OCSP Stapling for EDHOC

Certificate Revocation in Resource Constrained Environments

Yousef AbdElKhalek, RISE

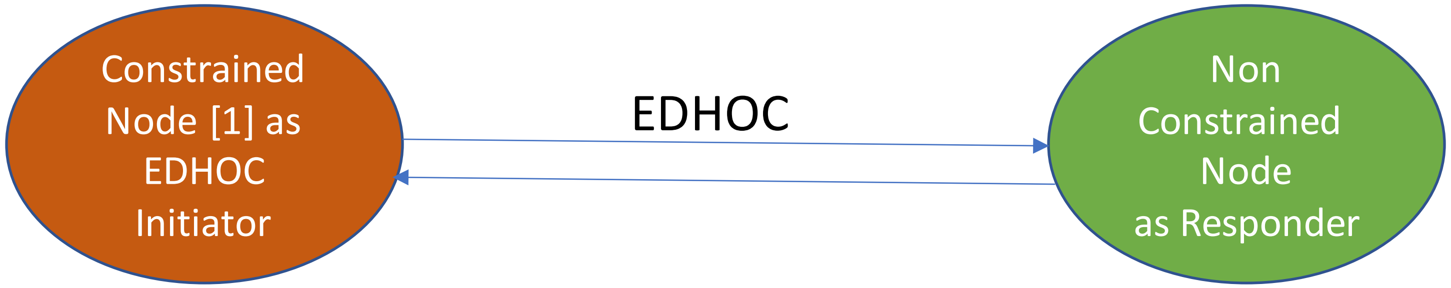
IETF 115 Meeting – London – November 8th, 2022

Motivation

PKI

As one more step for authenticating the responder, the Initiator wants to acquire Certificate-revocation information regarding responder's certificate

Initially untrusted Responder might be an adversary leveraging the absence of Certificate-revocation information transport to the constrained nodes



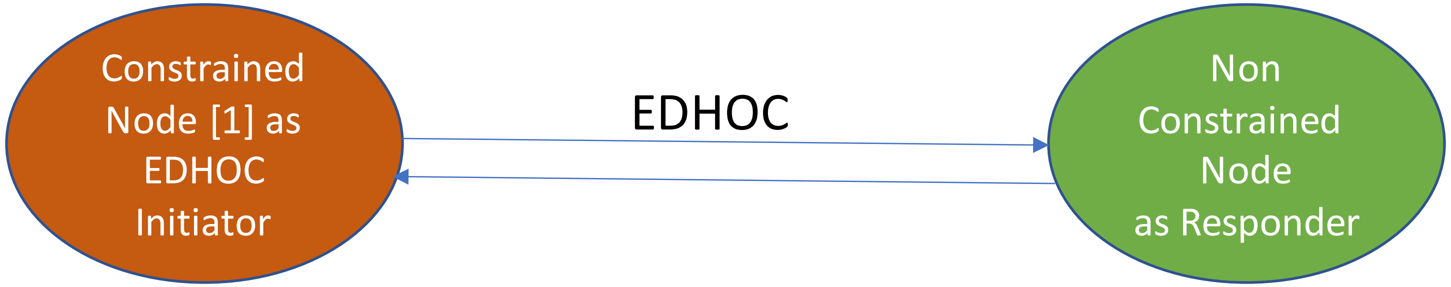
[1]-rfc7228

Motivation

PKI

-How is this achieved in a constrained environment ?

-How can a constrained node acquire certificate-revocation information ?

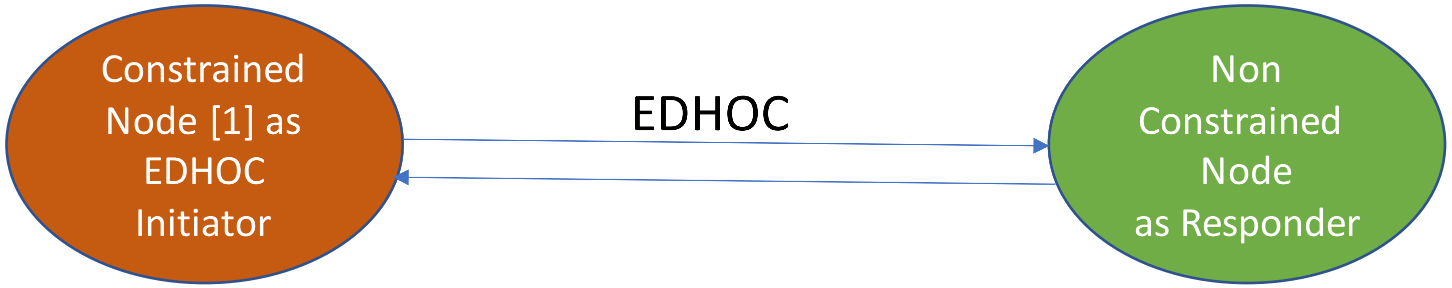


Motivation

PKI

-How is this achieved in a constrained environment ? -> Looking into transporting Revocation Information Via a Lightweight Key Exchange Protocol

