LAMPS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 22 April 2026

H. Tschofenig
H. Brockhaus
Siemens
J. Mandel
AKAYLA
S. Turner
sn3rd

19 October 2025

Nonce-based Freshness for Remote Attestation in Certificate Signing Requests (CSRs) for the Certification Management Protocol (CMP), for Enrollment over Secure Transport (EST), and for Certificate Management over CMS (CMC)

draft-ietf-lamps-attestation-freshness-05

Abstract

When an end entity includes attestation Evidence in a Certificate Signing Request (CSR), it may be necessary to demonstrate the freshness of the provided Evidence. Current attestation technology commonly achieves this using nonces.

This document outlines the process through which nonces are supplied to the end entity by an RA/CA for inclusion in Evidence, leveraging the Certificate Management Protocol (CMP), Enrollment over Secure Transport (EST), and Certificate Management over CMS (CMC).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 April 2026.

Table of Contents

1. Introduction

The management of certificates, encompassing issuance, CA certificate provisioning, renewal, and revocation, has been streamlined through standardized protocols.

The Certificate Management Protocol (CMP) [I-D.ietf-lamps-rfc4210bis] defines messages for X.509v3 certificate creation and management. CMP facilitates interactions between end entities and PKI management entities, such as Registration Authorities (RAs) and Certification Authorities (CAs). For Certificate Signing Requests (CSRs), CMP primarily utilizes the Certificate Request Message Format (CRMF) [RFC4211] but also supports PKCS#10 [RFC2986].

Enrollment over Secure Transport (EST) ([RFC7030], [RFC8295]) is another certificate management protocol that provides a subset of CMP's features, primarily using PKCS#10 for CSRs.

Certificate Management over CMS (CMC) [I-D.ietf-lamps-rfc5272bis] is a certificate management protocol using the Cryptographic Message Syntax (CMS).

When an end entity requests a certificate from a Certification Authority (CA), it may need to assert credible claims about the protections of the corresponding private key, such as the use of a hardware security module or the protective capabilities provided by the hardware, as well as claims about the platform itself.

To include these claims as Evidence in remote attestation, the remote attestation extension [I-D.ietf-lamps-csr-attestation] has been defined. It specifies how Evidence produced by an Attester is encoded for inclusion in CRMF or PKCS#10, along with any necessary certificates for its validation.

For a Verifier or Relying Party to ensure the freshness of the Evidence, knowing the exact time of its production is crucial. Current attestation technologies, like [TPM20] and [RFC9783], often employ nonces to ensure the freshness of Evidence. Further details on ensuring Evidence freshness can be found in Section 10 of [RFC9334].

Section 4 of [I-D.ietf-lamps-csr-attestation] provides examples where a CSR contains one or more Evidence statements. For each Evidence statement the end entity may wish to request a separate nonce.

Since an end entity requires one or more nonces from one or more Verifiers via the RA/CA, an additional roundtrip is necessary. However, a CSR is a one-shot message. Therefore, CMP, EST, and CMC enable the end entity to request information from the RA/CA before submitting a certification request conveniently.

Once a nonce is obtained, the end entity invokes the API on an Attester, providing the nonce as an input parameter. The Attester then returns Evidence, which is embedded into a CSR and potentially together with further Evidence statements, submitted back to the RA/CA in a certification request message.

Figure 1 illustrates this interaction:

* One or more nonces are requested in step (0) and obtained in step (1) using the extension to CMP/EST/CMC defined in this document.

* The CSR extension [I-D.ietf-lamps-csr-attestation] conveys one or more Evidence statements to the RA/CA in step (2).

* One or more Verifiers process the received Evidence and return the Attestation Result(s) to the Relying Party. The CA uses the Attestation Result(s) with the Appraisal Policy and other information to create the requested certificate. The certificate is returned to the End Entity in step (3).

