Workgroup: LAMPS Working Group
Internet-Draft:
draft-richardson-lamps-rfc7030-csrattrs-02
Published: 7 March 2022
Intended Status: Standards Track
Expires: 8 September 2022
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

Enrollment over Secure Transport (EST) is ambiguous in specification of the CSR Attributes Response. This has resulted in implementation challenges and implementor confusion. This document updates 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 that the server expects the client to include in its 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 in fact runs contrary to section 2.6. Section 2.6 says that the CSR attributes "provide additional descriptive information that the EST server cannot access itself". This extends to specifying that a particular attribute should exist, but not to the point of having the EST server actually specify the value.

The way in which the CSRattributes were understood by [RFC8994] turns out to be invalid. This document, therefore, updates section 2.6 to define this behavior.

This document also updates section 4.5 to include revised ASN.1 that covers all uses and is backward compatible with the existing use.

Additional examples are provided in an appendix.

2. CSR Attributes Handling

2.1. Current EST Specification

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

```
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}) }
```

That section also states the following:

the values indicating the particular
attributes desired to be included in the resulting certificate's
extensions

This has been interpreted by some implementations as meaning that the CSR Attributes response can only include values for the attribute OIDs that the client should include in its CSR, and cannot include the actual values of those attributes. This is further reinforced by the example:

```
Attribute:  type = extensionRequest (1.2.840.113549.1.9.14)
                   value = macAddress (1.3.6.1.1.1.1.22)
```

This example illustrates that the 'value' specified is an attribute OID, for example the macAddress OID, and not the value (such as "10-00-00-12-23-45") of the attribute itself.

There is no clearly documented mechanism with supporting examples that specifies how a CSR Attributes response can include a value for a given attribute such as SubjectAltName.

EST section 4.5.2 also states the following:

The structure of the CSR Attributes Response SHOULD, to the greatest extent possible, reflect the structure of the CSR it is requesting.

This statement aligns closely with the goal of this document. Additionally, EST Extensions [RFC8295] Appendix A has an informative appendix that outlines how a full CSR can be included in the CSR Attributes response.

3. Updated CSR Attributes Handling

The WG will pick one option as part of the adoption call.

3.1. Option two: Extend CSR structure to allow values:

This ASN.1 needs fixing.

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

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

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

Value { ATTRIBUTE:IOSet } ::= SEQUENCE {
     extType  ATTRIBUTE.&id({IOSet}),
     type     ATTRIBUTE.&Type({IOSet}{@type}),
     value    OCTET STRING
}
```

This would just add a value to the SEQUENCE:

```
  OBJECT challengePassword
  SEQUENCE
    OBJECT subjectAltName
    SET
      OBJECT someACPgoo
  SEQUENCE
    OBJECT id-ecPublicKey
    SET
      OBJECT secp384r1
      OBJECT ecdsa-with-SHA384
```

For example:

```
 0  30: SEQUENCE {
 2  28:   SEQUENCE {
 4   3:     OBJECT IDENTIFIER subjectAltName (2 5 29 17)
 9  21:     SET {
11  19:       [1] {
13  17:         UTF8String 'hello@example.com'
      :         }
      :       }
      :     }
      :   }
```

3.2. Option three: explicit content for the key specification

The following options support complete and unambiguous specification of

*CSR ingredients optionally including values to use,

*the type of the public key, which is given in the form of a public-key algorithm,

*and the hash algorithm to use for the self-signature.

CSR ingredients may be the subject DN, any X.509 extensions, and special attributes like a challenge password.

For specifying the type of keys allowed in CSRs, they use a to-the-point KeySpec type. It can be defined for instance as