-How can a constrained node acquire certificate-revocation information ?
> CRLs[2] ->OCSP[3]



[2]rfc5280

[3]rfc6960

Motivation

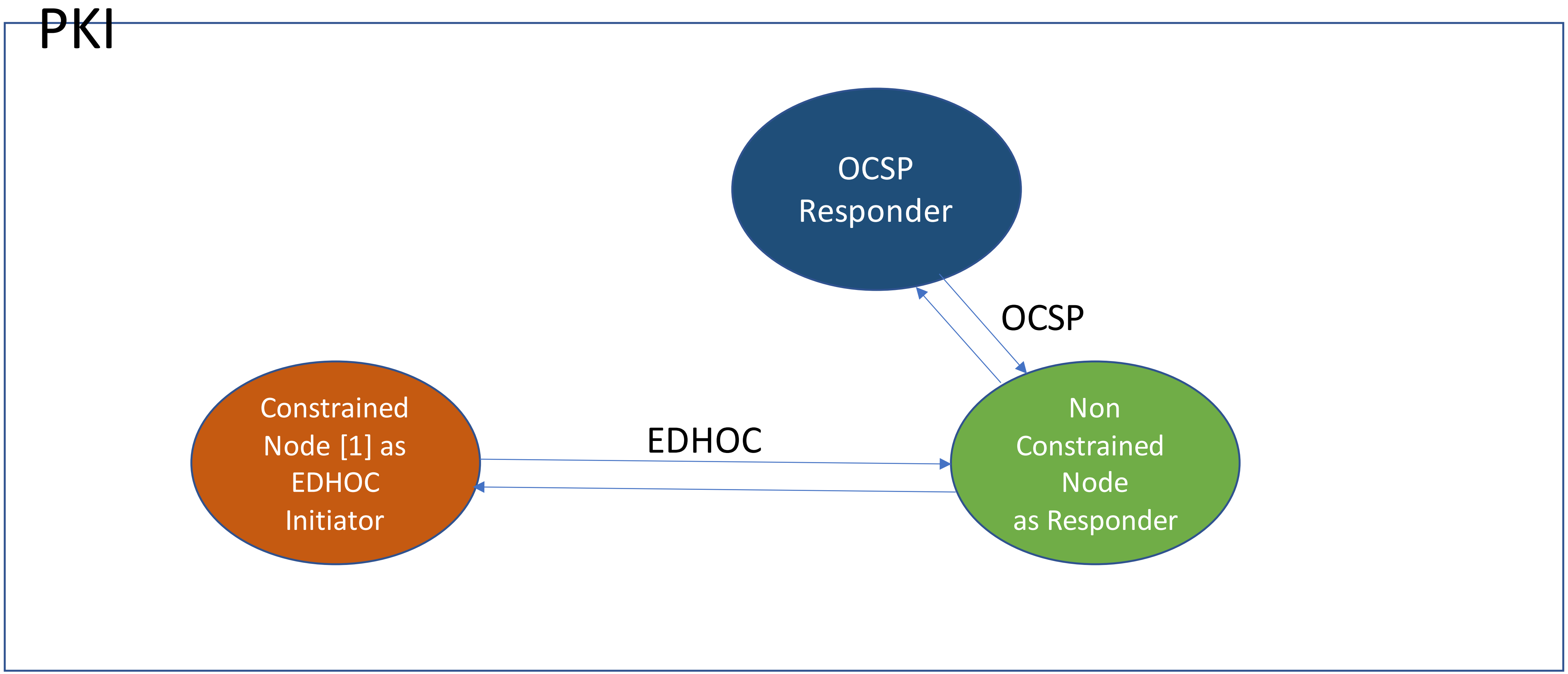
-CRLs[2] --> Too large, Constrained Node has limited RAM and Flash, cannot hold and cross-reference entire CRLs

-OCSP[3]--> Constrained Initiator Queries revocation status of Responder Certificate via an OCSP Request