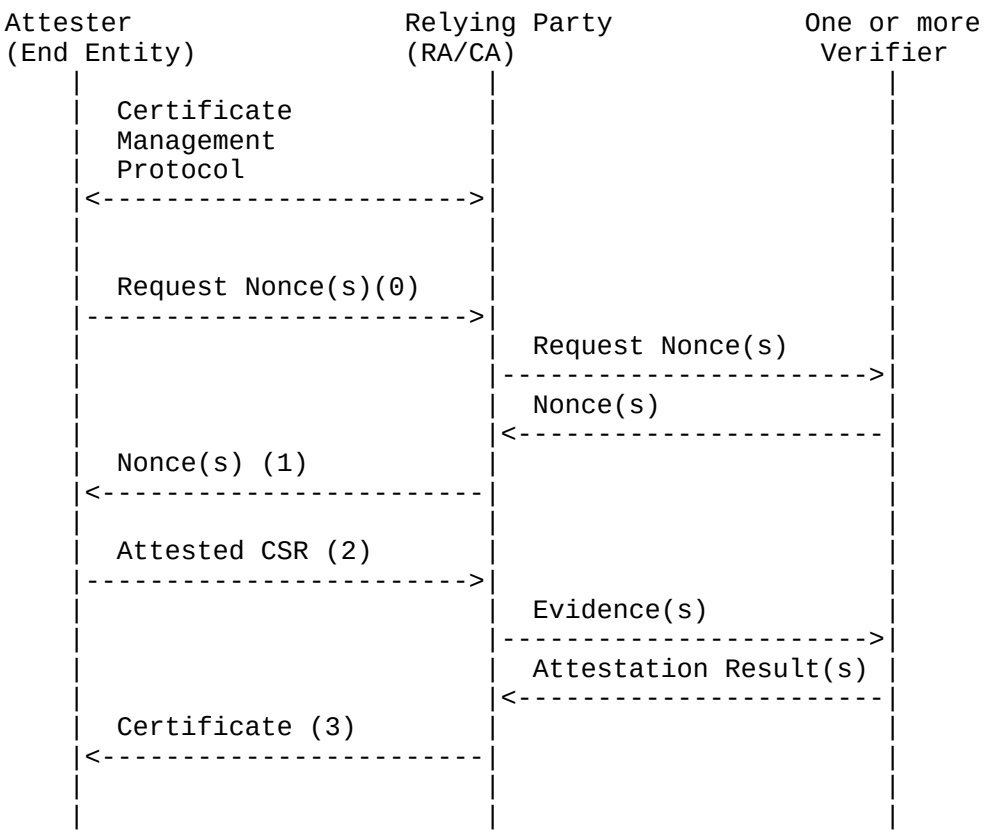

Figure 1: Architecture with Background Check Model.

The functionality described in this document is divided into three sections:

* Section 3 describes how to convey the nonce using CMP.

* Section 4 describes the equivalent functionality for EST.

* Section 5 describes the equivalent functionality for CMC.

2. Terminology and Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The terms Attester, Relying Party, Verifier and Evidence are defined in [RFC9334]. The terms end entity, certification authority (CA), and registration authority (RA) are defined in [RFC5280].

We use the terms Certificate Signing Request (CSR) and certification request interchangeably.

3. Conveying a Nonce in CMP

Section 5.3.19 of [I-D.ietf-lamps-rfc4210bis] defines the general request message (genm) and general response (genp). The NonceRequest payload of the genm message, sent by the end entity to request a nonce, optionally includes details on the required length of the nonce from the Attester. The NonceResponse payload of the genp message, sent by the CA/RA in response to the request, contains the nonce itself.

```
GenMsg:    {id-it TBD1}, NonceRequestValue
GenRep:    {id-it TBD2}, NonceResponseValue | < absent >

id-it-nonceRequest OBJECT IDENTIFIER ::= { id-it TBD1 }
NonceRequestValue ::= SEQUENCE SIZE (1..MAX) OF NonceRequest
NonceRequest ::= SEQUENCE {
   len INTEGER OPTIONAL,
   -- indicates the required length of the requested nonce
   type EVIDENCE-STATEMENT.&id({EvidenceStatementSet}) OPTIONAL,
   -- indicates which Evidence type to request a nonce for
   hint UTF8String OPTIONAL
   -- indicates which Verifier to request a nonce from
}

id-it-nonceResponse OBJECT IDENTIFIER ::= { id-it TBD2 }
NonceResponseValue ::= SEQUENCE SIZE (1..MAX) OF NonceResponse
NonceResponse ::= SEQUENCE {
   nonce OCTET STRING,
   -- contains the nonce of length len
   -- provided by the Verifier indicated with hint
   expiry INTEGER OPTIONAL,
   -- indicates how long in seconds the Verifier considers
   -- the nonce valid
   type EVIDENCE-STATEMENT.&id({EvidenceStatementSet}) OPTIONAL,
   -- indicates which Evidence type to request a nonce for
   hint UTF8String OPTIONAL
   -- indicates which Verifier to request a nonce from
}
```

