LAMPS M. StJohns

Internet-Draft NthPermutation Security LLC

Intended status: Standards Track 7 June 2023

Expires: 9 December 2023

Use of Attestation with Certification Signing Requests

draft-stjohns-csr-attest-02

Abstract

Utilizing information from a device or hardware security module about its posture can help to improve security of the overall system. Information about the manufacturer of the hardware, the version of the firmware running on this hardware and potentially about the layers of software above the firmware, the presence of hardware security functionality to protect keys and many more properties can be made available to remote parties in a cryptographically secured way. This functionality is accomplished with attestation technology.

This document describes extensions to encode evidence produced by an attester for inclusion in PKCS10 certificate signing requests. More specifically, two new ASN.1 Attribute definitions, and an ASN.1 CLASS definition to convey attestation information to a Registration Authority or to a Certification Authority are described.

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 9 December 2023.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

Provisions Relating to IETF Documents (https://trustee.ietf.org/

license-info) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights

and restrictions with respect to this document. Code Components

extracted from this document must include Revised BSD License text as

described in Section 4.e of the Trust Legal Provisions and are

provided without warranty as described in the Revised BSD License.

StJohns Expires 9 December 2023 [Page 1]

Internet-Draft CSR Attestation Attributes June 2023

Table of Contents

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . 2

1.1. Requirements Language . . . . . . . . . . . . . . . . . . 3

2. Definitions . . . . . . . . . . . . . . . . . . . . . . . . . 3

3. ASN.1 Elements . . . . . . . . . . . . . . . . . . . . . . . 4

3.1. Object Identifiers . . . . . . . . . . . . . . . . . . . 4

3.2. CertificateChoice . . . . . . . . . . . . . . . . . . . . 4

3.3. AttestAttribute . . . . . . . . . . . . . . . . . . . . . 6

3.4. AttestCertsAttribute . . . . . . . . . . . . . . . . . . 6

3.5. AttestStatement . . . . . . . . . . . . . . . . . . . . . 6

4. IANA Considerations . . . . . . . . . . . . . . . . . . . . . 8

4.1. Object Identifier Allocations . . . . . . . . . . . . . . 8

4.1.1. Module Registration - SMI Security for PKIX Module

Identifer . . . . . . . . . . . . . . . . . . . . . . 8

4.1.2. Object Identifier Registrations - SMI Security for S/

MIME Attributes . . . . . . . . . . . . . . . . . . . 8

4.2. "SMI Security for PKIX Attestation Statement Formats"

Registry . . . . . . . . . . . . . . . . . . . . . . . . 9

5. Security Considerations . . . . . . . . . . . . . . . . . . . 9

6. References . . . . . . . . . . . . . . . . . . . . . . . . . 10

6.1. Normative References . . . . . . . . . . . . . . . . . . 10

6.2. Informative References . . . . . . . . . . . . . . . . . 10

Appendix A. Examples . . . . . . . . . . . . . . . . . . . . . . 11

A.1. Simple Attestation Example . . . . . . . . . . . . . . . 11

A.2. Example TPM V2.0 Attestation Attribute - Non Normative . 11

Appendix B. ASN.1 Module for Attestation . . . . . . . . . . . . 13

Acknowledgements . . . . . . . . . . . . . . . . . . . . . . . . 17

Author's Address . . . . . . . . . . . . . . . . . . . . . . . . 17

1. Introduction

Remote attestation allows a relying party, the Registration Authority or the Certification Authority, to learn about the security posture of a verifier, which in the context of this specification is a device transmitting a certificate signing request using the newly defined attestation extension.

As outlined in RFC 9334, a verifier collects claims from its target environment and gets those claims signed by the attesting environment. The details of what claims are collected, how they are signed and what formats for serialization are used vary with a given attestation technology. At the time of writing several standardized and proprietary attestation technologies are in use. This specification thereby tries to be technology agnostic with regards to the transport of the produced signed claims.

RFC 9334 uses the term “evidence” for the information that is communicated by the verifier with remote parties. Since the information produced by the device can neither be trusted and often not even verified directly by the relying party an additional role with the verifier is introduced. The verifier has knowledge to verify the evidence, has information about what devices run what firmware and software, etc.

