CBOR Object Signing and Encryption (COSE): Header parameters for carrying and referencing X.509 certificates

draft-ietf-cose-x509-08

IETF 110, COSE, John Preuß Mattsson

Issues #29 #30 #31 #33

- Datatracker Status: Approved-announcement to be sent::Revised I-D Needed for 49 days

- However: 4 new issues on GitHub, confirmed by discussion on the list and during the interim

  - #29 Identification of end-entity cert / consistency with JWS
  - #30 Header protection and consistency with JWS
  - #31 What is the trust relationship for the x5u parameter?
  - #33 Allow OSCORE [RFC8613] for x5u CoAP URIs

- Pull Request #35 aims to aims to address issues #29 #30 #31 #33 based on the discussion on the list and during the last interim.

  - #35 Security fixes, clarification, and functionality fixes – Addressing #29 #30 #21 #33

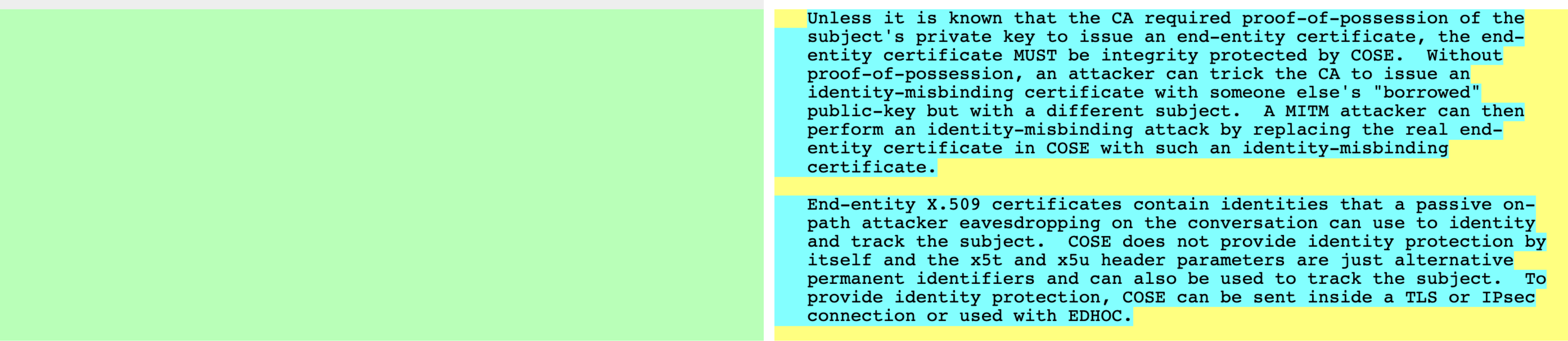
Pull Request #35

- The PR aims to aims to address issues #29 #30 #31 #33 based on the discussion on the list and during the last interim. The solution is to use x5t together with the other parameters as suggested by Russ:
  - Added to x5bag, x5chain, and x5u that integrity protection in COSE is required unless it is known that the CA did proof-of-possession.
  - Added that integrity protection can be achieved by combining x5t with x5bag, x5chain, or x5u.
  - Added explanation that sending x5bag or x5cahing in unprotected allows an intermediary to remove or add certificates.
  - Added clarification that x5t refer to an end-entity certificate.
  - Added media type application/cbor for a COSE_X509 chain.
  - Added that when the end-entity certificate is integrity protected by COSE, URI protection is not needed.
  - Security consideration on why integrity protection of the end-entity certificate is required is there was no proof-of-possession.
  - Security consideration on identity protection.

Pull Request #35 – Security Considerations

- Description of identity-misbinding attacks. Assumption is that the only reason to integrity protect the end-entity certificate is lack of proof-of-possession of the subjects private key as discussed in e.g. https://webee.technion.ac.il/~hugo/sigma-pdf.pdf

- Description of privacy issues with COSE not providing identity protection.

