EDHOC

EDHOC supports authentication with certificates or 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 of the credential (e.g., x5u, see [RFC9360]) 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 the EST client and server. 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.ietf-cose-cbor-encoded-cert].

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 a signature key, a proof-of-possession is generated by the client when it signs the PKCS#10 Request during the enrollment phase. In case of a static DH key, a proof-of-possession is generated by the client when it generates a MAC and includes it in the PKCS#10 request, as per Section 4.8.

Connection-based channel binding refers to the security binding between the PKCS#10 object and the underlying secure transport layer. This is typically achieved by including the challengePassword attribute in the PKCS#10 object that is dependent on the underlying security session. Connection-based proof-of-possession using the challengePassword attribute of the PKCS#10 object is OPTIONAL, see Section 6.

3.4. Optimizations

*The third 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 enrollment request and response) in two round trips [I-D.ietf-core-oscore-edhoc].

*The enrolled certificates MAY be the CBOR-encoded certificates defined in [I-D.ietf-cose-cbor-encoded-cert].

*The enrolled client certificate MAY be referenced instead of transported [RFC9360]. 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 specify a reference to the enrolled certificate rather than the certificate itself. The EST-oscore server MAY in the enrollment response to the EST-oscore client include a pointer to a directory or database where the certificate can be retrieved.

*The PKCS#10 object MAY request a certificate for a static DH key instead of a signature key. This may result in a more compact request because the use of static DH keys may imply a proof-of-

posession using a MAC, which is shorter than a signature. Additionally, subsequent EDHOC sessions using static DH keys for authentication have less overhead than key exchange protocols using signature-based authentication credentials.

4. Protocol Design and Layering

EST-oscore uses CoAP [RFC7252] and Block-Wise transfer [RFC7959] to transfer EST messages in the same way as [RFC9148]. Instead of DTLS record layer, OSCORE [RFC8613] is used to protect the messages conveying the EST payloads. External Authorization Data (EAD) fields of EDHOC are intentionally not used to carry EST payloads because EDHOC needs not be executed in the case of re-enrollment. The DTLS handshake is complemented by or replaced with EDHOC [RFC9528]. Figure 1 below shows the layered EST-oscore architecture. Note that Figure 1 does not illustrate the potential use of DTLS. Protocol design also allows that OSCORE and EDHOC messages are carried within the same CoAP message, as per [I-D.ietf-core-oscore-edhoc].

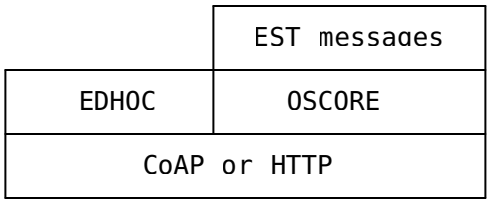

Figure 1: The stack diagram of EST protected with OSCORE.

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 4.1 of [RFC9148], as well as the new Resource Type defined in Section 8.2 of [RFC9148] apply to EST-oscore. Support for OSCORE is indicated by the "osc" attribute defined in Section 9 of [RFC8613].

Example:

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

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

The use of the "osc" attribute is REQUIRED. In scenarios where OSCORE and DTLS are combined, the absence of the "osc" attribute might wrongly suggest that the EST server is actually using EST-coaps, because of the scheme "coaps", when it is using EST-oscore.

4.2. 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 4.2 in [RFC9148] 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.

4.2.1. /crts

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.

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.ietf-cose-cbor-encoded-cert].

4.3. Payload formats

Similar to EST-coaps, EST-oscore allows transport of DER-encoded objects of a given Media-Type. When transporting DER-encoded objects, EST-oscore uses the same CoAP Content-Format identifiers as EST-coaps when transferring EST requests and responses. In addition, EST-oscore allows the transport of CBOR-encoded objects, signaled via their corresponding Media-Type.

EST-oscore servers MUST support both the DER-encoded ASN.1 objects and the CBOR-encoded objects. This means supporting formats detailed in Section 4.3.1 and Section 4.3.2. It is up to the client to support only DER-encoded ASN.1, CBOR encoding, or both. As a reminder, Content-Format negotiation happens through CoAP's Accept option present in the requests.

4.3.1. DER-encoded ASN.1 Objects

Table 2 summarizes the information from Section 4.3 in [RFC9148] in what concerns the transport of DER-encoded ASN.1 objects.

