Workgroup: LAMPS Working Group
Internet-Draft:
draft-ietf-lamps-rfc7030-csrattrs-09
Updates: 7030 (if approved)
Published: 4 April 2024
Intended Status: Standards Track
Expires: 6 October 2024
Authors: M. Richardson, Ed. O. Friel D. von Oheimb
Sandelman Software Works Cisco Siemens
D. Harkins
The Industrial Lounge

Clarification of RFC7030 CSR Attributes definition

Abstract

The Enrollment over Secure Transport (EST, RFC7030) is ambiguous in its specification of the CSR Attributes Response. This has resulted in implementation challenges and implementor confusion.

This document updates RFC7030 (EST) and clarifies how the CSR Attributes Response can be used by an EST server to specify both CSR attribute OIDs and also CSR attribute values, in particular X.509 extension values, that the server expects the client to include in subsequent CSR request.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Enrollment over Secure Transport [RFC7030] (EST) has been used in a wide variety of applications. In particular, [RFC8994] and [RFC8995] describe a way to use it in order to build out an autonomic control plane (ACP) [RFC8368].

The ACP requires that each node be given a very specific subjectAltName. In the ACP specification, the solution was for the EST server to use section 2.6 of [RFC7030] to convey to the EST client the actual subjectAltName that will end up in its certificate.

As a result of some implementation challenges, it came to light that this particular way of using the CSR attributes was not universally agreed upon, and it was suggested that it went contrary to section 2.6.

Section 2.6 says that the CSR attributes "can provide additional descriptive information that the EST server cannot access itself". This is extended to describe how the EST server can provide values that it demands to use.

After significant discussion, it has been determined that Section 4.5 of [RFC7030] specification is sufficiently difficult to read and ambiguous to interpret that clarification is needed.

This document motivates the different use cases, and provides additional worked out examples.

Also, section 4.5.2 is extended to clarify the use of the existing ASN.1 syntax [X.680][X.690]. This covers all uses and is fully backward compatible with existing use.

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.

3. CSR Attributes Handling

3.1. Extensions to RFC 7030 section 2.6.

Replace the second paragraph with the following text:

These attributes can provide additional descriptive information that the EST server cannot access itself, such as the Media Access Control (MAC) address of an interface of the EST client. The EST server can also provide concrete values that it tells the client to include in the CSR, such as a specific X.509 Subject Alternative Name extension. Moreover, these attributes can indicate the type of the included public key or which crypto algorithms to use for the self-signature, such as a specific elliptic curve or a specific hash function that the client is expected to use when generating the CSR.

3.2. Extensions to RFC 7030 section 4.5.2.

The ASN.1 syntax for CSR Attributes as defined in EST section 4.5.2 is as follows:

```
CsrAttrs ::= SEQUENCE SIZE (0..MAX) OF AttrOrOID

AttrOrOID ::= CHOICE (oid OBJECT IDENTIFIER, attribute Attribute }

Attribute { ATTRIBUTE:IOSet } ::= SEQUENCE {
     type   ATTRIBUTE.&id({IOSet}),
     values SET SIZE(1..MAX) OF ATTRIBUTE.&Type({IOSet}{@type}) }
```

This remains unchanged, such that bits-on-the-wire compatibility is maintained.

Key parts that were unclear were which OID to use in the 'type' field and that the 'values' field can contain an entire sequence of X.509 extensions.

The OID to use for such attributes in the 'type' field MUST be extensionRequest, which has the numerical value 1.2.840.113549.1.9.14. There MUST be only one such Attribute.

The 'values' field of this attribute MUST contain a set with exactly one element, and this element MUST be of type Extensions, as per Section 4.1 of [RFC5280]:

```
Extensions  ::=  SEQUENCE SIZE (1..MAX) OF Extension

Extension  ::=  SEQUENCE  {
     extnID      OBJECT IDENTIFIER,
     critical    BOOLEAN DEFAULT FALSE,
     extnValue   OCTET STRING
                 -- contains the DER encoding of an ASN.1 value
                 -- corresponding to the extension type identified
                 -- by extnID
     }
```

An Extension comprises the OID of the specific X.509 extension (extnID), optionally the 'critical' bit, and the extension value (extnValue).