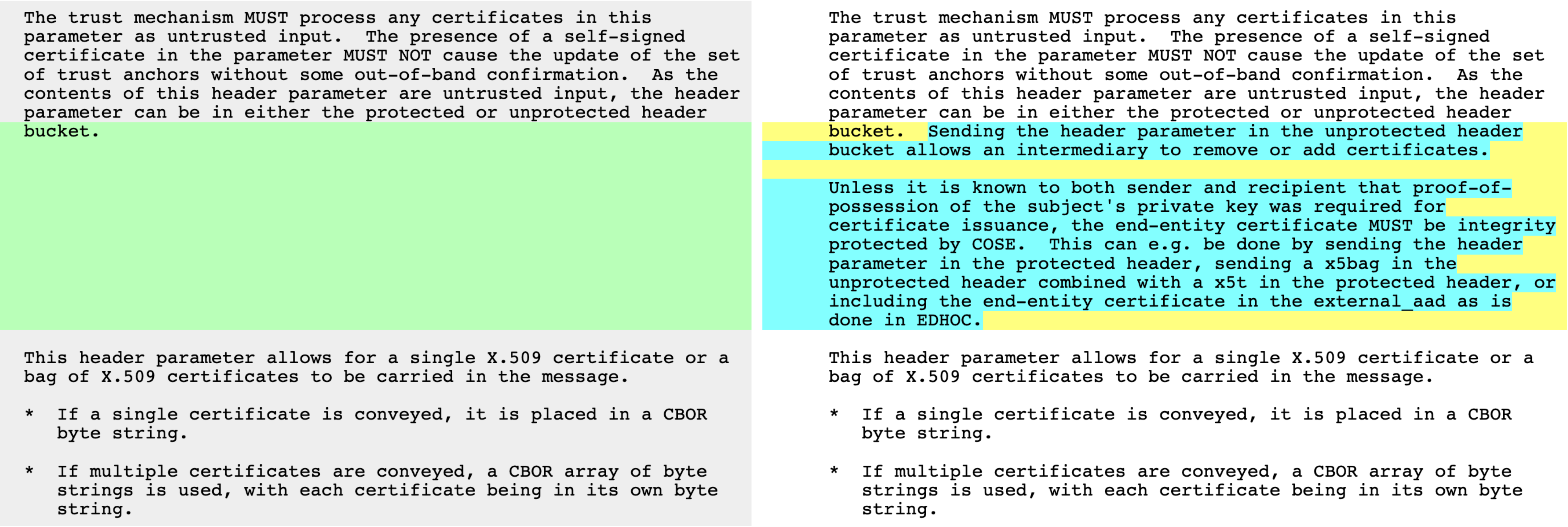


Unless it is known that the CA required proof-of-possession of the subject's private key to issue an end-entity certificate, the end-entity certificate MUST be integrity protected by COSE. Without proof-of-possession, an attacker can trick the CA to issue an identity-misbinding certificate with someone else's "borrowed" public-key but with a different subject. A MITM attacker can then perform an identity-misbinding attack by replacing the real end-entity certificate in COSE with such an identity-misbinding certificate.

End-entity X.509 certificates contain identities that a passive on-path attacker eavesdropping on the conversation can use to identity and track the subject. COSE does not provide identity protection by itself and the x5t and x5u header parameters are just alternative permanent identifiers and can also be used to track the subject. To provide identity protection, COSE can be sent inside a TLS or IPsec connection or used with EDHOC.

Pull Request #35 – x5bag

- Still possible to send x5bag in unprotected and let an intermediary remove or add CA certs.
- Example: Unprotected = { x5bag: COSE_X509 } Protected = { x5t: COSE_CertHash }



The trust mechanism MUST process any certificates in this parameter as untrusted input. The presence of a self-signed certificate in the parameter MUST NOT cause the update of the set of trust anchors without some out-of-band confirmation. As the contents of this header parameter are untrusted input, the header parameter can be in either the protected or unprotected header bucket.

This header parameter allows for a single X.509 certificate or a bag of X.509 certificates to be carried in the message.

* If a single certificate is conveyed, it is placed in a CBOR byte string.

* If multiple certificates are conveyed, a CBOR array of byte strings is used, with each certificate being in its own byte string.

The trust mechanism MUST process any certificates in this parameter as untrusted input. The presence of a self-signed certificate in the parameter MUST NOT cause the update of the set of trust anchors without some out-of-band confirmation. As the contents of this header parameter are untrusted input, the header parameter can be in either the protected or unprotected header bucket. Sending the header parameter in the unprotected header bucket allows an intermediary to remove or add certificates.