This document creates two ATTRIBUTE/Attribute definitions. The first Attribute may be used to carry a set of certificates or public keys that may be necessary to validate an evidence. The second Attribute carries a structure that may be used to carry key attestation statements, signatures and related data.

With these extensions a Certification Authority (CA) has additional information about whether to issuer a certificate and what information to populate into the certificate.

2. Requirements Language and Terminology

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].

This document re-uses the terms defined in RFC 9334 related to remote attestation. Readers of this document are assumed to be familiar with the following terms: evidence, claim, attestation result, attester, verifier, and relying party.

3. Architecture

Figure 1 shows the high-level communication pattern of the passport model where the attester transmits the evidence in the CSR to the RA and the CA. The verifier processes the received evidence and computes an attestation result, which is the processed by the RA/CA prior to the certificate issuance. Note that the verifier is a logical role that may be included in the RA/CA product. In this case the interaction between the relying party and the verifier are local.

.-------------.

| | Compare Evidence

| Verifier | against

| | policy

'--------+----'

^ |

Evidence | | Attestation

| | Result

| v

.------------. .----|----------.

| +-------------->|---' | Compare Attestation

| Attester | Evidence | Relying | Result against

| | in CSR | Party (RA/CA) | policy

'------------' '---------------'

Figure 1: Communication Pattern

As discussed in RFC 9334 different security and privacy aspects need to be considered. For example, evidence may need to be protected against replay and Section 10 of RFC 9334 lists approach for offering freshness. There are also concerns about the exposure of persistent identifiers by utilizing attestation technology, which are discussed in Section 11 of RFC 9334. Finally, the keying material used by the attester need to be protected against unauthorized access. This aspect is described in Section 12 of RFC 9334. Most of these aspects are, however, outside the scope of this specification but relevant for use with a given attestation technology. The focus of this specification is on the transport of evidence from the attester to the relying party via existing certificate signing request (CSR) messages.

3. ASN.1 Elements

3.1. Object Identifiers

Placeholder for now, waiting on guidance.

-- Root of IETF's PKIX OID tree

id-pkix OBJECT IDENTIFIER ::= { iso(1) identified-organization(3)

dod(6) internet(1) security(5) mechanisms(5) pkix(7) }

-- S/Mime attributes - can be used here.

id-aa OBJECT IDENTIFIER ::= {iso(1) member-body(2) usa(840)

rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) attributes(2)}

-- Branch for attestation statement types

id-ata OBJECT IDENTIFIER ::= { id-pkix (TBD1) }

3.2. CertificateChoice

This is an ASN.1 CHOICE construct used to represent an encoding of a

broad variety of certificate types.

StJohns Expires 9 December 2023 [Page 4]

Internet-Draft CSR Attestation Attributes June 2023

CertificateChoice ::=

CHOICE {

cert Certificate, -- typical X.509 cert

opaqueCert [0] IMPLICIT OCTET STRING, -- Format implicitly agreed upon

-- by sender and receiver

typedCert [1] IMPLICIT TypedCert,

typedFlatCert [2] IMPLICIT TypedFlatCert

}

"Certificate" is a standard X.509 certificate that MUST be compliant

with RFC5280. Enforcement of this constraint is left to the relying

parties.

"opaqueCert" should be used sparingly as it requires the receiving

party to implictly know its format. It is encoded as an OCTET

STRING.

"TypedCert" is an ASN.1 construct that has the charateristics of a

certificate, but is not encoded as an X.509 certificate. The

certTypeField indicates how to interpret the certBody field. While

it is possible to carry any type of data in this structure, it's

intended the content field include data for at least one public key

formatted as a SubjectPublicKeyInfo (see [RFC5912]).

TYPED-CERT ::= TYPE-IDENTIFIER -- basically an object id and a matching ASN1

-- structure encoded as a sequence

CertType ::= TYPED-CERT.&id

TypedCert ::= SEQUENCE {

certType TYPED-CERT.&id({TypedCertSet}),

content TYPED-CERT.&Type ({TypedCertSet}{@certType})

}

TypedCertSet TYPED-CERT ::= {

... -- Empty for now,

}

"TypedFlatCert" is a certificate that does not have a valid ASN.1

encoding. Think compact or implicit certificates as might be used

with smart cards. certType indicates the format of the data in the

certBody field, and ideally refers to a published specification.