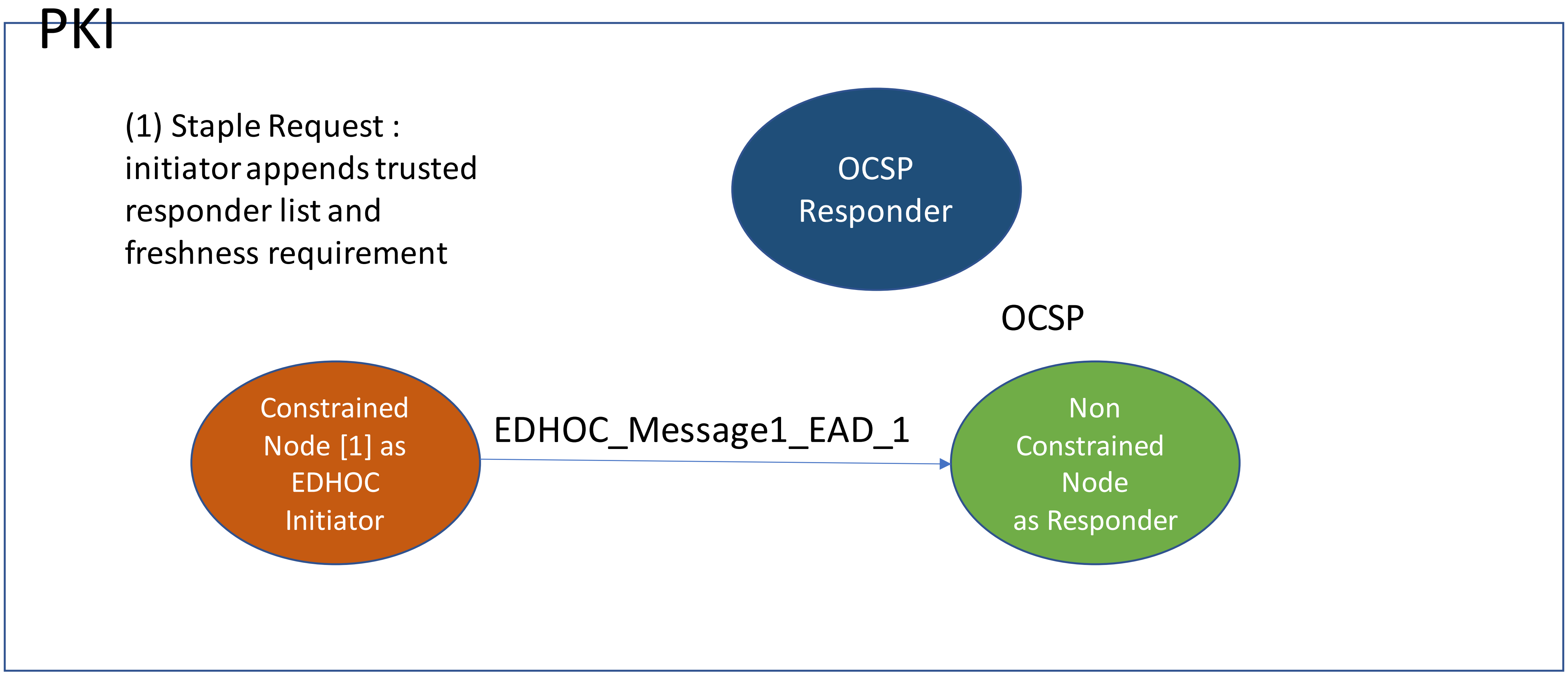
-What's better ? --> Remove the load on constrained node to perform the request
-> Instead leverage OCSP Stapling[4]