Unless it is known to both sender and recipient that proof-of-possession of the subject's private key was required for certificate issuance, the end-entity certificate MUST be integrity protected by COSE. This can e.g. be done by sending the header parameter in the protected header, sending a x5bag in the unprotected header combined with a x5t in the protected header, or including the end-entity certificate in the external_aad as is done in EDHOC.

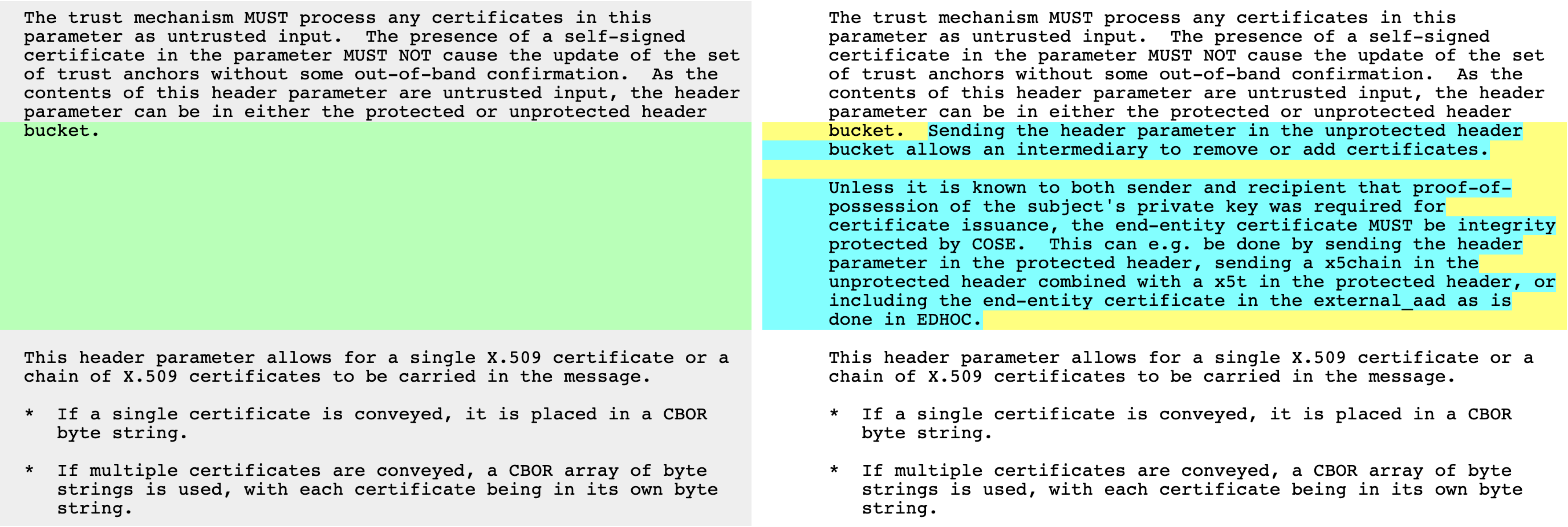
This header parameter allows for a single X.509 certificate or a bag of X.509 certificates to be carried in the message.

* If a single certificate is conveyed, it is placed in a CBOR byte string.

* If multiple certificates are conveyed, a CBOR array of byte strings is used, with each certificate being in its own byte string.

Pull Request #35 – x5chain

- Still possible to send x5chain in unprotected and let an intermediary remove or add CA certs.
- Example: Unprotected = { x5chain: COSE_X509 } Protected = { x5t: COSE_CertHash }