The end entity may request one or more nonces for different Verifiers. The EVIDENCE-STATEMENT type is defined in [I-D.ietf-lamps-csr-attestation]. It allows the Attester to specify to the Relying Party which Verifier should be contacted to obtain a nonce. If a NonceRequest structure does not contain type or hint, the RA/CA MAY generate a nonce itself and include it in the response.

The use of the general request/response message exchange introduces an additional round trip for transmitting nonce(s) from the CA/RA to the end entity (and subsequently to the Attester within the end entity).

The end entity MUST construct an id-it-nonceRequest message to prompt the RA/CA to send one or more nonces in response. The message may contain one or more NonceRequest structures, at a maximum one per Evidence statement the end entity wishes to provide in a CSR. If a NonceRequest structure does neither contain a type nor a hint, the RA/CA MAY generate a nonce itself and provide it in the respective NonceResponse structure. If an RA/CA is not able to provide a requested nonce, it MUST provide an empty OCTET STRING in the respective NonceResponse structure.

NonceRequest, NonceResponse, and EvidenceStatement structures can contain a type field and a hint field. In terms of type and hint content, the order in which the NonceRequest structures were sent in the request message MUST match the order of the NonceResponse structures in the response message and the EvidenceStatements in the CSR later. This matching ensures that the RA/CA can send each Evidence statement to the same Verifier that generated the corresponding nonce used by the Attester.

When receiving nonces from the RA/CA in an id-it-nonceResponse message, the end entity MUST use them to request Evidence statements from the respective Attester, as optionally indicated by type and hint. If a nonce is provided in a NonceResponse structure without indicating any type or hint, it can be used for all Evidence statements requiring a nonce.

An Evidence statement generated using a nonce provided with an expiry value will be accepted by the Verifier as valid until the respective expiry time has elapsed. It is expected that the respective messages are delivered in a timely manner.

The interaction is illustrated in Figure 2.

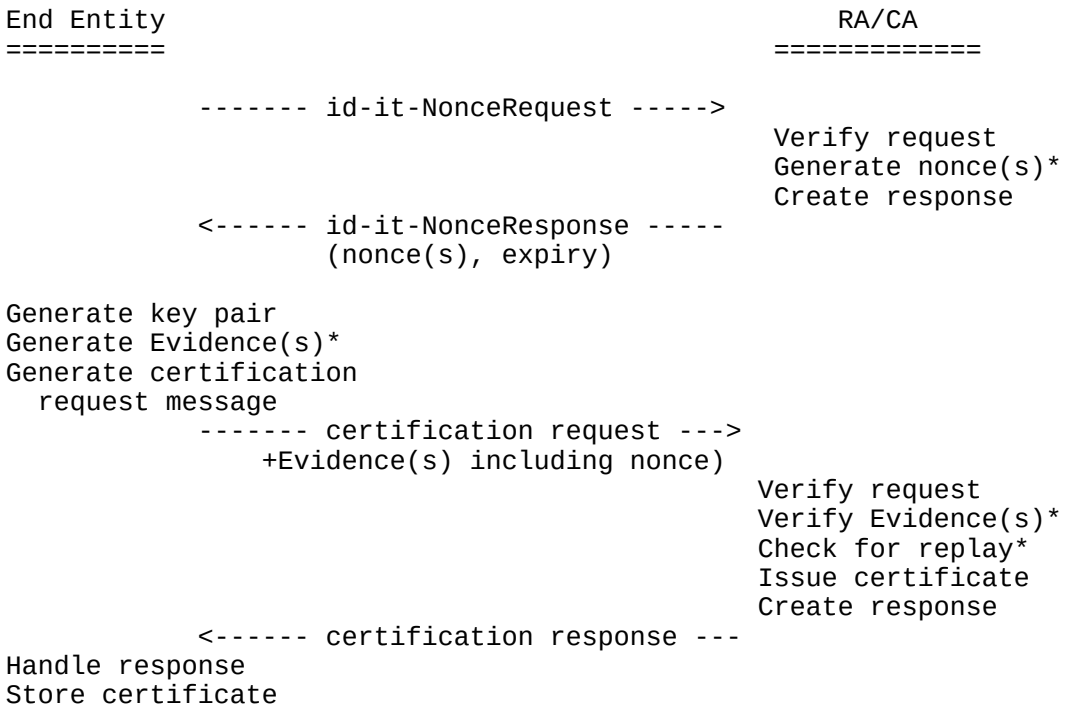

Figure 2: CMP Exchange with Nonce and Evidence.