[2]rfc5280
[3]rfc6960
[4]rfc6961

OCSP Stapling in EDHOC –



OCSP Stapling in EDHOC – Staple Request



OCSP Stapling in EDHOC – Staple Request

EDHOC_Message1_EAD_1

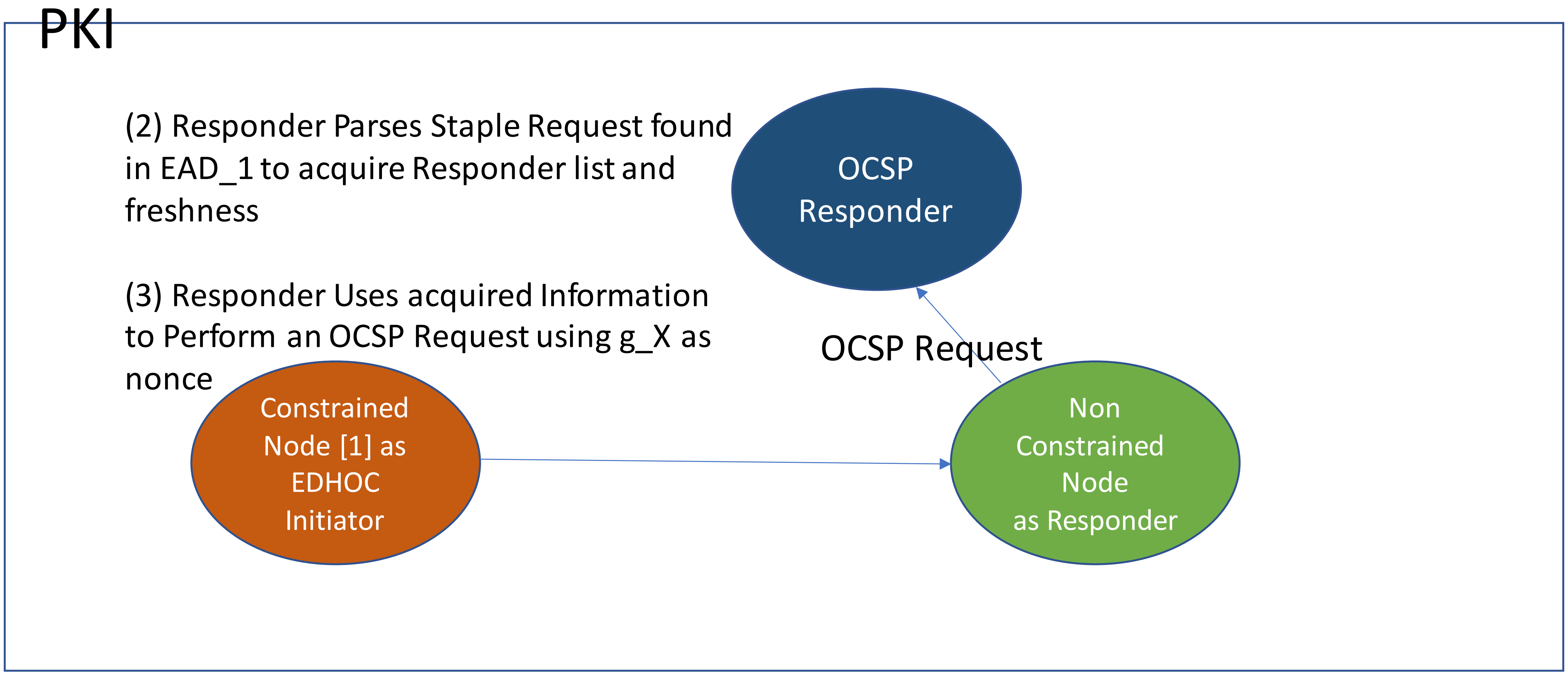
```
ead = 1* (
  ead_label : int,
  ead_value : bstr (staple-req)
 )
```

```
staple-req = (
    responderID_list : bstr,
    ? Fresh              : cbor[5] True (unsigned int)
 )
```

ead_label : Negative label for Critical EAD

[5]rfc8949

OCSP Stapling in EDHOC – OCSP Request



OCSP Request

- An OCSP Request gives a signed Timestamped DER Encoded ASN1 Response -> Size: 1600+ bytes

-No possibility to re-encode at responder side as the initiator needs the response to be signed by one of the OCSP responders in the trusted_responderList attached in staple request

-A tiny version of the OCSP response needs to be returned by the OCSP responder -> by tiny here we mean C509[6] OCSP Response

-How to signal an OCSP Responder to give a C509 response ?