URI	Media Type	Type	#IANA
/crts	N/A	req	-
	application/pkix-cert	res	287
	application/pkcs7-mime;smime-type=certs-only	res	281
/sen	application/pkcs10	req	286
	application/pkix-cert	res	287
	application/pkcs7-mime;smime-type=certs-only	res	281
/sren	application/pkcs10	req	286
	application/pkix-cert	res	287
	application/pkcs7-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 the associated ASN.1 CoAP Content-Format identifiers.

Content-Format 281 and Content-Format 287 MUST be supported by EST-oscore servers. It is up to the client to support only Content-Format 281, 287 or both. As indicated in Section 4.3 of [RFC9148], the client will use a CoAP Accept Option in the request to express the preferred response Content-Format. If an Accept Option is not included in the request, the client is not expressing any preference and the server SHOULD choose format 281.

The generated response for /skg and /skc requests contains two parts: certificate and the corresponding private key. Section 4.8 of [RFC9148] specifies that the private key in response to /skc request may be either an encrypted (PKCS #7) or unencrypted (PKCS #8) key, depending on whether the CSR request included SMIMECapabilities.

Due to the use of OSCORE, which protects the communication between the EST client and the EST server end-to-end, it is possible to return the private key to /skc or /skg as an unencrypted PKCS #8 object (Content-Format identifier 284). Therefore, when making the CSR to /skc or /skg, the EST client MUST NOT include SMIMECapabilities. As a consequence, the private key part of the response to /skc or /skg is an unencrypted PKCS #8 object.

Function	DER-encoded ASN.1 Response, Part 1	DER-encoded ASN.1 Response, Part 2
/skg	284	281
/skc	284	287

Table 3: Response Content-Format identifiers for /skg and /skc in case of DER-encoded ASN.1 objects.

4.3.2. CBOR-encoded Objects

Table 4 presents the equivalent information to Section 4.3.1 when CBOR-encoded objects are in use.

URI	Media Type	Type	#IANA
/crts	N/A	req	-
	application/cose-c509-cert	res	TBD6
/sen	application/cose-c509-pkcs10	req	TBD7
	application/cose-c509-cert	res	TBD6
/sren	application/cose-c509-pkcs10	req	TBD7
	application/cose-c509-cert	res	TBD6
/skg	application/cose-c509-pkcs10	req	TBD7
	application/multipart-core	res	62
/skc	N/A	req	-
	N/A	res	-
/att	N/A	req	-
	application/csrattrs	res	TBD5

Table 4: EST functions and the associated CBOR CoAP Content-Format identifiers.

In case of CBOR-encoded objects, there is a single Content-Format, TBD6, that MUST be supported by both the EST-oscore servers and clients.

EDITOR NOTE: Specify the CDDL structure of /csrattrs and point to appropriate document for its semantics.

In the case of CBOR-encoded request to /skg, the two parts of the response are also CBOR encoded. The certificate part is encoded as the application/cose-c509-cert object (Content-Format identifier TBD6), while the corresponding private key is encoded as application/cose-c509-privkey (Content-Format identifier TBD10). The function /skc is not available when using CBOR-encoded objects, and for server-side generated keys, clients MUST use the /skg function.

Table 5 summarizes the Content-Format identifiers used in responses to the /skg function.

Function	CBOR Response, Part 1	CBOR Response Part 2
/skg	101	TBD6

Table 5: Response Content-Format identifiers for /skg in case of CBOR-encoded objects.

4.4. Message Bindings

Note that the EST-oscore message characteristics are identical to those specified in Section 4.4 of [RFC9148]. It is therefore required 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. In case DTLS is additionally used, the translation target is the scheme "coaps", instead of "coap".

4.5. CoAP response codes

See Section 4.5 in [RFC9148].

4.6. 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 [RFC9528]). Together with various measures listed in this document such as CBOR-encoded payloads [RFC8949], CBOR certificates [I-D.ietf-cose-cbor-encoded-cert], certificates by reference (Section 3.4), and trust anchors without signature (Section 4.2.1), a significant reduction of message sizes can be achieved.

Nevertheless, depending on the application, the protocol messages may become larger than the available frame size thus 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 6LoWPAN fragmentation, since it involves an error-prone datagram reassembly. To limit the size of the CoAP payload, this document specifies the requirements on implementing CoAP options Block1 and Block2. EST-oscore servers MUST implement Block1 and Block2. EST-oscore clients MUST implement Block2 and MAY implement Block1.