If HTTP is used to transfer the NonceRequest and NonceResponse messages, the OPTIONAL <operation> path segment defined in Section 3.6 of [I-D.ietf-lamps-rfc4210bis] MAY be used.

Operation	Operation path	Details
Get Attestation Freshness Nonce	getnonce	{{CMP}}

If CoAP is used for transferring NonceRequest and NonceResponse messages, the OPTIONAL <operation> path segment defined in Section 2.1 of [RFC9482] MAY be used.

Operation	Operation path	Details
Get Attestation Freshness Nonce	nonce	{{CMP}}

4. Conveying a Nonce in EST

The EST client requests one or more nonces for its Attester from the EST server. This function typically follows the request for CA certificates and precedes other EST operations.

The EST server MUST support the path-prefix of "/.well-known/" as defined in [RFC5785] and the registered name of "est". Therefore, a valid EST server URI path begins with "https://www.example.com/.well-known/est". Each EST operation is indicated by a path-suffix that specifies the intended operation.

The following operation is defined by this specification:

Operation	Operation path	Details
Retrieval of a nonce	/nonce	{{EST}}

The operation path is appended to the path-prefix to form the URI used with HTTP GET or POST to perform the desired EST operation. An example of a valid URI absolute path for the "/nonce" operation is "/.well-known/est/nonce".

4.1. Request Methods

An EST client uses either a GET or a POST method, depending on whether additional parameters need to be conveyed:

* A GET request MUST be used when the EST client does not want to convey extra parameters.

* A POST request MUST be used when parameters, such as nonce length or a hint about the verification service, are included in the request.

Message type (per operation)	Media type(s)	Reference
Nonce Request	N/A (for GET) or application/json (for POST)	This section
Nonce Response	application/json	This section

4.2. Request Payload (POST)

The payload in a POST request MUST be of content-type "application/json" and MUST contain an array of JSON objects [RFC8259] with the optional members "len", "type", and "hint".

* The optional "len" member indicates the length of the requested nonce value in bytes.

* The optional "type" member contains an EvidenceStatement OID (dotted-decimal string) as defined in [I-D.ietf-lamps-csr-attestation].

* The optional "hint" member contains an FQDN or URI identifying the Verifier, following the EvidenceHint structure as defined in [I-D.ietf-lamps-csr-attestation].

The order of objects in the JSON array is significant and MUST be preserved by the server. The response array MUST contain the same number of elements in the same order so clients can correlate requests and responses by array index.

4.2.1. Example GET

```
GET /.well-known/est/nonce HTTP/1.1
```

4.2.2. Example POST

To retrieve one or more nonces while optionally specifying the length, type, and/or hint:

```
POST /.well-known/est/nonce HTTP/1.1
Content-Type: application/json
[
  {
    "len": 8,
    "type": "1.2.3.4.5",
    "hint": "https://example.com"
  }
]
```

4.3. Server Response

If successful, the EST server MUST respond with an HTTP 200 status code and a content-type of "application/json", containing an array of JSON objects [RFC8259] with the "nonce" member. The "expiry" member is optional and indicates the absolute expiry time of the nonce encoded as an RFC 3339 timestamp string. The optional "type" and "hint" members MAY be copied from the request to aid correlation.

Note: CMP encodes "expiry" as an INTEGER representing seconds of validity. EST encodes "expiry" as an absolute timestamp.