An Extensions structure, which is a sequence of elements of type Extension, MUST NOT include more than one element with a particiular extnID.

With this understanding, the needs of [RFC8994] and [RFC8995] are satisfied with no change to the bits on the wire.

3.3. Alternative: Use of CSR templates

[RFC8295], Appendix B suggests an alternative that avoids the piecemeal inclusion of attributes that [RFC7030] documented. Instead, an entire CSR object is returned with various fields filled out, and other fields waiting to be filled in, in a pKCS7PDU attribute. In the suggested approach, the pKCS7PDU attribute includes a Full PKI Data content type [RFC5272] and that in turn includes a CSR or CRMF formatted request; see [RFC6268] Sections 5 and 9, respectively.

The drawback to this approach, particularly for the CSR, is that some required fields are "faked"; specifically, the signature field on the CSR is faked with an empty bit string. To avoid this drawback, this specification defines the Certificate Request Information Template attribute for CsrAttrs, see Section 3.2, that is request minus the useless signature wrapper as follows:

```
aa-certificationRequestInfoTemplate ATTRIBUTE ::=
  { TYPE CertificationRequestInfoTemplate IDENTIFIED BY
    id-aa-certificationRequestInfoTemplate }

id-aa-certificationRequestInfoTemplate OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9)
    smime(16) aa(2) csrinfo(TBD2) }

CertificateRequestInfoTemplate ::= CertificationRequestInfo
```

The CertificationRequestInfoTemplate uses the CertificationRequestInfo from [RFC5912], Section 5 and is included here for convenience:

```
CertificationRequestInfo ::= SEQUENCE {
  version       INTEGER { v1(0) } (v1,...),
  subject       Name,
  subjectPKInfo SubjectPublicKeyInfo{{ PKInfoAlgorithms }},
  attributes    [0] Attributes{{ CRIAttributes }}
}
```

Note: This method has also been defined in CMP Updates [RFC9480] and the Lightweight CMP profile [RFC9483], Section 4.3.3, using a CSR template as defined for CRMF [RFC4211].

Legacy servers MAY continue to use the [RFC7030] style piecemeal attribute/value pairs, and MAY also include the template style described here. Clients which understand both MUST use the template only, and ignore all other CSRattrs elements. Older clients will ignore this new element.

The version code is always v1 (0). As shown in the example below, any empty values in the subject DN, and in any included X509v3 extensions are expected to be filled in by the client.

The SubjectPublicKeyInfo field MUST be present, but it MUST have an empty bit string for the key, as the server does not know what key will be used. The server MAY specify (in the OID), the type of the key to use, but otherwise the OID type MUST be NULL.

Each of the attributes has a single attribute value instance in the values set. Even though the syntax is defined as a set, there MUST be exactly one instance of AttributeValue present.

4. Co-existence with existing implementations

5. Examples

Each example has a high-level (English) explanation of what is expected. Some mapping back to the Attribute and Extension definitions above are included. The base64 DER encoding is then shown. The output of "dumpasn1" is then provided to detail what the contents are.

5.1. RFC8994/ACP subjectAltName with specific otherName

A single subjectAltName extension is specified in a single Extension attribute. This is what might be created by an [RFC8995] Registrar that is asking for [RFC8994] AcpNodeName format otherNames.

