ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2026
G. Selander
Ericsson AB
S. Raza
RISE
M. Furuhed
Nexus
M. Vuini
Inria
T. Claeys
2 March 2026

Protecting EST Payloads with OSCORE
draft-ietf-ace-coap-est-oscore-10

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

Copyright Notice

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 [RFC9110][RFC9112] and TLS [RFC8446].

This document describes a method for protecting EST payloads over CoAP with OSCORE [RFC8613]. OSCORE specifies an extension to CoAP that 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 (see Section 11 of [RFC8613]). 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 relationship between the EST-oscore client and the EST-oscore server. This trust relationship may be based on the pre-shared OSCORE security context, or 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 EST-oscore. How this trust relationship is established is out of the scope of this document.

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

EST-oscore defines a number of optimizations with respect to EST-coaps:

* The DTLS record layer is replaced by OSCORE.

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

* Compact CBOR representations of X.509 certificates and EST payloads (see [I-D.ietf-cose-cbor-encoded-cert]) are optionally used.

* Certificates by reference (see [RFC9360]) are optionally used.

* The EST payloads protected by OSCORE can be proxied between constrained networks supporting CoAP and non-constrained networks supporting HTTP/HTTPs, through a CoAP-HTTP proxy without any security processing at the proxy (see Section 5). The concept "Registrar" and its required trust relationship with the EST server as described in Section 5 of [RFC9148] is therefore not applicable.

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 certificates with keys that are used for signing, this document also specifies how to enroll certificates with keys that are used for Diffie-Hellman (DH) operations (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 the DTLS handshake in EST-coaps with the lightweight authenticated key exchange protocol EDHOC [RFC9528]. The enrollment using EST-oscore is based on the existence of an OSCORE Security Context protecting the messages conveying the EST payloads. This Security Context is typically established through an EDHOC session preceding the initial enrollment. Re-enrollment does not require a new EDHOC session.

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 perform mutual authentication. The EST server and EST client are responsible for ensuring that an acceptable cipher suite is negotiated. The client authenticates the server before accepting any server response. The server authenticates the client. These requirements are fulfilled 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., c5u, see [I-D.ietf-cose-cbor-encoded-cert]) 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. The requirements on managing the Implicit and Explicit TA databases are discussed in Section 9.1 of [RFC9148] and Section 9.2 of [RFC9148] and apply to EST-oscore.

The EST client and EST server 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]. The EST client indicates its preference for the type of the certificate it supports through the CoAP Accept option included in the request to the EST server.

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

This section contains optional behavior that may be used to reduce message sizes or round trips based on the application configuration.

* The third message of the EDHOC protocol, message_3, MAY be combined with an OSCORE-protected request [RFC9668], enabling authenticated Diffie-Hellman key exchange and a protected CoAP request/response (which may contain an enrollment request and response) in two round trips [RFC9668].

* The enrolled client certificate 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 response to the PKCS#10 request MAY specify a reference to the enrolled certificate rather than the certificate itself (see Section 4.9).

* 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-possession 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.

* When the EDHOC handshake precedes the enrollment request, it is RECOMMENDED for the EST-client to leverage the information from the EDHOC session on the selected cipher suite when making a decision on which type of credential to enroll.

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 the DTLS record layer, OSCORE [RFC8613] is used to protect the messages conveying the EST payloads. External Authorization Data (EAD) fields of EDHOC messages are intentionally not used to carry EST payloads because EDHOC needs not be executed in the case of re-enrollment. The DTLS handshake is replaced with EDHOC [RFC9528]. Figure 1 below shows the layered EST-oscore architecture. Protocol design also allows that OSCORE and EDHOC messages are carried within the same CoAP message, as per [RFC9668].

```
             +----------------+
             |  EST messages  |
+------------+----------------+
|   EDHOC    |    OSCORE      |
+------------+----------------+
|        CoAP or HTTP         |
+-----------------------------+
```

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

EST-oscore follows much of the EST-coaps and EST design. This includes the need to authenticate the EST-server before performing any request on the different EST endpoints specified in this document.

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. In a web link targeting a resource for EST-oscore, it is REQUIRED to indicate that the resource is only accessible using OSCORE, by means of the "osc" target attribute defined in Section 9 of [RFC8613].