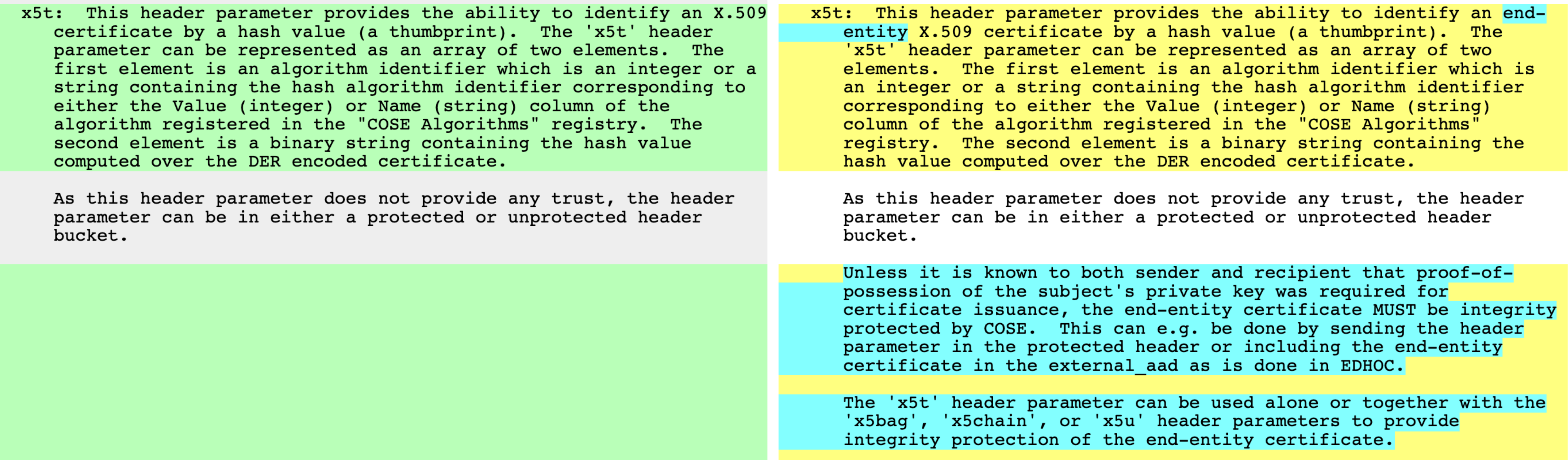


```
The trust mechanism MUST process any certificates in this
parameter as untrusted input.  The presence of a self-signed
certificate in the parameter MUST NOT cause the update of the set
of trust anchors without some out-of-band confirmation.  As the
contents of this header parameter are untrusted input, the header
parameter can be in either the protected or unprotected header
bucket.

This header parameter allows for a single X.509 certificate or a
chain of X.509 certificates to be carried in the message.

*  If a single certificate is conveyed, it is placed in a CBOR
   byte string.

*  If multiple certificates are conveyed, a CBOR array of byte
   strings is used, with each certificate being in its own byte
   string.
```

```
The trust mechanism MUST process any certificates in this
parameter as untrusted input.  The presence of a self-signed
certificate in the parameter MUST NOT cause the update of the set
of trust anchors without some out-of-band confirmation.  As the
contents of this header parameter are untrusted input, the header
parameter can be in either the protected or unprotected header
bucket.  Sending the header parameter in the unprotected header
bucket allows an intermediary to remove or add certificates.

Unless it is known to both sender and recipient that proof-of-
possession of the subject's private key was required for
certificate issuance, the end-entity certificate MUST be integrity
protected by COSE.  This can e.g. be done by sending the header
parameter in the protected header, sending a x5chain in the
unprotected header combined with a x5t in the protected header, or
including the end-entity certificate in the external_aad as is
done in EDHOC.

This header parameter allows for a single X.509 certificate or a
chain of X.509 certificates to be carried in the message.

*  If a single certificate is conveyed, it is placed in a CBOR
   byte string.

*  If multiple certificates are conveyed, a CBOR array of byte
   strings is used, with each certificate being in its own byte
   string.
```

Pull Request #35 – x5t

- Added that x5t identifies an end-entity certificate.