Below is an example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
[
  {
    "nonce": "MTIzNDU2Nzg5MDEyMzQ1Njc4OTAxMjM0NTY3ODkwMTI=",
    "expiry": "2031-10-12T07:20:50.52Z",
    "type": "1.2.3.4.5",
    "hint": "https://example.com"
  }
]
```

The EST server MAY request HTTP-based client authentication, as explained in Section 3.2.3 of [RFC7030].

Open Issue: Should a specific content type be registered for use with EST over CoAP, where the nonce and expiry fields are encoded in a CBOR structure?

5. Conveying a Nonce in CMC

CMC defines Simple and Full PKI Requests for the client to use to request a certificate. Full PKI Requests provide the client with more functionality through the use of Controls, defined in Section 6 of [I-D.ietf-lamps-rfc5272bis]. Currently, the client sends an initial request containing a certification request (CRMF, PKCS#10, or other). To allow the client to request a nonce prior to sending a certification request, this section defines the nonceReq and nonceResp.

Generally a Full PKI Request is encapsulated in a SignedData or AuthenticatedData with an encapsulated content type of 'id-cct-PKIData'. To accommodate a generic request for a nonce, the Client/Server SHOULD use the Data content type; id-data, to transmit the nonceReq and nonceResp controls. The syntax for the controls uses the same syntax as the CMP information types defined in Section 3.

The NonceRequest control is identified by:

id-cmc-nonceReq OBJECT IDENTIFIER ::= { id-it TBD1 }

The NonceResponse control is identified by: ~~~ id-cmc-nonceResp OBJECT IDENTIFIER ::= { id-it TBD2 } ~~~

5.1. Generic Nonce Request Message Flow

The client sends id-cmc-nonceReq structure to the server. Upon receiving and processing the request, the server responds with id-cmc-nonceResp.

Once this round-trip transaction is complete, the client will include the nonce in either a Simple or Full PKI Request.

Client to Server: ~~~ ContentInfo.contentType = id-Data ContentInfo.content eContentType = id-cct-PKIData eContent controlSequence {101, id-cmc-senderNonce, 10001} {102, id-cmc-nonceReq, <sequence of nonce request>} ~~~

Server to Client: ~~~ ContentInfo.contentType = id-Data ContentInfo.content eContentType = id-cct-PKIData eContent controlSequence {101, id-cmc-senderNonce, 10005} {102, id-cmc-recipientNonce, 10001} {103, id-cmc-nonceResp, <sequence of nonce response>} ~~~

6. Nonce Processing Guidelines

When the RA/CA is requested to provide a nonce to an end entity, it interacts with the Verifier. According to the IETF RATS architecture [RFC9334], the Verifier is responsible for validating Evidence about an Attester and generating Attestation Results for use by a Relying Party. The Verifier also acts as the source of the nonce to prevent replay attacks.

The nonce value MUST contain a random byte sequence with at least 64 bits of entropy. The RA/CA MUST ensure that nonces are unique and MUST NOT be reused. The length of the nonce depends on the remote attestation technology in use, as specific nonce lengths may be required by the end entity. This specification assumes that the RA/CA possesses knowledge, either out-of-band or through the len field in the NonceRequest, regarding the required nonce length for the attestation technology. Nonces of incorrect length will cause the remote attestation protocol to fail.

For instance, the PSA attestation token [RFC9783] supports nonce lengths of 32, 48, and 64 bytes. Other attestation technologies employ nonces of similar lengths.

If a specific length was requested, the RA/CA MUST provide a nonce of that size. The end entity MUST use the received nonce if the remote attestation supports the requested length. If necessary, the end entity MAY adjust the length of the nonce by truncating or padding it accordingly.

While this specification does not address the semantics of the attestation API or the underlying software/hardware architecture, the API returns Evidence from the Attester in a format specific to the attestation technology used and specified by the type and hint. The returned Evidence is encapsulated within the CSR, as defined in [I-D.ietf-lamps-csr-attestation]. The software generating the CSR treats the Evidence as an opaque blob and does not interpret its format. It’s crucial to note that the nonce is included in the Evidence, either implicitly or explicitly, and MUST NOT be conveyed in CSR structures outside of the Evidence payload.

The processing of CSRs containing Evidence is detailed in [I-D.ietf-lamps-csr-attestation]. Importantly, certificates issued based on this process do not contain the nonce, as specified in [I-D.ietf-lamps-csr-attestation].

7. IANA Considerations

This document adds new entries to the "CMP Well-Known URI Path Segments" registry defined in [RFC8615].

Path Segment	Description	Reference
getnonce	Get Attestation Freshness Nonce over HTTP	{{CMP}}