5.1.1. Base64 encoded example

The Base64:

```
MGQwYgYJKoZIhvcNAQkOMVUwUwYDVR0RAQH/BEmgRzBFBggr
BgEFBQcICgw5cmZjODk5NCtmZDczOWZjMjNjMzQ0MDExMjIz
MzQ0NTUwMDAwMDAwMCtAYWNwLmV4YW1wbGUuY29t
```

5.1.2. ASN.1 DUMP output

There is a single subjectAltName Extension with an Attribute with Extension type.

```
    <30 64>
  0 100: SEQUENCE {
    <30 62>
  2  98:   SEQUENCE {
    <06 09>
  4   9:     OBJECT IDENTIFIER extensionRequest (1 2 840 113549 1 9 14)
       :       (PKCS #9 via CRMF)
    <31 55>
 15  85:     SET {
    <30 53>
 17  83:       SEQUENCE {
    <06 03>
 19   3:         OBJECT IDENTIFIER subjectAltName (2 5 29 17)
       :           (X.509 extension)
    <01 01>
 24   1:         BOOLEAN TRUE
    <04 49>
 27  73:         OCTET STRING
       :           A0 47 30 45 06 08 2B 06    .G0E..+.
       :           01 05 05 07 08 0A 0C 39    .......9
       :           72 66 63 38 39 39 34 2B    rfc8994+
       :           66 64 37 33 39 66 63 32    fd739fc2
       :           33 63 33 34 34 30 31 31    3c344011
       :           32 32 33 33 34 34 35 35    22334455
       :           30 30 30 30 30 30 30 30    00000000
       :           2B 40 61 63 70 2E 65 78    +@acp.ex
       :           61 6D 70 6C 65 2E 63 6F    ample.co
       :           6D                         m
       :         }
       :       }
       :     }
       :   }
```

5.2. RFC7030 original example

In this example, taken from [RFC7030], a few different attributes are included.

5.2.1. Base64 encoded example

The Base64:

```
MEEGCSqGSIb3DQEJBzASBgcqhkjOPQIBMQcGBSuBBAAiMBYG
CSqGSIb3DQEJDjEJBgcrBgEBAQEWBggqhkjOPQQDAw==
```

5.2.2. ASN.1 DUMP output

1. The challengePassword attribute is included to indicate that the CSR should include this value.

2. An ecPublicKey attribute is provided with the value secp384r1 to indicate what kind of key should be submitted.

3. An extensionRequest container with an OID 1.3.6.1.1.1.1.22 (macAddress), but without a value, to indicate that the CSR should include an X.509v3 extension with this value.

4. The ecdsaWithSHA384 OID is included to indicate what kind of hash is expected to be used for the self-signature of the PCKS#10 CSR structure.

```
    <30 41>
  0  65: SEQUENCE {
    <06 09>
  2   9:   OBJECT IDENTIFIER challengePassword (1 2 840 113549 1 9 7)
       :     (PKCS #9)
    <30 12>
 13  18:   SEQUENCE {
    <06 07>
 15   7:     OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
       :       (ANSI X9.62 public key type)
    <31 07>
 24   7:     SET {
    <06 05>
 26   5:       OBJECT IDENTIFIER secp384r1 (1 3 132 0 34)
       :         (SECG (Certicom) named elliptic curve)
       :       }
       :     }
    <30 16>
 33  22:   SEQUENCE {
    <06 09>
 35   9:     OBJECT IDENTIFIER extensionRequest (1 2 840 113549 1 9 14)
       :       (PKCS #9 via CRMF)
    <31 09>
 46   9:     SET {
    <06 07>
 48   7:       OBJECT IDENTIFIER '1 3 6 1 1 1 1 22'
       :       }
       :     }
    <06 08>
 57   8:   OBJECT IDENTIFIER ecdsaWithSHA384 (1 2 840 10045 4 3 3)
       :     (ANSI X9.62 ECDSA algorithm with SHA384)
       :   }
```

5.3. EST server requires a specific subjectAltName extension

This example is the same as the previous one except that instead of the OID for a macAddress, a subjectAltName is specified as the only Extension element.