4.7. Delayed Responses

See Section 4.7 in [RFC9148].

4.8. Enrollment of Static DH Keys

This section specifies how the EST client enrolls a static DH key. In general, a given key pair should only be used for a single purpose, such as key establishment, digital signature, key transport.

The EST client attempting to enroll a DH key for a key usage operation other than digital signature SHOULD use an alternative proof-of-possession algorithm: The EST client SHOULD prepare the PKCS#10 object and compute a MAC, replacing the signature, over the certification request information by following the steps in Section 6 of [RFC6955]. The Key Derivation Function (KDF) and the MAC MUST be set to the HDKF and HMAC algorithms used by OSCORE. The KDF and MAC is thus defined by the hash algorithm used by OSCORE in HKDF and HMAC, which by default is SHA-256. When EDHOC is used, then the hash algorithm is the application hash algorithm of the selected cipher suite.

To generate a MAC according to the algorithm outlined in Section 6 of [RFC6955], the client needs to know the public DH key of the proof-of-possession recipient/verifier, i.e. the EST server. In the general case, the EST client MAY obtain the CA certs including the CA's DH certificate using the /crts function using an explicit request/response flow. The obtained certificate indicates the DH group parameters which MUST be respected by the EST client when generating its own DH key pair.

As an optimization, when EDHOC precedes the enrollment and combined OSCORE-EDHOC flow is being used in EDHOC message_3 and message_4 per [I-D.ietf-core-oscore-edhoc], the client MUST use the public ephemeral key of the EDHOC Responder, G_Y, as the recipient public key in the algorithm outlined in Section 6 of [RFC6955]. When generating its DH key pair, the client uses the group parameters as indicated by the EDHOC cipher suite in use in the EDHOC session. Because the combined delivery is used per [I-D.ietf-core-oscore-edhoc], the client has already in EDHOC message_2 obtained the ephemeral key G_Y of the server.

In some cases, it may be beneficial to exceptionally use the static DH private key associated to the public key used in enrollment for a one-time signing operation of the CSR. While a key pair should only be used for a single purpose (e.g. key establishment or signing), this exceptional use for one-time signing of the CSR is allowed, as discussed in Section 5.6.3.2 of [SP-800-56A] and Section 5.2 of [SP-800-57].

5. HTTP-CoAP Proxy

As noted in Section 5 of [RFC9148], 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 messages conveying the EST payloads can be protected end-to-end between the EST client and EST server, irrespective 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 5 of [RFC9148] is not applicable.

The mappings between CoAP and HTTP referred to in Section 8.1 of [RFC9148] 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 additional use of TLS and DTLS is optional. If a secure association is needed between the EST Client and the CoAP-to-HTTP Proxy, this may also rely on OSCORE [I-D.tiloca-core-oscore-capable-proxies].

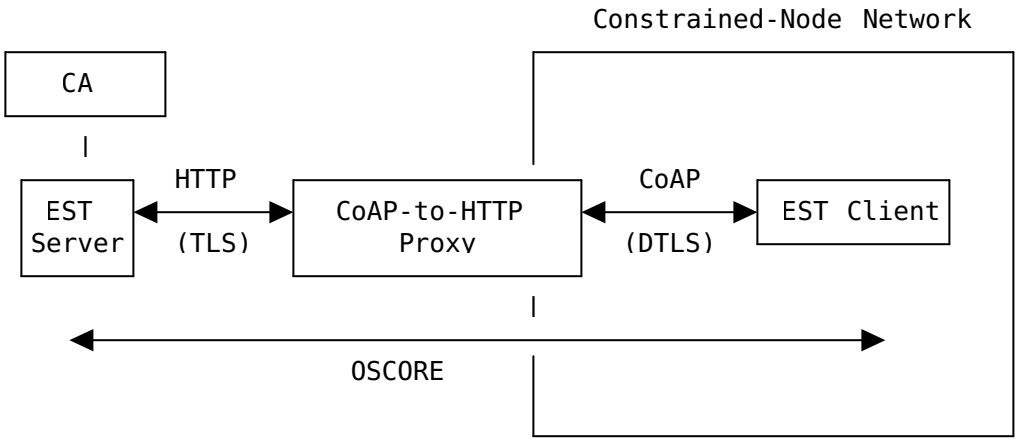

Figure 2: CoAP-to-HTTP proxy at the CoAP boundary.