TypedFlatCert ::= SEQUENCE {

certType OBJECT IDENTIFIER,

certBody OCTET STRING

}

StJohns Expires 9 December 2023 [Page 5]

Internet-Draft CSR Attestation Attributes June 2023

3.3. AttestAttribute

By definition, Attributes within a Certification Signing Request are

typed as ATTRIBUTE. This attribute definition contains one or more

attestation statements of a type "AttestStatement".

id-aa-attestStatement OBJECT IDENTIFIER ::= { id-aa (TBDAA2) }

AttestAttribute ATTRIBUTE ::= {

TYPE AttestStatement

IDENTIFIED BY id-aa-attestStatement

}

3.4. AttestCertsAttribute

The "AttestCertsAttribute" contains a sequence of certificates that

may be needed to validate the contents of an attestation statement

contained in an attestAttribute. By convention, the first element of

the SEQUENCE SHOULD contain the object that contains the public key

needed to directly validate the attestAttribute. The remaining

elements should chain that data back to an agreed upon root of trust

for the attestation.

id-aa-attestChainCerts OBJECT IDENTIFIER ::= { id-aa (TBDAA1) }

attestCertsAttribute ATTRIBUTE ::= {

TYPE SEQUENCE OF CertificateChoice

COUNTS MAX 1

IDENTIFIED BY id-aa-attestChainCerts

}

3.5. AttestStatement

An AttestStatement is an object of class ATTEST-STATEMENT encoded as

a sequence fields, of which the type of the "value" field is

controlled by the value of the "type" field, similar to an Attribute

definition.

StJohns Expires 9 December 2023 [Page 6]

Internet-Draft CSR Attestation Attributes June 2023

ATTEST-STATEMENT ::= CLASS {

&id OBJECT IDENTIFIER UNIQUE,

&Type, -- NOT optional

&algidPresent ParamOptions DEFAULT absent,

&sigPresent ParamOptions DEFAULT absent,

&ancillaryPresent ParamOptions DEFAULT absent,

&sigType DEFAULT OCTET STRING

&ancillaryType DEFAULT OCTET STRING

} WITH SYNTAX {

TYPE &Type

IDENTIFIED BY &id

[ALGID IS &algidPresent]

[SIGNATURE [TYPE &sigType] IS &sigPresent]

[ANCILLARY [TYPE &ancillaryType] IS &ancillaryPresent]

}

AttestStatement { ATTEST-STATEMENT:IOSet} ::= SEQUENCE

{

type ATTEST-STATEMENT.&id({IOSet}),

value ATTEST-STATEMENT.&Type({IOSet}{@type}),

algId [0] IMPLICIT AlgorithmIdentifier OPTIONAL,

signature [1] ATTEST-STATEMENT.&sigType OPTIONAL -- NOT implicit

ancillaryData [2] ATTEST-STATEMENT.&ancillaryType OPTIONAL

}

Depending on whether the "value" field contains an entire signed

attestation, or only the toBeSigned portion, the algId field may or

may not be present. If present it contains the AlgorithmIdentifier

of the signature algorithm used to sign the attestation statement.

If absent, either the value field contains an indication of the

signature algorithm, or the signature algorithm is fixed for that

specific type of AttestStatement.

Similarly for the "signature" field, if the "value" field contains

only the toBeSigned portion of the attestation statement, this field

SHOULD be present. The "signature" field may by typed as any valid

ASN.1 type. Opaque signature types SHOULD specify the use of sub-

typed OCTET STRING. For example:

MyOpaqueSignature ::= OCTET STRING

If possible, the ATTEST-STATEMENT SHOULD specify an un-wrapped

representation of a signature, rather than an OCTET STRING or BIT

STRING wrapped ASN.1 structure. I.e., by specifying ECDSA-Sig-Value

from PKIXAlgs-2009 (see [RFC5912]) to encode an ECDSA signature.