5.3.1. Base64 encoded example

The Base64:

```
MGYGCSqGSIb3DQEJBzASBgcqhkjOPQIBMQcGBSuBBAAiMDsG
CSqGSIb3DQEJDjEuMCwGA1UdEQEB/wQioCAwHgYIKwYBBQUH
CAoMEnBvdGF0b0BleGFtcGxlLmNvbQYIKoZIzj0EAwM=
```

5.3.2. ASN.1 DUMP output

1. The challengePassword attribute is included to indicate that the CSR should include this value.

2. An ecPublicKey attribute is provided with the value secp384r1 to indicate what kind of key should be submitted.

3. An extensionRequest container with a subjectAltName value containing the name potato@example.com

4. The ecdsaWithSHA384 OID is included to indicate what kind of hash is expected to be used for the self-signature of the PCKS#10 CSR structure.

```
    <30 66>
  0 102: SEQUENCE {
    <06 09>
  2   9:   OBJECT IDENTIFIER challengePassword (1 2 840 113549 1 9 7)
       :     (PKCS #9)
    <30 12>
 13  18:   SEQUENCE {
    <06 07>
 15   7:     OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
       :       (ANSI X9.62 public key type)
    <31 07>
 24   7:     SET {
    <06 05>
 26   5:       OBJECT IDENTIFIER secp384r1 (1 3 132 0 34)
       :         (SECG (Certicom) named elliptic curve)
       :       }
       :     }
    <30 3B>
 33  59:   SEQUENCE {
    <06 09>
 35   9:     OBJECT IDENTIFIER extensionRequest (1 2 840 113549 1 9 14)
       :       (PKCS #9 via CRMF)
    <31 2E>
 46  46:     SET {
    <30 2C>
 48  44:       SEQUENCE {
    <06 03>
 50   3:         OBJECT IDENTIFIER subjectAltName (2 5 29 17)
       :           (X.509 extension)
    <01 01>
 55   1:         BOOLEAN TRUE
    <04 22>
 58  34:         OCTET STRING
       :           A0 20 30 1E 06 08 2B 06    . 0...+.
       :           01 05 05 07 08 0A 0C 12    ........
       :           70 6F 74 61 74 6F 40 65    potato@e
       :           78 61 6D 70 6C 65 2E 63    xample.c
       :           6F 6D                      om
       :         }
       :       }
       :     }
    <06 08>
 94   8:   OBJECT IDENTIFIER ecdsaWithSHA384 (1 2 840 10045 4 3 3)
       :     (ANSI X9.62 ECDSA algorithm with SHA384)
       :   }
```

5.4. Require a public key of a specific size

The CSR requires a public key of a specific size

5.4.1. Base64 encoded example

The Base64:

```
MCkGCSqGSIb3DQEJBzARBgkqhkiG9w0BAQExBAICEAAGCSqG
SIb3DQEBCw==
```

5.4.2. ASN.1 DUMP output

1. Provide a CSR with an RSA key that's 4096 bits and sign it with sha256

```
    <30 29>
  0  41: SEQUENCE {
    <06 09>
  2   9:   OBJECT IDENTIFIER challengePassword (1 2 840 113549 1 9 7)
       :     (PKCS #9)
    <30 11>
 13  17:   SEQUENCE {
    <06 09>
 15   9:     OBJECT IDENTIFIER rsaEncryption (1 2 840 113549 1 1 1)
       :       (PKCS #1)
    <31 04>
 26   4:     SET {
    <02 02>
 28   2:       INTEGER 4096
       :       }
       :     }
    <06 09>
 32   9:   OBJECT IDENTIFIER sha256WithRSAEncryption
                            (1 2 840 113549 1 1 11)
       :     (PKCS #1)
       :   }
```

5.5. Require a public key of a specific curve

The CSR requires a public key with a specific curve

5.5.1. Base64 encoded example

The Base64:

```
MD0GCSqGSIb3DQEJBzASBgcqhkjOPQIBMQcGBSuBBAAiMBIGCSqGSIb3DQEJDjEF
BgNVBAUGCCqGSM49BAMD
```