nonce	Get Attestation Freshness Nonce over CoAP	{{CMP}}

[Open Issue: Register path segments for EST]

IANA is also requested to register the following ASN.1 [X.680] module OID in the "SMI Security for PKIX Module Identifier" registry (1.3.6.1.5.5.7.0). This OID is defined in Appendix A.

Decimal	Description	References
TBDMOD	id-mod-att-fresh-req	This-RFC

Table 1

8. Security Considerations

This specification details the process of obtaining a nonce via CMP, EST, and CMC, assuming that the nonce does not require confidentiality protection while maintaining the security properties of the remote attestation protocol. [RFC9334] defines the IETF remote attestation architecture and extensively discusses nonce-based freshness.

Section 8.4 of [RFC9711] specifies requirements for the randomness and privacy of nonce generation when used with the Entity Attestation Token (EAT). These requirements, which are also adopted by attestation technologies like the PSA attestation token [RFC9783], provide general utility:

* The nonce MUST have at least 64 bits of entropy.

* To prevent disclosure of privacy-sensitive information, it should be derived using a salt from a genuinely random number generator or another reliable source of randomness.

Each attestation technology specification offers guidance on replay protection using nonces and other techniques. Specific recommendations are deferred to these individual specifications in this document.

Regarding the use of Evidence in a CSR, the security considerations outlined in [I-D.ietf-lamps-csr-attestation] are pertinent to this specification.

9. Acknowledgments

We would like to thank Russ Housley, Thomas Fossati, Watson Ladd, Ionut Mihalcea, Carl Wallace, and Michael StJohns for their review comments.

10. References

10.1. Normative References

[I-D.ietf-lamps-csr-attestation]
Ounsworth, M., Tschofenig, H., Birkholz, H., Wiseman, M., and N. Smith, "Use of Remote Attestation with Certification Signing Requests", Work in Progress, Internet-Draft, draft-ietf-lamps-csr-attestation-21, 5 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-csr-attestation-21>.

[I-D.ietf-lamps-rfc4210bis]
Brockhaus, H., von Oheimb, D., Ounsworth, M., and J. Gray, "Internet X.509 Public Key Infrastructure -- Certificate Management Protocol (CMP)", Work in Progress, Internet-Draft, draft-ietf-lamps-rfc4210bis-18, 30 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-rfc4210bis-18>.

[I-D.ietf-lamps-rfc5272bis]
Mandel, J. and S. Turner, "Certificate Management over CMS (CMC)", Work in Progress, Internet-Draft, draft-ietf-lamps-rfc5272bis-08, 29 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-rfc5272bis-08>.

[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/rfc/rfc2119>.

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

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, DOI 10.17487/RFC5785, April 2010, <https://www.rfc-editor.org/rfc/rfc5785>.

[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/rfc/rfc7030>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[RFC8295] Turner, S., "EST (Enrollment over Secure Transport) Extensions", RFC 8295, DOI 10.17487/RFC8295, January 2018, <https://www.rfc-editor.org/rfc/rfc8295>.

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <https://www.rfc-editor.org/rfc/rfc8615>.

[RFC9482] Sahni, M., Ed. and S. Tripathi, Ed., "Constrained Application Protocol (CoAP) Transfer for the Certificate Management Protocol", RFC 9482, DOI 10.17487/RFC9482, November 2023, <https://www.rfc-editor.org/rfc/rfc9482>.

[X.680] ITU-T, "Information Technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T Recommendation X.680 , February 2021, <https://www.itu.int/rec/T-REC.X.680>.

[X.690] ITU-T, "Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690 , February 2021, <https://www.itu.int/rec/T-REC.X.690>.

10.2. Informative References

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

[RFC4211] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, DOI 10.17487/RFC4211, September 2005, <https://www.rfc-editor.org/rfc/rfc4211>.

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <https://www.rfc-editor.org/rfc/rfc9334>.

[RFC9711] Lundblade, L., Mandyam, G., O’Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", RFC 9711, DOI 10.17487/RFC9711, April 2025, <https://www.rfc-editor.org/rfc/rfc9711>.

[RFC9783] Tschofenig, H., Frost, S., Brossard, M., Shaw, A., and T. Fossati, "Arm’s Platform Security Architecture (PSA) Attestation Token", RFC 9783, DOI 10.17487/RFC9783, June 2025, <https://www.rfc-editor.org/rfc/rfc9783>.

[TPM20] Trusted Computing Group, "Trusted Platform Module Library Specification, Family 2.0, Level 00, Revision 01.59", November 2019, <https://trustedcomputinggroup.org/resource/tpm-library-specification/>.

Appendix A. ASN.1 Module

The following module adheres to ASN.1 specifications [X.680] and [X.690].