```
KeySpec ::= CHOICE {
                keyAlg AlgorithmIdentifier,
                rsaKeyLen INTEGER
}
```

The keyAlg type use used to specify public-key alorithms and can include parameters, such as the name of an elliptic curve. The rsaKeyLen choice allows specifying the size of RSA keys, which it is not possible using values of type AlgorithmIdentifier.

The keySpec could also be sequence of such specs, such that the server can give several key types from which the client can choose, e.g., EC keys on certain curves and/or RSA keys of certain sizes.

Stick for syntactic backward compatibility with

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

Each OID given in AttrOrOID must occur only once.

Plain OIDs are used mostly for challengePassword.

Attributes are used mostly for any X.509 extensions, subject DN, key spec, and hash alg, while defining new generally usable OIDs for

*a subject DN of type Name

*a key spec of type KeySpec

*a hash alg spec of type AlgorithmIdentifier

to be given on demand as attribute IDs of type ATTRIBUTE.&id({IOSet}).

3.3. Option four: explicit members for unique attributes

Define a new and more to-the-point type, which does not require new OIDs:

```
CsrAttrs ::= SEQUENCE {
     oids      SEQUENCE OF OBJECT IDENTIFIER,
     attrs     SEQUENCE OF Attribute,
     subject   [0] Name OPTIONAL,
     keySpec   [1] KeySpec OPTIONAL,
     hashAlg   [2] AlgorithmIdentifier OPTIONAL

}
```

Each OID given in oids or attrs must occur only once.

The oids are used mostly for requiring a challenge password.

The atttrs are used mostly for requiring certain X.509 extensions.

This is, typically just challengePassword and extensionRequest are used.

3.4. Option five: more specific structure, simpler extensions

Define a new fully to-the-point type, which does not require any (direct) OIDs:

```
CsrAttrs ::= SEQUENCE {
     subject                Name OPTIONAL,
     extensions             SEQUENCE OF Extension,
     challengePassword      BOOLEAN,
     keySpec            [0] KeySpec OPTIONAL,
     hashAlg            [1] AlgorithmIdentifier OPTIONAL

}
```

4. Co-existence with existing implementations

There are some ways in which the new CSRattributes could co-exist with RFC7030.

4.1. Use a new MIME type

The client can signal that it supports the new attribute format by using an Accept: header in the transaction. This acts as a signal to a server that it can/should return the attributes in the new format.

4.2. Use a new end point of the new format

Clients that want to use the new format would use a new end point, such as "csrvalues" which would only support the new format. A client which supported both would have to try both "csrvalues" and then fall back "csrattrs" if the EST server did not support the new format. Some uses (such as [RFC8994]) require the new format, so if it was not suppored, that would be a protocol error.

4.3. Insist new format is upwardly compatible with old format

ASN.1 encoding is self-describing, and some formats proposed above could possibly be parsed by legacy clients without a problem.

4.4. Return new format to new clients only

The Registrar may know which clients are which by the kind of authentication that they do. An [RFC8994] client which has just performed a [RFC8995] enrollment would be assumed to require the new format only. A client which authenticates with an LDevID for a renewal would be strongly identified, and the Registrar could be programmed whether to return new format, or legacy CSR attributes.

5. Whether or not to Base64 encoding of results

[RFC8951] clarified that the csrattrs end point was to be Base64 encoded even though the HTTP transport was 8-bit clean.

If this document establishes a new end point, then the new end point will not be base64 encoded according to current HTTP usage.

6. Examples

6.1. RFC8994/ACP subjectAltName with specific otherName included

TBD

6.2. EST server requires public keys of a specific size

TBD

6.3. EST server requires a public key of a specific algorithm/curve

TBD

6.4. EST server requires a specific extension to be present

TBD

7. Security Considerations

All security considertions from EST [RFC7030] section 6 are applicable.

7.1. 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.

8. IANA Considerations

None.

9. Acknowledgements

TODO

10. Changelog

11. References

11.1. Normative References

[BCP14] 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>.

[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>.

[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/info/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/info/rfc8995>.

11.2. Informative References

[RFC8295] Turner, S., "EST (Enrollment over Secure Transport) Extensions", RFC 8295, DOI 10.17487/RFC8295, January 2018, <https://www.rfc-editor.org/info/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/info/rfc8368>.

[RFC8951] Richardson, M., Werner, T., and W. Pan, "Clarification of Enrollment over Secure Transport (EST): Transfer Encodings and ASN.1", RFC 8951, DOI 10.17487/RFC8951, November 2020, <https://www.rfc-editor.org/info/rfc8951>.

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