5.5.2. ASN.1 DUMP output

Provide a CSR with an ECC key from p384, include your serial number, and sign it with sha384.

```
   <30 3D>
 0  61: SEQUENCE {
   <06 09>
 2   9:   OBJECT IDENTIFIER challengePassword (1 2 840 113549 1 9 7)
      :     (PKCS #9)
   <30 12>
13  18:   SEQUENCE {
   <06 07>
15   7:     OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
      :       (ANSI X9.62 public key type)
   <31 07>
24   7:     SET {
   <06 05>
26   5:       OBJECT IDENTIFIER secp384r1 (1 3 132 0 34)
      :         (SECG (Certicom) named elliptic curve)
      :       }
      :     }
   <30 12>
33  18:   SEQUENCE {
   <06 09>
35   9:     OBJECT IDENTIFIER extensionRequest (1 2 840 113549 1 9 14)
      :       (PKCS #9 via CRMF)
   <31 05>
46   5:     SET {
   <06 03>
48   3:       OBJECT IDENTIFIER serialNumber (2 5 4 5)
      :         (X.520 DN component)
      :       }
      :     }
   <06 08>
53   8:   OBJECT IDENTIFIER ecdsaWithSHA384 (1 2 840 10045 4 3 3)
      :     (ANSI X9.62 ECDSA algorithm with SHA384)
      :   }
```

5.6. Require a specific extension

The CSR is required to have an EC key, to include a serial number, a friendly name, favorite drink, and be signed with SHA512.

5.6.1. Base64 encoded example

The Base64:

MFQGCSqGSIb3DQEJBzASBgcqhkjOPQIBMQcGBSuBBAAjMCkG
CSqGSIb3DQEJDjEcBgNVBAUGCSqGSIb3DQEJFAYKCZImiZPy
LGQBBQYIKoZIzj0EAwQ=

5.6.2. ASN.1 DUMP output

Provide a CSR with an EC key from sha521, include your serial number, friendly name, and favorite drink, and sign it with sha512

```
   <30 54>
 0  84: SEQUENCE {
   <06 09>
 2   9:   OBJECT IDENTIFIER challengePassword (1 2 840 113549 1 9 7)
      :     (PKCS #9)
   <30 12>
13  18:   SEQUENCE {
   <06 07>
15   7:     OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
      :       (ANSI X9.62 public key type)
   <31 07>
24   7:     SET {
   <06 05>
26   5:       OBJECT IDENTIFIER secp521r1 (1 3 132 0 35)
      :         (SECG (Certicom) named elliptic curve)
      :       }
      :     }
   <30 29>
33  41:   SEQUENCE {
   <06 09>
35   9:     OBJECT IDENTIFIER extensionRequest (1 2 840 113549 1 9 14)
      :       (PKCS #9 via CRMF)
   <31 1C>
46  28:     SET {
   <06 03>
48   3:       OBJECT IDENTIFIER serialNumber (2 5 4 5)
      :         (X.520 DN component)
   <06 09>
53   9:       OBJECT IDENTIFIER
      :         friendlyName (for PKCS #12) (1 2 840 113549 1 9 20)
      :         (PKCS #9 via PKCS #12)
   <06 0A>
64  10:       OBJECT IDENTIFIER '0 9 2342 19200300 100 1 5'
      :       }
      :     }
   <06 08>
76   8:   OBJECT IDENTIFIER ecdsaWithSHA512 (1 2 840 10045 4 3 4)
      :     (ANSI X9.62 ECDSA algorithm with SHA512)
      :   }
```

6. Security Considerations

The security considerations from EST [RFC7030] section 6 are unchanged.

6.1. Identity and Privacy Considerations

An EST server may use this mechanism to instruct the EST client about the identities it should include in the CSR it sends as part of enrollment. The client may only be aware of its IDevID Subject, which includes a manufacturer serial number. The EST server can use this mechanism to tell the client to include a specific fully qualified domain name in the CSR in order to complete domain ownership proofs required by the CA. Additionally, the EST server may deem the manufacturer serial number in an IDevID as personally identifiable information, and may want to specify a new random opaque identifier that the pledge should use in its CSR. This may be desirable if the CA and EST server have different operators.

7. IANA Considerations

IANA is asked to allocate two new Object Identifiers:

*One (TBD1) from the SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0) registry for the ASN.1 module: id-mod-critemplate; see Appendix A, and

*One (TBD2) from the SMI Security for S/MIME Attributes (1.2.840.113549.1.9.16.2) registry for the Certification Request Information Template (csrinfo) attribute; see Section 3.3 and Appendix A.