```
   <CODE BEGINS>

   att-fresh-req
     { iso(1) identified-organization(3) dod(6) internet(1)
     security(5) mechanisms(5) pkix(7) id-mod(0)
     id-mod-att-fresh-req (TBDMOD) }

   DEFINITIONS IMPLICIT TAGS ::=
   BEGIN
   EXPORTS ALL;
   IMPORTS

   id-it, InfoTypeAndValue{}
     FROM PKIXCMP-2023
       { iso(1) identified-organization(3) dod(6) internet(1)
         security(5) mechanisms(5) pkix(7) id-mod(0)
         id-mod-cmp2023-02(TBD-PKIXCMP-23) }
   -- RFC Editor: The value for id-mod-cmp2023-02 must be set as soon
   -- as it is assigned by I-D.ietf-lamps-rfc4210bis

   EVIDENCE-STATEMENT, EvidenceStatementSet
     FROM CSR-ATTESTATION-2023
       { iso(1) identified-organization(3) dod(6) internet(1) security(5)
         mechanisms(5) pkix(7) id-mod(0) id-mod-pkix-attest-01(TBD-CSR-ATTESTATION-2023)
}
   -- RFC Editor: The value for id-mod-pkix-attest-01 must be set as soon
   -- as it is assigned by I-D.ietf-lamps-csr-attestation

   CMC-CONTROL
   FROM EnrollmentMessageSyntax-2025
      { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-enrollMsgSyntax-2025(TBD1) }

   ;

   -- NonceRequest and NonceResponse messages

    id-it-nonceRequest OBJECT IDENTIFIER ::= { id-it TBD1 }
    NonceRequestValue ::= SEQUENCE SIZE (1..MAX) OF NonceRequest
    NonceRequest ::= SEQUENCE {
       len    INTEGER OPTIONAL,
       -- indicates the required length of the requested nonce
       type   EVIDENCE-STATEMENT.&id({EvidenceStatementSet}) OPTIONAL,
       -- indicates which Evidence type to request a nonce for
       hint   UTF8String OPTIONAL
       -- indicates which Verifier to request a nonce from
    }
```

```
    id-it-nonceResponse OBJECT IDENTIFIER ::= { id-it TBD2 }
    NonceResponseValue ::= SEQUENCE SIZE (1..MAX) OF NonceResponse
    NonceResponse ::= SEQUENCE {
       nonce  OCTET STRING,
       -- contains the nonce of length len
       -- provided by the Verifier indicated with hint
       expiry INTEGER OPTIONAL,
       -- indicates how long in seconds the Verifier considers
       -- the nonce valid
       type   EVIDENCE-STATEMENT.&id({EvidenceStatementSet}) OPTIONAL,
       -- indicates which Evidence type to request a nonce for
       hint UTF8String OPTIONAL
       -- indicates which Verifier to request a nonce from
    }

   id-cmc-nonceReq ::= { id-it TBD1 }

   cmc-nonceReq CMC-CONTROL ::=
         { NonceRequest IDENTIFIED BY id-cmc-nonceReq }

   id-cmc-nonceResp ::= { id-it TBD2 }

   cmc-nonceResp CMC-CONTROL ::=
         { NonceResponse IDENTIFIED BY id-cmc-nonceResp }

   END
   <CODE ENDS>
```

Authors' Addresses

Hannes Tschofenig
Siemens
Germany
Email: hannes.tschofenig@gmx.net

Hendrik Brockhaus
Siemens
Werner-von-Siemens-Strasse 1
80333 Munich
Germany
Email: hendrik.brockhaus@siemens.com
URI: https://www.siemens.com

Joe Mandel
AKAYLA, Inc.
Email: joe@akayla.com

Sean Turner
sn3rd
Email: sean@sn3rd.com