6. Security Considerations

6.1. Server-generated Private Keys

This document enables the EST client to request generation of private keys and the enrollment of the corresponding public key through /skg and /skc functions. As discussed in Section 9 of [RFC9148], the transport of private keys generated at EST-server is inherently risky. The use of server-generated private keys may lead to the increased probability of digital identity theft. Therefore, implementations SHOULD NOT use server-generated private key EST functions.

A cryptographically secure pseudo-random number generator is required to be available to generate good quality private keys on EST-clients.

A cryptographically secure pseudo-random number generator is also a dependency of many security protocols. This includes the EDHOC protocol, which EST-oscore uses for the mutual authentication of EST-client and EST-server. If EDHOC is used and a secure pseudo-random number generator is available, the EST-client MUST NOT use server-generated private key EST functions. However, EST-oscore also allows pre-shared OSCORE contexts to be used for authentication, meaning that EDHOC may not necessarily be required in the protocol stack of an EST-client. If EDHOC is not used for authentication, and the EST-client device does not have a cryptographically secure pseudo-random number generator, then the EST-client MAY use the server-generated private key functions.

Although hardware random number generators are becoming dominantly present in modern IoT devices, it has been shown that many available hardware modules contain vulnerabilities and do not produce cryptographically secure random numbers. It is therefore important to use multiple randomness sources to seed the cryptographically secure pseudo-random number generator.

6.2. Considerations on Channel Binding

Section 3 of [RFC9148] specifies that the use of connection-based channel binding is optional, and achieves it by including the tls-unique value in the CSR. This specification when used with EDHOC for the enrollment of static DH keys achieves connection-based channel binding by using the EDHOC ephemeral key of the responder as the public key in the proof-of-possession algorithm which generates a PKCS#10 MAC. Therefore, connection-based channel binding is in this case achieved without any additional overhead.

Other cases include pre-shared OSCORE contexts and the case where the signature key used for signing the CSR is different from the key used in the EDHOC run. In these other cases, this specification makes explicit channel binding based on the challengePassword attribute in PKCS#10 requests OPTIONAL. For example, the challengePassword attribute could be used for freshness in the case of pre-shared OSCORE contexts and a re-enrollment request.

How challengePassword is generated is outside of the scope of this specification and can be specified by an application profile.

7. IANA Considerations

This document does not require any IANA registrations.

8. References

8.1. Normative References

[I-D.ietf-core-oscore-edhoc] Palombini, F., Tiloca, M., Höglund, R., Hristozov, S., and G. Selander, "Using Ephemeral Diffie-Hellman Over COSE (EDHOC) with the Constrained Application

Protocol (CoAP) and Object Security for Constrained RESTful Environments (OSCORE)", Work in Progress, Internet-Draft, draft-ietf-core-oscore-edhoc-11, 9 April 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-core-oscore-edhoc-11>.

[I-D.ietf-cose-cbor-encoded-cert] Mattsson, J. P., Selander, G., Raza, S., Höglund, J., and M. Furuhed, "CBOR Encoded X.509 Certificates (C509 Certificates)", Work in Progress, Internet-Draft, draft-ietf-cose-cbor-encoded-cert-11, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-cbor-encoded-cert-11>.

[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>.

[RFC6955] Schaad, J. and H. Prafullchandra, "Diffie-Hellman Proof-of-Possession Algorithms", RFC 6955, DOI 10.17487/RFC6955, May 2013, <https://www.rfc-editor.org/info/rfc6955>.

[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>.

[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>.

[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>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/

RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/info/rfc9052>.

[RFC9053] Schaad, J., "CBOR Object Signing and Encryption (COSE): Initial Algorithms", RFC 9053, DOI 10.17487/RFC9053, August 2022, <https://www.rfc-editor.org/info/rfc9053>.

[RFC9147] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022, <https://www.rfc-editor.org/info/rfc9147>.

[RFC9148] van der Stok, P., Kampanakis, P., Richardson, M., and S. Raza, "EST-coaps: Enrollment over Secure Transport with the Secure Constrained Application Protocol", RFC 9148, DOI 10.17487/RFC9148, April 2022, <https://www.rfc-editor.org/info/rfc9148>.

[RFC9360] Schaad, J., "CBOR Object Signing and Encryption (COSE): Header Parameters for Carrying and Referencing X.509 Certificates", RFC 9360, DOI 10.17487/RFC9360, February 2023, <https://www.rfc-editor.org/info/rfc9360>.