StJohns Expires 9 December 2023 [Page 7]

Internet-Draft CSR Attestation Attributes June 2023

ECDSA-Sig-Value ::= SEQUENCE {

r INTEGER,

s INTEGER

}

The ancillaryData field contains data provided externally to the

attestation engine,and/or data that may be needed to relate the

attestation to other PKIX elements. The format or content of the

externally provided data is not under the control of the attestation

engine. For example, this field might contain a freshness nonce

generated by the relying party, a signed time stamp, or even a hash

of protocol data or nonce data. See below for a few different

examples.

4. IANA Considerations

The IANA is requested to open one new registrie, allocate a value

from the "SMI Security for PKIX Module Identifier" registry for the

included ASN.1 module, and allocate values from "SMI Security for S/

MIME Attributes" to identify two Attributes defined within.

4.1. Object Identifier Allocations

4.1.1. Module Registration - SMI Security for PKIX Module Identifer

\* Decimal: IANA Assigned - Replace TBDMOD

\* Description: Attest-2023 - id-mod-pkix-attest-01

\* References: This Document

4.1.2. Object Identifier Registrations - SMI Security for S/MIME

Attributes

Attest Statement

\* Decimal: IANA Assigned - Replace TBDAA2

\* Description: id-aa-attestStatement

\* References: This Document

Attest Certificate Chain

\* Decimal: IANA Assigned - Replace TBDAA1

\* Description: id-aa-attestChainCerts

\* References: This Document

StJohns Expires 9 December 2023 [Page 8]

Internet-Draft CSR Attestation Attributes June 2023

4.2. "SMI Security for PKIX Attestation Statement Formats" Registry

Please open up a registry for Attestation Statement Formats within

the SMI-numbers registry, allocating an assignment from id-pkix ("SMI

Security for PKIX" Registry) for the purpose.

\* Decimal: IANA Assigned - replace TBD1

\* Description: id-ata

\* References: This document

\* Initial contents: None

\* Registration Regime: Specification Required. Document must

specify an ATTEST-STATEMENT definition to which this Object

Identifier shall be bound.

Columns:

\* Decimal: The subcomponent under id-ata

\* Description: Begins with id-ata

\* References: RFC or other document

5. Security Considerations

The attributes and structures defined in this document are primarily

meant to be used as additional Attributes for a PKCS10 Certification

Signing Request (CSR). As such, it's up to the receiving/relying

party to place as much or as little trust in the contents of these

attributes as necessary to satisfy its own policies.

A relying party will need either a specification defining how an

attestation type was formed and how to validate that type, or a

trusted method of verifying the attestation. In the former case, a

relying party should consider the information available from any

certificate chain covering the attesting key when deciding to accept

the attestation.

Most attestations will need to provide a method to convert the

attested key representation into the equivalent SubjectPublicKey info

structure and the attested key MUST be compared for equivalence to

the public key provided in the CSR before accepting the attestation.

The relying party, as always, is responsible for setting the rules

for what it will accept. The presence of an AttestAttribute is not

required by any current standard, but such attribute may provide the

relying party with additional assurance as a prerequisite to issuing

certificates or other credentials. That acceptance criteria is out

of scope for this document. Whether to require an AttestAttribute or

its contents in any specific use case is out-of-scope for this

document.

StJohns Expires 9 December 2023 [Page 9]

Internet-Draft CSR Attestation Attributes June 2023

6. References

6.1. 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/info/rfc2119>.

6.2. Informative References

[EQMV] Certicom Research, "Elliptic Curve Qu-Vanstone Implicit

Certificate Scheme (ECQV)", Standards for Efficient

Cryptography SEC-4, January 2013,

<https://www.secg.org/sec4-1.0.pdf>.

[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/info/rfc5912>.

[RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/

Multipurpose Internet Mail Extensions (S/MIME) Version 4.0

Message Specification", RFC 8551, DOI 10.17487/RFC8551,

April 2019, <https://www.rfc-editor.org/info/rfc8551>.

[TPM20] Trusted Computing Group, "Trusted Platform Module Library

- Part 1: Architecture", TPM 2.0 Module Library Part1,

00-01.59, November 2019.

Trusted Computing Group, "Trusted Platform Module Library

- Part 2: Structures", TPM 2.0 Module Library Part2,

00-01.59, November 2019.