Example:

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

RES: 2.05 Content
  Content-Format: 40 (application/link-format)
  Payload:
    </est>;rt=ace.est.sen;osc
```

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-oscore provides the /crts operation. A successful request from the client to this resource will be answered with one or more certificates, subsequently installed in the TA database (once the server is authenticated), resulting in Explicit TAs.

A trust anchor is commonly a self-signed certificate of the CA public key, of the format indicated by the CoAP Accept option present in the request. In order to reduce transport overhead, the trust anchor could be a CBOR encoding of an X.509 certificate [I-D.ietf-cose-cbor-encoded-cert], or a CWT Claims Set (CCS) [RFC8392], containing the CA public key and associated data without a signature.

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, as indicated by their corresponding Media-Type.

EST-oscore servers MUST support both the DER-encoded ASN.1 objects and the CBOR-encoded objects, i.e., they MUST support 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, only CBOR encoding, or both. Based on the client encoding of the CSR (DER encoding or CBOR encoding), the server is able to tell whether the client prefers a DER-encoded object (Section 4.3.1) or a CBOR-encoded object (Section 4.3.2) in response. In addition, Content-Format negotiation for specific objects happens through the CoAP Accept option present in the requests. The CoAP Accept option may not be present; this is a case which carries special semantics, see Section 4.3.1 and Section 4.3.2.

4.3.1. DER-encoded ASN.1 Objects

Table 2 summarizes the information from Section 4.3 in [RFC9148] for 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, only Content-Format 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. An exception to this "SHOULD" is in the case when the request contains a CBOR-encoded object (e.g. application/cose-c509-pkcs10), when the server SHOULD respond with a CBOR-encoded object (see Section 4.3.2).

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	TBD3
/sen	application/cose-c509-pkcs10	req	TBD4
	application/cose-c509-cert	res	TBD3
	application/multipart-core	res	62
/sren	application/cose-c509-pkcs10	req	TBD4
	application/cose-c509-cert	res	TBD3
	application/multipart-core	res	62
/skg	application/cose-c509-pkcs10	req	TBD4
	application/multipart-core	res	62
/skc	N/A	req	-
	N/A	res	-
/att	N/A	req	-
	application/cose-c509-crtemplate	res	TBD19

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

Please note that Section 4.4 of [I-D.ietf-cose-cbor-encoded-cert] defines the format and the semantics of the response to /att.

Content-Format TBD3 MUST be supported by both EST-oscore clients and servers. A preference for any (future) Content-Format is to be expressed by the EST-client through the Accept option.

If a CoAP Accept option is not included in the request, the client is not expressing preference and the server SHOULD respond with a response application/multipart-core that includes the reference(s) to the enrolled certificate (e.g., c5t, c5u). The application/multipart-core response MUST include the reference(s) to the enrolled certificate which allows the client or any other party to retrieve it (e.g., through a coap URI secured with OSCORE). The application/multipart-core response MAY also include the actual certificate. The exact contents of the application/multipart-core response are dependent on the application policy. Because x5u and c5u references are of the same type application/cbor and the corresponding CoAP Content Format (60), mixing them in the same application/multipart-core response is not supported. An exception to the "SHOULD" is in the case when the request contains a DER-encoded ASN.1 object (e.g., application/pkcs10), when the server SHOULD respond with an appropriate ASN.1 object (see Section 4.3.1).

In the case of a request to /skg, the response contains two parts: certificate and the corresponding private key. The certificate part is encoded as the application/cose-c509-cert object (Content-Format identifier TBD3), 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 clients MUST use the /skg function for server-side generated keys.

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

Function	CBOR Response, Part 1	CBOR Response Part 2
/skg	TBD10	TBD3

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]. Therefore, the following applies:

* EST-oscore endpoints MUST support delayed responses (see Section 4.7 of [RFC9148])

* EST-oscore endpoints MUST support the following CoAP options: OSCORE, Uri-Host, Uri-Path, Uri-Port, Content-Format, Block1, Block2, and Accept. EST-oscore servers MUST implement Block1 and Block2. EST-oscore clients MUST implement Block2 and MAY implement Block1.

* The EST-coaps URLs based on coaps:// are translated to coap://, but with mandatory use of the CoAP OSCORE option.

4.5. CoAP response codes

See Section 4.5 in [RFC9148].

4.6. Message Fragmentation

The EDHOC key exchange is optimized for low message overhead, in particular when using 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 based on 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 the CoAP options Block1 and Block2 (see Section 4.4).

4.7. Delayed Responses

See Section 4.7 of [RFC9148].

4.8. Enrollment of Certificates with 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, or key transport.

The EST client attempting to enroll a DH key for a key usage operation other than digital signature can 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.

In some cases, it may be beneficial to exceptionally use the static DH private key associated with 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].

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 the optimized workflow based on the EDHOC + OSCORE combined request is being used as per Section 3 of [RFC9668], the client MUST use the ephemeral public 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 selected cipher suite used in the EDHOC session.

4.9. Enrollment of Certificates by Reference

The EST client may indicate preference for enrolling a certificate by reference. There are two cases to distinguish: 1) any certificate reference, or 2) a specific Content-Format. In the first case, the EST client indicates preference for receiving any certificate by reference by sending a CBOR-encoded request without the CoAP Accept option. In the second case, the EST client includes a Content-Format identifier in the CoAP Accept option indicating preference for receiving a specific reference (e.g., application/cose-certhash, application/cose-certhash;usage=c509, application/cbor containing a URI [I-D.ietf-cose-cbor-encoded-cert]). It is out of the scope of this specification how the certificate by reference gets resolved to the actual certificate by other parties participating in the communication with the EST client.

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 be 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 the transport protocol or potential transport layer security that may need to be terminated at the proxy, see Figure 2. Therefore, the concept "Registrar" and its required trust relationship 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 the EST Server and EST Client. If a secure association is needed between the EST Client and the CoAP-to-HTTP Proxy, this may also rely on OSCORE [I-D.ietf-core-oscore-capable-proxies].