8. Acknowledgements

Corey Bonnell crafted example02 using a different tool, and this helped debug other running code.

9. Changelog

10. References

10.1. Normative References

[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>.

[RFC5272] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC)", RFC 5272, DOI 10.17487/RFC5272, June 2008, <https://www.rfc-editor.org/rfc/rfc5272>.

[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>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/rfc/rfc5652>.

[RFC5911] Hoffman, P. and J. Schaad, "New ASN.1 Modules for Cryptographic Message Syntax (CMS) and S/MIME", RFC 5911, DOI 10.17487/RFC5911, June 2010, <https://www.rfc-editor.org/rfc/rfc5911>.

[RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <https://www.rfc-editor.org/rfc/rfc5912>.

[RFC6268] Schaad, J. and S. Turner, "Additional New ASN.1 Modules for the Cryptographic Message Syntax (CMS) and the Public Key Infrastructure Using X.509 (PKIX)", RFC 6268, DOI 10.17487/RFC6268, July 2011, <https://www.rfc-editor.org/rfc/rfc6268>.

[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>.

[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/rfc/rfc8174>.

[RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <https://www.rfc-editor.org/rfc/rfc8994>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

[X.680] ITU-T, "Information technology -- Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T Recommendation X.680, ISO/IEC 8824-1:2021, 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, ISO/IEC 8825-1:2021, February 2021, <https://www.itu.int/rec/T-REC-X.680>.

10.2. Informative References

[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>.

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

[RFC8368] Eckert, T., Ed. and M. Behringer, "Using an Autonomic Control Plane for Stable Connectivity of Network Operations, Administration, and Maintenance (OAM)", RFC 8368, DOI 10.17487/RFC8368, May 2018, <https://www.rfc-editor.org/rfc/rfc8368>.

[RFC9480] Brockhaus, H., von Oheimb, D., and J. Gray, "Certificate Management Protocol (CMP) Updates", RFC 9480, DOI 10.17487/RFC9480, November 2023, <https://www.rfc-editor.org/rfc/rfc9480>.

[RFC9483] Brockhaus, H., von Oheimb, D., and S. Fries, "Lightweight Certificate Management Protocol (CMP) Profile", RFC 9483, DOI 10.17487/RFC9483, November 2023, <https://www.rfc-editor.org/rfc/rfc9483>.

Appendix A. ASN.1 Module

RFC EDITOR: Please replace TBD1 and TBD2 with the value assigned by IANA during the publication of [I-D.ietf-lamps-rfc7030-csrattrs].

This appendix provides an ASN.1 module [X.680] for the Certification Request Information Template attribute, and it follows the conventions established in [RFC5911], [RFC5912], and [RFC6268].

```
CRITemplateModule
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
    pkcs-9(9) smime(16) modules(0) id-mod-critemplate(TBD1) }

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

ATTRIBUTE -- [RFC5911]
 FROM PKIX-CommonTypes-2009
   { iso(1) identified-organization(3) dod(6) internet(1)
     security(5) mechanisms(5) pkix(7)
     id-mod(0) id-mod-pkixCommon-02(57) }

CertificationRequestInfo -- [RFC5912]
  FROM PKCS-10
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7)
      id-mod(0) id-mod-pkcs10-2009(69) }

;

aa-certificationRequestInfoTemplate ATTRIBUTE ::=
  { TYPE CertificationRequestInfoTemplate IDENTIFIED BY
    id-aa-certificationRequestInfoTemplate }

id-aa-certificationRequestInfoTemplate OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9)
    smime(16) aa(2) csrinfo(TBD2) }

CertificationRequestInfoTemplate ::= CertificationRequestInfo

END
```

Authors' Addresses

Michael Richardson (editor)
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

Owen Friel
Cisco

Email: ofriel@cisco.com

Dr. David von Oheimb
Siemens

Email: dev@ddvo.net

Dan Harkins
The Industrial Lounge

Email: dharkins@lounge.org