Trusted Computing Group, "Trusted Platform Module Library

- Part 2: Commands", TPM 2.0 Module Library Part2,

00-01.59, November 2019.

[TR-03110-3]

Federal Office for Information Security, "Advanced

Security Mechanisms for Machine Readable Travel Documents

and eIDAS Token - Part 3 Common Specifications V2.21",

Federal Republic of Germany, Technical Guideline TR-03110,

December 2016, <https://www.bsi.bund.de/SharedDocs/Downloa

ds/EN/BSI/Publications/TechGuidelines/TR03110/BSI\_TR-

03110\_Part-3-V2\_2.pdf?\_\_blob=publicationFile&v=1>.

StJohns Expires 9 December 2023 [Page 10]

Internet-Draft CSR Attestation Attributes June 2023

Appendix A. Examples

A.1. Simple Attestation Example

This is a fragment of ASN.1 meant to demonstrate an absolute minimal

definition of an ATTEST-STATEMENT. A similar fragment could be used

to define an ATTEST-STATEMENT for an opaque HSM vendor specific

atterstation model.

-- This OCTET STRING is not like any other OCTET STRING

-- Please see https://example.com/simple-attest.txt,

-- Structure labled "Mike's simple attest" for the

-- structure of this field and how to verify the attestation

MikesSimpleAttestData ::= OCTET STRING

mikesSimpleAttestOid OBJECT IDENTIFIER ::= { id-mikes-root 1 }

MikesSimpleAttest ATTEST-STATEMENT ::= {

TYPE MikesSimpleAttestData

IDENTIFIED BY mikesSimpleAttestOid

-- These are all implied

-- ALGID IS absent

-- SIGNATURE is absent

-- ANCILLARY is absent

}

A.2. Example TPM V2.0 Attestation Attribute - Non Normative

What follows is a fragment of an ASN.1 module that might be used to

define an attestation statment attribute to carry a TPM V2.0 key

attestation - i.e., the output of the TPM2\_Certify command. This is

an example and NOT a registered definition. It's provided simply to

give an example of how to write an ATTEST-STATEMENT definition and

module.

-- IMPORT these.

-- PKI normal form for an ECDSA signature

ECDSA-Sig-Value ::= SEQUENCE {

r INTEGER,

s INTEGER

}

-- Octet string of size n/8 where n is the

-- bit size of the public modulus

RSASignature ::= OCTET STRING

-- One or the other of these depending on the value in TPMT\_SIGNATURE

StJohns Expires 9 December 2023 [Page 11]

Internet-Draft CSR Attestation Attributes June 2023

TpmSignature CHOICE ::= {

ecSig [0] IMPLICIT ECDSA-Sig-Value,

rsaSig [1] IMPLICIT RSASignature

}

-- The TPM form of the public key being attested.

-- Needed to verify the attestation - this is the TPMT\_PUBLIC structure.

TpmtPublic ::= OCTET STRING

-- The TPMS\_ATTEST structure as defined in TPM2.0

-- Unwrapped from the TPM2B\_ATTEST provided

-- by the TPM2\_Certify command.

TpmsAttest ::= OCTET STRING

-- The qualifying data provided to a TPM2\_Certify call, may be absent

-- This is the contents of data field of the TPM2B\_DATA structure.

QualifyingData ::= OCTET STRING

TpmAncillary ::= SEQUENCE {

toBeAttestedPublic TpmtPublic,

qualifyingData QualifyingData OPTIONAL

}

-- This represents a maximally unwrapped TPM V2.0 attestation. The

-- output of TPM2\_Certify is a TPM2B\_ATTEST and a TPMT\_SIGNATURE.

-- The former is unwrapped into a TPMS\_ATTEST and the latter is

-- decomposed to provide the contents of the algId and signature fields.