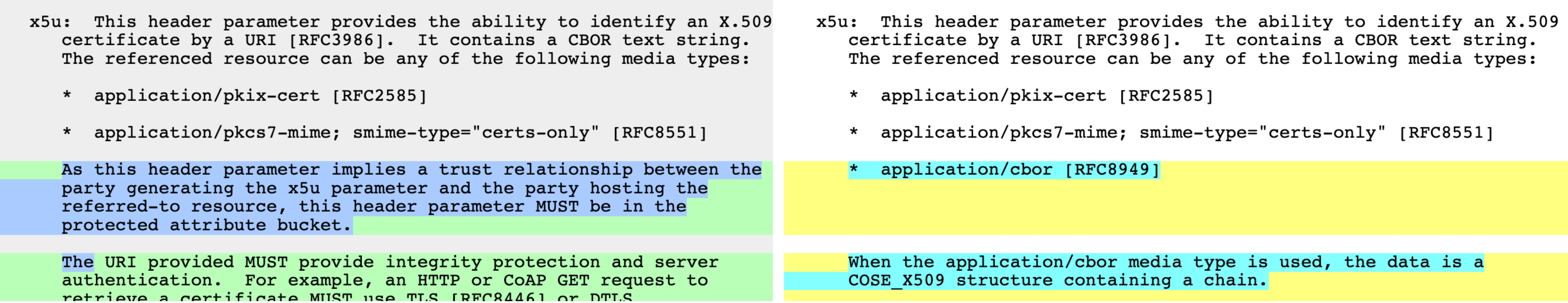


```
x5t:  This header parameter provides the ability to identify an X.509
   certificate by a hash value (a thumbprint).  The 'x5t' header
   parameter can be represented as an array of two elements.  The
   first element is an algorithm identifier which is an integer or a
   string containing the hash algorithm identifier corresponding to
   either the Value (integer) or Name (string) column of the
   algorithm registered in the "COSE Algorithms" registry.  The
   second element is a binary string containing the hash value
   computed over the DER encoded certificate.

   As this header parameter does not provide any trust, the header
   parameter can be in either a protected or unprotected header
   bucket.
```

```
x5t:  This header parameter provides the ability to identify an end-
   entity X.509 certificate by a hash value (a thumbprint).  The
   'x5t' header parameter can be represented as an array of two
   elements.  The first element is an algorithm identifier which is
   an integer or a string containing the hash algorithm identifier
   corresponding to either the Value (integer) or Name (string)
   column of the algorithm registered in the "COSE Algorithms"
   registry.  The second element is a binary string containing the
   hash value computed over the DER encoded certificate.

   As this header parameter does not provide any trust, the header
   parameter can be in either a protected or unprotected header
   bucket.

   Unless it is known to both sender and recipient that proof-of-
   possession of the subject's private key was required for
   certificate issuance, the end-entity certificate MUST be integrity
   protected by COSE.  This can e.g. be done by sending the header
   parameter in the protected header or including the end-entity
   certificate in the external_aad as is done in EDHOC.

   The 'x5t' header parameter can be used alone or together with the
   'x5bag', 'x5chain', or 'x5u' header parameters to provide
   integrity protection of the end-entity certificate.
```

Pull Request #35 – x5u (page 1 / 2)

— Allow media type application/cbor with COSE_X509 containing a chain.

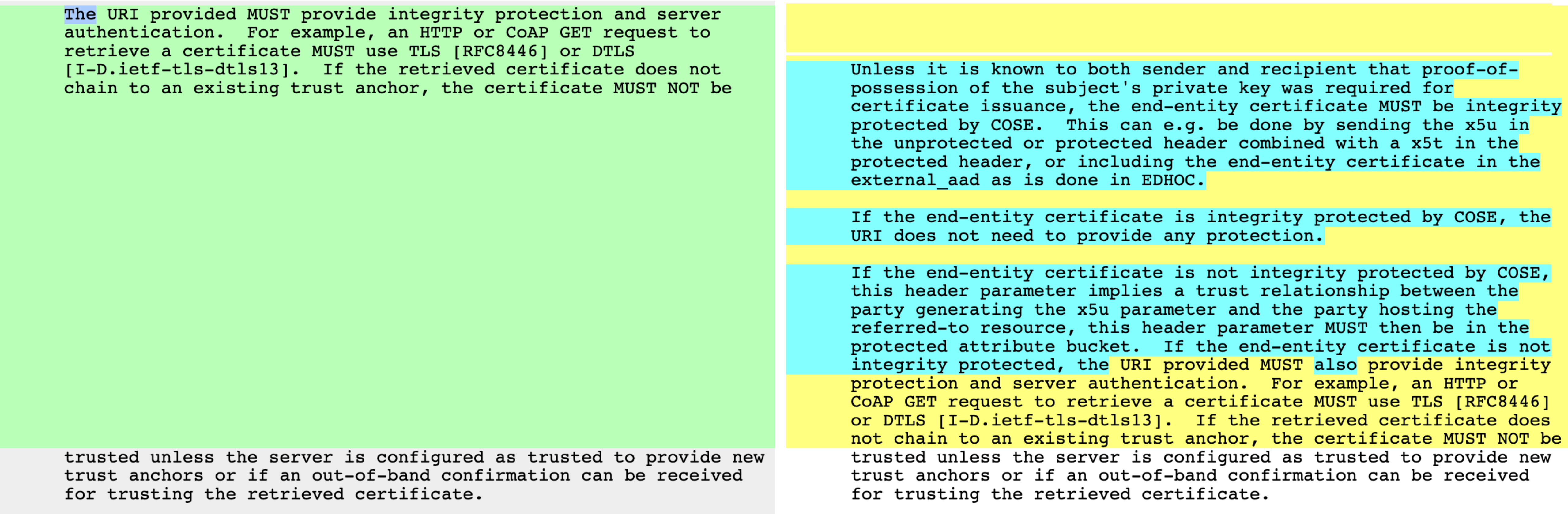


```
x5u:  This header parameter provides the ability to identify an X.509
   certificate by a URI [RFC3986].  It contains a CBOR text string.
   The referenced resource can be any of the following media types:

   *  application/pkix-cert [RFC2585]

   *  application/pkcs7-mime; smime-type="certs-only" [RFC8551]

   As this header parameter implies a trust relationship between the
   party generating the x5u parameter and the party hosting the
   referred-to resource, this header parameter MUST be in the
   protected attribute bucket.

   The URI provided MUST provide integrity protection and server
   authentication.  For example, an HTTP or CoAP GET request to
   retrieve a certificate MUST use TLS [RFC8446] or DTLS
```

```
x5u:  This header parameter provides the ability to identify an X.509
   certificate by a URI [RFC3986].  It contains a CBOR text string.
   The referenced resource can be any of the following media types:

   *  application/pkix-cert [RFC2585]

   *  application/pkcs7-mime; smime-type="certs-only" [RFC8551]

   *  application/cbor [RFC8949]



   When the application/cbor media type is used, the data is a
   COSE_X509 structure containing a chain.
```