```
                                           Constrained-Node Network
      .---------.                      .-----------------------------.
      |   CA    |                      |                             |
      '---------'                      |                             |
           |                           |                             |
       .------.  HTTP   .-----------------.   CoAP   .-----------.   |
       | EST  |<------->|  CoAP-to-HTTP   |<-------->| EST Client|   |
       |Server|  (TLS)  |      Proxy      |          '-----------'   |
       '------'         '-----------------'                          |
                                       |                             |
           <------------------------------------------------>        |
                            OSCORE     |                             |
                                       |                             |
                                       '-----------------------------'
```

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 the 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 the 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 security contexts to be used for authentication, meaning that EDHOC may not necessarily be present 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. As a special case, when used with EDHOC for the enrollment of static DH keys, this specification achieves connection-based channel binding by using the EDHOC ephemeral public key of the Responder as the public key in the proof-of-possession algorithm that 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 security contexts and the case where the signature key used for signing the CSR is different from the key used in the EDHOC session. 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 security contexts and of a re-enrollment request.

EST-servers MUST support the challengePassword attribute in PKCS#10 requests. How challengePassword is processed is outside of the scope of this specification and can be specified by an application policy.

7. IANA Considerations

This document does not require any IANA registrations.

8. References

8.1. Normative References

[I-D.ietf-cose-cbor-encoded-cert]
Mattsson, J. P., Selander, G., Raza, S., Hglund, J., and M. Furuhed, "CBOR Encoded X.509 Certificates (C509 Certificates)", Work in Progress, Internet-Draft, draft-ietf-cose-cbor-encoded-cert-16, 25 January 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-cbor-encoded-cert-16>.

[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>.

[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>.

[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., Preu 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>.

[RFC9668] Palombini, F., Tiloca, M., Hglund, 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)", RFC 9668, DOI 10.17487/RFC9668, November 2024, <https://www.rfc-editor.org/info/rfc9668>.

8.2. Informative References

[I-D.ietf-core-oscore-capable-proxies]
Tiloca, M. and R. Hglund, "OSCORE-capable Proxies", Work in Progress, Internet-Draft, draft-ietf-core-oscore-capable-proxies-05, 3 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-core-oscore-capable-proxies-05>.

[I-D.ietf-core-oscore-groupcomm]
Tiloca, M., Selander, G., Palombini, F., Mattsson, J. P., and R. Hglund, "Group Object Security for Constrained RESTful Environments (Group OSCORE)", Work in Progress, Internet-Draft, draft-ietf-core-oscore-groupcomm-28, 23 December 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-core-oscore-groupcomm-28>.

[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] Vuini, 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>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[RFC9112] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <https://www.rfc-editor.org/info/rfc9112>.

[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 [RFC9668]. 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 that contains the certificate.

```
EST Client                    EST Server
|                                      |
|     EDHOC message_1                  |
+------------------------------------->|
|     EDHOC message_2                  |
|<-------------------------------------+
|     EDHOC message_3                  |
|            +                         |
|     OSCORE request to /sen           |
+------------------------------------->|
|           POST /.well-known/est/sen  |
|           Accept: TBD3               |
|           Content-Format: TBD4       |
|           Payload: cose-c509-pkcs10  |
|                                      |
|     OSCORE response                  |
|<-------------------------------------+
|           2.04 Changed               |
|           Content-Format: TBD3       |
|           Payload: cose-c509-cert    |
```

Figure 3: Enrollment EST-oscore flow with optimizations.

Acknowledgments

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

Work on this document has in part been supported by the Horizon Europe Framework Programme project OpenSwarm (grant agreement No. 101093046).

Authors' Addresses

Gran Selander
Ericsson AB
Email: goran.selander@ericsson.com

Shahid Raza
RISE
Email: shahid.raza@ri.se

Martin Furuhed
Nexus
Email: martin.furuhed@nexusgroup.com

Malia Vuini
Inria
Email: malisa.vucinic@inria.fr

Timothy Claeys
Email: timothy.claeys@gmail.com