--

-- This attestation statement can be verified by:

-- Signature siggy = Signature.getInstance (stmt.algId);

-- siggy.init (attestPublicKey, VERIFY);

-- siggy.update ((short)stmt.value.length) // todo: big or little endian

-- siggy.update (stmt.value.data)

-- bool verified = siggy.verify (getSigData(stmt.signature)); //

unwrap the signature

--

TpmV2AttestStatement ATTEST STATEMENT ::= {

TYPE TpmsAttest

IDENTIFIED BY id-ata-tpmv20-1

ALGID IS present

SIGNATURE TYPE TpmSignature IS present

ANCILLARY TYPE TpmAncillary IS present

}

This attestation is the result of executing a TPM2\_Certify command

over a TPM key. See [TPM20] for more details.

StJohns Expires 9 December 2023 [Page 12]

Internet-Draft CSR Attestation Attributes June 2023

The data portion of the value field encoded as OCTET STRING is the

attestationData from the TPM2B\_ATTEST produced by the TPM. In other

words, strip off the TPM2B\_ATTEST "size" field and place the

TPMS\_ATTEST encoded structure in the OCTET STRING data field.

The algId is derived from the "sigAlg" field of the TPMT\_SIGNATURE

structure.

The signature field is a TpmSignature, created by transforming the

TPMU\_SIGNATURE field to the appropriate structure given the signature

type.

The ancillary field contains a structure with the TPMT\_PUBLIC

structure that contains the TPM's format of the key to be attested.

The attestation statement data contains a hash of this structure, and

not the key itself, so the hash of this structure needs to be

compared to the value in the attestation attestation statement. If

that passes, the key needs to be transformed into a PKIX style key

and compared to the key in the certificate signing request to

complete the attestation verification.

The ancillary field also contains an optional OCTET STRING which is

used if the TPM2\_Certify command is called with a non-zero length

"qualifyingData" argument to contain that data.

An AttestCertChain attribute MUST be present if this attribute is

used as part of a certificate signing request.

Appendix B. ASN.1 Module for Attestation

The following module imports definitions from modules defined in

[RFC5912] and [RFC8551].

IANA Note: Please replace TBDMOD, TBD1 and TBD2 with assigned values.

-- This module provides a definition for two attributes thay may be

-- used to carry key attestation information within a

-- CertificationSigningRequest (aka PKCS10), or for other purposes.

-- IANA - Value needed

Attest-2023

{iso(1) identified-organization(3) dod(6) internet(1) security(5)

mechanisms(5) pkix(7) id-mod(0) id-mod-pkix-attest-01(TBDMOD) }

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

StJohns Expires 9 December 2023 [Page 13]

Internet-Draft CSR Attestation Attributes June 2023

IMPORTS

Attribute, SingleAttribute, id-pkix, Certificate

FROM PKIX1Explicit-2009

{iso(1) identified-organization(3) dod(6) internet(1)

security(5) mechanisms(5) pkix(7) id-mod(0)

id-mod-pkix1-explicit-02(51)}

ATTRIBUTE,AttributeSet

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)}

ParamChoice

FROM AlgorithmInformation-2009

{iso(1) identified-organization(3) dod(6) internet(1) security(5)

mechanisms(5) pkix(7) id-mod(0)

id-mod-algorithmInformation-02(58)}

id-aa

FROM SecureMimeMessageV3dot1

{ iso(1) member-body(2) us(840) rsadsi(113549)

pkcs(1) pkcs-9(9) smime(16) modules(0) msg-v3dot1(21) }

-- Repeated here for easy reference.

-- id-aa OBJECT IDENTIFIER ::= {iso(1) member-body(2) usa(840)

-- rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) attributes(2)}

-- IANA - Values needed

-- Branch for attestation statement types

id-ata OBJECT IDENTIFIER ::= { id-pkix (TBD1) }

-- A general comment is that a certificate is a signed binding between

-- public key and some identifying info. Below "cert" is an X.509

-- "Certificate". "opaqueCert" is just string of bytes that the

-- receiving CA must know how to parse given information not carried

-- in this object. "typedCert" and "typedFlatCert" both use an OID to

-- identify their types, but differ in that the encoding for typedCert

-- is always valid ASN1, whereas the typedFlatCert is just a string of

-- bytes that must be interpreted according to the type. Note that a

-- typedFlatCert MAY contain an encapsulated ASN1 object, but this is

-- not the best use of the type and is hereby discouraged.