[6]draft-ietf-cose-cbor-encoded-cert

OCSP Request

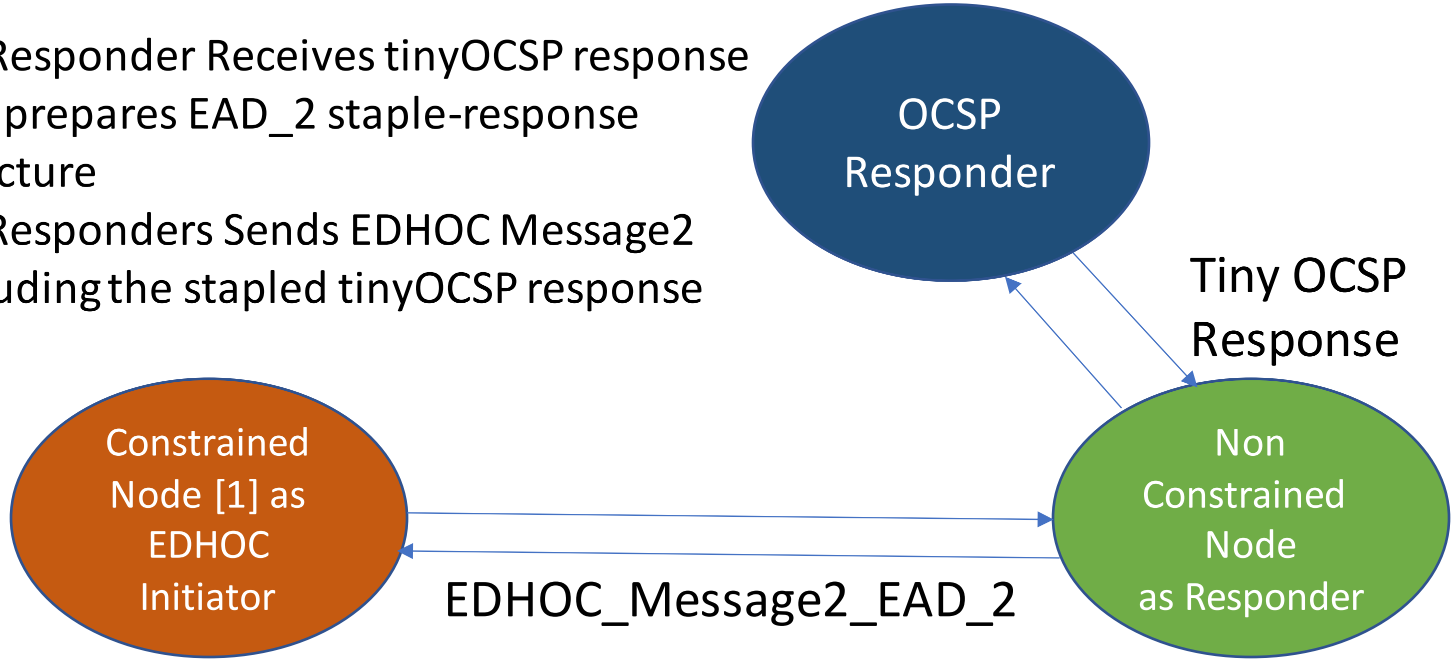
OCSP Request Data:
    Version: 1 (0x0)
    Requestor List:
        Certificate ID:
          Hash Algorithm: sha1
          Issuer Name Hash: 90C248EB881AAD4C41E5F8A862CCCD1FC246CA7B
          Issuer Key Hash: A176FA3149B9E1C9F640086E4A650E30DC314562
          Serial Number: 1001
    Request Extensions:
        OCSP Nonce: 8af6f430ebe18d3484017a9a11bf511c8dff8f834730b96c1b7c8dbca2fc3b6
        PreferredSignatureAlgorithms:
            algorithm=rsa-with-sha256-C509

Preferred signature extension is used to specify tiny response

OCSP Stapling in EDHOC – Staple Request

PKI

(4) Responder Receives tinyOCSP response and prepares EAD_2 staple-response structure
(5) Responders Sends EDHOC Message2 including the stapled tinyOCSP response