Pull Request #35 – x5u (page 2 / 2)

- Example: Unprotected = { x5bag: COSE_X509 } Protected = { x5t: COSE_CertHash }
- Two cases. If the end-entity certiticate is integrity protected by COSE, TLS or DTLS is not needed.



```
   The URI provided MUST provide integrity protection and server
   authentication.  For example, an HTTP or CoAP GET request to
   retrieve a certificate MUST use TLS [RFC8446] or DTLS
   [I-D.ietf-tls-dtls13].  If the retrieved certificate does not
   chain to an existing trust anchor, the certificate MUST NOT be
   trusted unless the server is configured as trusted to provide new
   trust anchors or if an out-of-band confirmation can be received
   for trusting the retrieved certificate.
```

```
   Unless it is known to both sender and recipient that proof-of-
   possession of the subject's private key was required for
   certificate issuance, the end-entity certificate MUST be integrity
   protected by COSE.  This can e.g. be done by sending the x5u in
   the unprotected or protected header combined with a x5t in the
   protected header, or including the end-entity certificate in the
   external_aad as is done in EDHOC.

   If the end-entity certificate is integrity protected by COSE, the
   URI does not need to provide any protection.

   If the end-entity certificate is not integrity protected by COSE,
   this header parameter implies a trust relationship between the
   party generating the x5u parameter and the party hosting the
   referred-to resource, this header parameter MUST then be in the
   protected attribute bucket.  If the end-entity certificate is not
   integrity protected, the URI provided MUST also provide integrity
   protection and server authentication.  For example, an HTTP or
   CoAP GET request to retrieve a certificate MUST use TLS [RFC8446]
   or DTLS [I-D.ietf-tls-dtls13].  If the retrieved certificate does
   not chain to an existing trust anchor, the certificate MUST NOT be
   trusted unless the server is configured as trusted to provide new
   trust anchors or if an out-of-band confirmation can be received
   for trusting the retrieved certificate.
```

Pull Request #35

- Proposed changes were made so that no existing secure deployment need to change their implementation. Could otherwise be discussed if integrity protection should be a MUST, but that would change existing implementations (which is they do proof-of-possession are already secure).

- The PR aims to addresses all the related use case and security issues.
  - If the requirement are followed, it is secure.
  - No changes required to existing secure deployments.
  - Still possible to send x5bag and x5chain in unprotected.
  - No extra overhead is required when used in EDHOC.
  - Implementation of application/pkcs7-mime is not required for chains
  - When used in EDHOC, plain unprotected CoAP can be used.