CertificateChoice ::=

CHOICE {

cert Certificate, -- typical X.509 cert

opaqueCert [0] IMPLICIT OCTET STRING,

StJohns Expires 9 December 2023 [Page 14]

Internet-Draft CSR Attestation Attributes June 2023

typedCert [1] IMPLICIT TypedCert, -- not ASN1 parseable

typedFlatCert [2] IMPLICIT TypedFlatCert

}

-- Cribbed from definition of CONTENT-TYPE

-- Alternately as TypedCert ::= SingleAttribute

--

TYPED-CERT ::= TYPE-IDENTIFIER -- object id and a matching ASN1

-- structure encoded as a sequence

CertType ::= TYPED-CERT.&id

TypedCert ::= SEQUENCE {

certType TYPED-CERT.&id({TypedCertSet}),

content TYPED-CERT.&Type ({TypedCertSet}{@certType})

}

TypedCertSet TYPED-CERT ::=

... -- Empty for now,

}

-- The receiving entity is expected to be able to parse the certBody

-- field given the value of the certType field. This differs from

-- TypedCert in that the contents of the certBody field are not

-- necessarily well formed ASN1 in this case the certType tells you

-- how to parse the body of the OCTET STRING,

TypedFlatCert ::= SEQUENCE {

certType OBJECT IDENTIFIER,

certBody OCTET STRING

}

-- A sequence of certificates used to validate an attestation chain.

-- By convention, the first certificate in the chain is the one that

-- contains the public key used to verify the attestation. If the

-- related attestStatementAttribute contains more than a single

-- attestation, this attribute is expected to contain all of the

-- certificates needed to validate all attestations

id-aa-attestChainCerts OBJECT IDENTIFIER ::= { id-aa (TBDAA1) }

attestCertCertsAttribute ATTRIBUTE ::= {

TYPE SEQUENCE OF CertificateChoice

COUNTS MAX 1

IDENTIFIED BY id-aa-attestChainCerts

StJohns Expires 9 December 2023 [Page 15]

Internet-Draft CSR Attestation Attributes June 2023

}

-- If the signature is provided separately, the value field need not

-- contain the signature. Note that some attestation methods include

-- a signature method in the part signed by the signature and some do

-- not.

ATTEST-STATEMENT ::= CLASS {

&id OBJECT IDENTIFIER UNIQUE,

&Type, -- NOT optional

&algidPresent ParamOptions DEFAULT absent,

&sigPresent ParamOptions DEFAULT absent,

&sigType DEFAULT OCTET STRING

&ancillaryPresent ParamOptions DEFAULT absent,

&ancillaryType DEFAULT OCTET STRING

} WITH SYNTAX {

TYPE &Type

IDENTIFIED BY &id

[ALGID IS &algidPresent]

[SIGNATURE [TYPE &sigType] IS &sigPresent]

[ANCILLARY [TYPE &ancillaryType] IS &ancillaryPresent]

}

AttestStatement { ATTEST-STATEMENT:IOSet} ::= SEQUENCE

{

type ATTEST-STATEMENT.&id({IOSet}),

value ATTEST-STATEMENT.&Type({IOSet}{@type}),

algId [0] IMPLICIT AlgorithmIdentifier OPTIONAL,

signature [1] ATTEST-STATEMENT.&sigType OPTIONAL -- NOT implicit

ancillaryData [2] ATTEST-STATEMENT.&ancillaryType OPTIONAL

}

-- An attribute that contains a attestation statement.

id-aa-attestStatement OBJECT IDENTIFIER ::= { id-aa (TBDAA2) }

attestAttribute ATTRIBUTE ::= {

TYPE AttestStatement

IDENTIFIED BY id-aa-attestStatement

}

END

StJohns Expires 9 December 2023 [Page 16]

Internet-Draft CSR Attestation Attributes June 2023

Acknowledgements

Thanks to Russ Housley for a first and useful pass over the original

ASN.1. Thanks to Mike Ounsworth for not complaining too much when I

wrote this. Placeholder here for people who spend time reviewing the

draft!

Author's Address

Michael StJohns

NthPermutation Security LLC

Germantown, MD 20874

United States of America

Email: msj@nthpermutation.com

StJohns Expires 9 December 2023 [Page 17]