OCSP Stapling in EDHOC – tinyOCSP Response

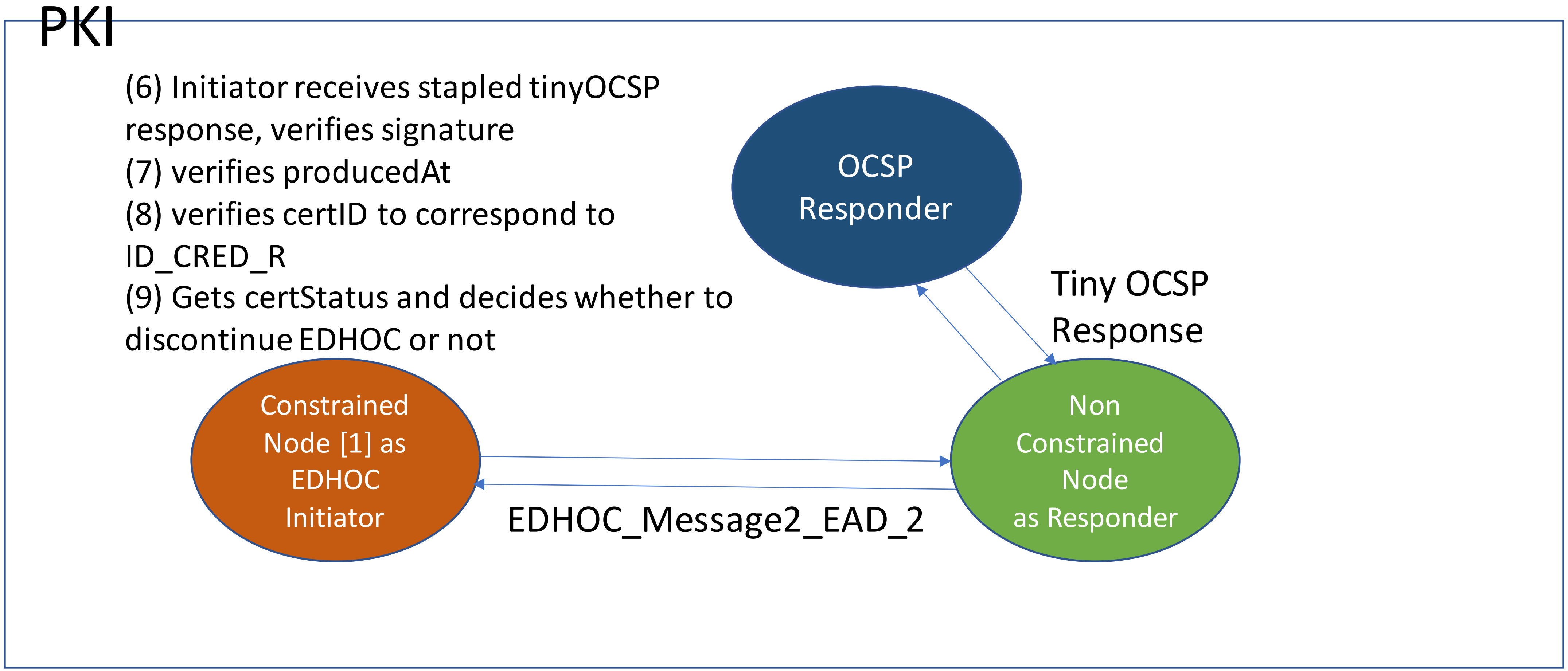
EDHOC_Message2_EAD_2

```
ead = 1* (
  ead_label : int,
  ead_value : bstr (staple-resp)
 )
```

```
  Staple-resp = (
  ResponseData : tinyOCSP_response
  SignatureVal    : bstr
  SignatureAlg    : unsigned int
 )
```

```
tinyOCSP_response = (
response_type: unsigned int
responderID:      byteString
ProducedAt:       cbor Time
nonce:            bytestring
certID:           c509 certID
cert status:      unsigned int
signatureVal:     bytestring
signatureAlg:     cose Label
)
```

OCSP Stapling in EDHOC – Staple Request



Message Size – tinyOCSP Response

-> A tiny OCSP Response is 275 bytes, compared to OCSP of 1600+ bytes,
This gives approximately an 83% reduction of the OCSP response size.

-> the tinyOCSP profile removes signerCert (leveraging ResponderList) from the OCSP response and removes the DER ASN1 structure, instead uses cbor for encoding and C509 conversion of X509 profiles

Transport Overhead in EDHOC

-> staple-req size in EAD_1 is determined by length of responderList which can be made NULL in the case of out of band agreement

-> An EDHOC_MESSAGE2 including staple-resp in EAD_2 can go up to 700 bytes which is still well within the range for a constrained node to handle

Implementation

-> Implemented tinyOCSP into openssl 3.0.5[7] mantaining regular functionality for OCSP (PR tbd)

->P.O.C extension of Stefan Hristosov's uosocore-uedhoc[8] library to handle EAD items that can affect the state of the protocol (Critical EADs) and linked with openssl tinyOCSP to acquire a tinyOCSP response (PR TBD)

->Testing environment uses an NRF52840 as constrained Initiator and Responder is a linux computer.

[7]https://github.com/openssl/openss

[8]https://github.com/eriptic/uosco
-uedhoc

Summary

->Showed a vector and the overhead of OCSP stapling in EDHOC to acquire certificate-revocation information in Resource Constrained Environments

->Introduced tinyOCSP which is an OCSP response profile leveraging c509

Next Steps

->Two Pull requests to be done

-> Power consumption overhead measurement of EDHOC with certificate revocation vs without.

->An internet draft regarding use of EAD items for certificate revocation (If found appropriate)

Thank you