[RFC9528] Selander, G., Preuß Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", RFC 9528, DOI 10.17487/RFC9528, March 2024, <https://www.rfc-editor.org/info/rfc9528>.

8.2. Informative References

[I-D.ietf-core-oscore-groupcomm] Tiloca, M., Selander, G., Palombini, F., Mattsson, J. P., and R. Höglund, "Group Object Security for Constrained RESTful Environments (Group OSCORE)", Work in Progress, Internet-Draft, draft-ietf-core-oscore-groupcomm-23, 26 September 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-core-oscore-groupcomm-23>.

[I-D.tiloca-core-oscore-capable-proxies] Tiloca, M. and R. Höglund, "OSCORE-capable Proxies", Work in Progress, Internet-Draft, draft-tiloca-core-oscore-capable-proxies-07, 10 July 2023, <https://datatracker.ietf.org/doc/html/draft-tiloca-core-oscore-capable-proxies-07>.

[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>.

[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>.

[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>.

[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>.

[RFC9031] Vučinić, M., Ed., Simon, J., Pister, K., and M. Richardson, "Constrained Join Protocol (CoJP) for 6TiSCH", RFC 9031, DOI 10.17487/RFC9031, May 2021, <https://www.rfc-editor.org/info/rfc9031>.

[SP-800-56A] Barker, E., Chen, L., Roginsky, A., Vassilev, A., and R. Davis, "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography", NIST Special Publication 800-56A Revision 3, April 2018, <https://doi.org/10.6028/NIST.SP.800-56Ar3>.

[SP-800-57] Barker, E., "Recommendation for Key Management", NIST Special Publication 800-57 Revision 5, May 2020, <https://doi.org/10.6028/NIST.SP.800-57pt1r5>.

Appendix A. Example Enrollment With Optimizations

The message flow starts with the EST client sending EDHOC message_1. The EDHOC handshake follows and concludes with the EDHOC message_3. EDHOC message_3 is carried in the same message as the OSCORE enrollment request, as specified in [I-D.ietf-core-oscore-edhoc]. The OSCORE enrollment request contains a CoAP POST to the /sen endpoint. This POST request includes the Content-Format option set to the value application/cose-c509-pkcs10, and the Accept option set to the value application/cose-c509-cert, indicating the support for CBOR-encoded objects. In response, the client receives the application/cose-c509-cert object which contains the certificate.

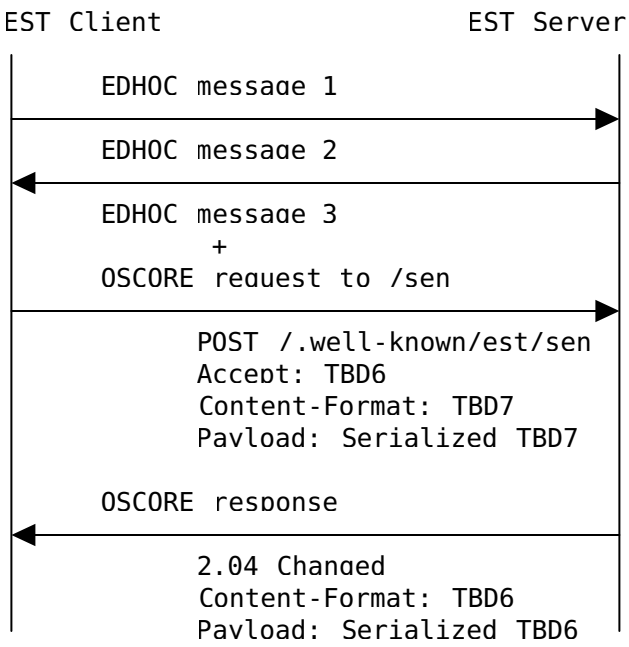

Figure 3: Enrollment EST-oscore flow with optimizations.

Acknowledgments

The authors would like to thank Marco Tiloca, and John Mattsson for providing a review of the document.

Authors' Addresses

Göran Selander
Ericsson AB

Email: goran.selander@ericsson.com

Shahid Raza
RISE

Email: shahid.raza@ri.se

Martin Furuhed
Nexus

Email: martin.furuhed@nexusgroup.com

Mališa Vučinić
Inria

Email: malisa.vucinic@inria.fr

Timothy Claeys

Email: timothy.claeys@gmail.com