Workgroup: Network Working Group
Internet-Draft:
draft-ietf-lamps-cert-binding-for-multi-auth-00
Published: 24 February 2023
Intended Status: Standards Track
Expires: 28 August 2023
Authors: A. Becker R. Guthrie M. Jenkins
NSA NSA NSA

Related Certificates for Use in Multiple Authentications within a Protocol

Abstract

This document defines a new CSR attribute, relatedCertRequest, and a new X.509 certificate extension, RelatedCertificate. The use of the relatedCertRequest attribute in a CSR and the inclusion of the RelatedCertificate extension in the resulting certificate together provide additional assurance that two certificates each belong to the same end entity. This mechanism is particularly useful in the context of non-composite hybrid authentication, which enables users to employ the same certificates in hybrid authentication as in authentication done with only traditional or post-quantum algorithms.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The goal of this document is to define a method for providing assurance that multiple X.509 (aka PKIX) end-entity certificates are owned by the same entity, in order to perform multiple authentications where each certificate corresponds to a distinct digital signature. This method aims to facilitate post-quantum (PQ) migration while minimizing changes to the certificate format [RFC5280] and to current PKI best practices.

When using non-composite hybrid public key mechanisms, the party relying on a certificate (an authentication verifier or a key-establishment initiator) will want assurance that the private keys associated with each certificate are under the control of the same entity. This document defines a certificate extension, RelatedCertificate, that signals that the certificate containing the extension is able to be used in combination with the other specified certificate.

A certification authority (CA) organization that is asked to issue a certificate with such an extension may want assurance from a registration authority (RA) that the private keys (for example, corresponding to two public keys - one in an extant certificate, and one in a current request) belong to the same entity. To facilitate this, a CSR attribute is defined, called relatedCertRequest, that permits an RA to make such an assertion.

1.1. Overview

The general roadmap of this design is best illustrated via an entity (device, service, user token, etc.) that has an existing traditional certificate and requests a new PQ certificate, perhaps as part of an organization's migration to post-quantum cryptography. After the PQ certificate is issued, the use of the PQ and traditional certificates will depend on the protocols it supports and the organization's transition strategy.

* For protocols where authentication is not negotiated, and rather is limited by what the signer offers, such as in CMS and S/MIME, either the traditional signing key, the PQ signing key, or both signing keys may be invoked, according to which validators the signer anticipates.

* For protocols where authentication is negotiated in-protocol, such as TLS and IKEv2, either the traditional or PQ signing key may be invoked, according to the preference of the validator. If the protocol is enabled to do so, peers may request that both traditional and PQ authentication are used.

[It is possible for a strategy to comprise non-composite (such as described here) and composite schemes (as defined in [I-D.draft-ounsworth-pq-composite-sigs]). Because the mechanisms described in this document are not intended to effect composite certificate issuance, we do not further explore such a strategy.]

A validator that prefers multiple authentication types may be assisted by the inclusion of relevant information in the signer's certificate – that is, information that indicates the existence of a related certificate, and some assurance that those certificates have been issued to the same entity. This document describes a certificate request attribute and certificate extension that provide such assurance.

To support this concept, this document defines a new CSR attribute, relatedCertRequest, which contains information on how to locate a previously issued certificate and provides evidence that the requesting entity owns that certificate. When the RA makes the request to the CA, the CA uses the given information to locate the traditional certificate and then verifies ownership before generating the PQ certificate.

1.2. Use Case

This document defines the relatedCertRequest CSR attribute and the RelatedCertificate extension for specific use within the migration and transition to PQ cryptography. The intent is for a CA issuing a PQ certificate to add an X.509 extension that provides information about a traditional certificate in which the private key is under control of the same end entity as the PQ certificate, in order to facilitate a non-composite hybrid authentication mechanism.

The purpose of the CSR attribute detailed in this document is to serve as a tool for the RA to provide assurance to the CA organization that the entity that controls the private key of the PQ certificate being requested also controls the private key of a previously-issued traditional certificate. Similarly, the X.509 extension discussed in this document creates an association between the PQ certificate and the traditional certificate via end-entity ownership.

The attribute and subsequent extension together provide assurance from the CA organization that the same end-entity controls the private keys of both certificates. It is neither a requirement nor a mandate that either certificate be used with the other; it is simply an assurance that they can be used together, as they are under the control of the same entity.

2. 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 [RFC2119].

3. CSR and Related Certificates

3.1. The relatedCertRequest Attribute

This section defines a new CSR attribute designed to allow the RA to attest that the owner of the public key in the CSR also owns the public key associated with the end-entity certificate identified in this attribute. The relatedCertRequest attribute indicates the location of a previously issued certificate that the end-entity owns and wants identified in the new certificate requested through the CSR.

The relatedCertRequest attribute has the following syntax:

```
relatedCertRequest ATTRIBUTE ::= {
    WITH SYNTAX RequesterCertificate
    ID { TBD }
}

RequesterCertificate ::= SEQUENCE {
        certID        IssuerAndSerialNumber
        requestTime   BinaryTime
        locationInfo  AccessDescription
        signature     BIT STRING
}
```

The RequesterCertificate type has four fields:

* The certID field uses the IssuerAndSerialNumber type [RFC5652] to identify a previously issued end-entity certificate that the requesting entity also owns. IssuerAndSerialNumber is repeated here for convenience:

```
IssuerAndSerialNumber ::= SEQUENCE {
        issuer       Name,
        serialNumber CertificateSerialNumber }

CertificateSerialNumber ::= INTEGER
```

* The requestTime field uses the BinaryTime type [RFC6019] in order to ensure freshness, such that the signed data can only be used at the time of the initial CSR. The means by which the CA and RA synchronize time is outside the scope of this document. BinaryTime is repeated here for convenience:

```
BinaryTime ::= INTEGER (0..MAX)
```

* The locationInfo field uses AccessDescription [RFC5280] to provide information on the location of the other certificate the requesting entity owns. AccessDescription is repeated here for convenience:

```
AccessDescription  ::= SEQUENCE {
        accessMethod id-ad-relatedCerts,
        accessLocation GeneralName }

id-ad-relatedCerts OBJECT IDENTIFIER ::= { TBD }
```

The accessMethod value is id-ad-relatedCerts, which is used when the subject is an end-entity that owns certificates published in a repository.

* This document describes two acceptable values for accessLocation in the relatedCertRequest attribute; one value for when the same CA organization issues the PQ certificate and the referenced traditional certificate, and another value for when a different CA organization previously issued the traditional certificate.

  - If the CA organization is the same for both certificates, the accessLocation value SHOULD be available via HTTP or FTP, and therefore must be a URI that points to a file containing a certificate or certificate chain that the requesting entity owns, as detailed in [RFC5280]. The file must permit access to a PKCS#7 ‘certs-only’ repository containing either a single DER encoded X.509 certificate or an entire certificate chain.

  - If the CA organization issuing the PQ certificate is not the same as the CA organization that issued the traditional certificate referenced in the CSR, then the accessLocation value URI MAY be a dataURI [RFC2397] containing inline degenerate PKCS#7 consisting of all the certificates and CRLs required to validate the traditional certificate. This allows validation without the CA organization having to retrieve certificates/CRLs from another CA. Further discussion of requirements for this scenario is in Section 5.

* The signature field provides evidence that the requesting entity owns the certificate indicated by the certID. Specifically, the signature field contains a digital signature over the concatenation of DER encoded requestTime and IssuerAndSerialNumber (without tag and length). The concatenated value is signed using the signature algorithm and private key associated with the certificate identified by the certID field.

  - If the related certificate is a KE certificate, use the ECC KE private key to sign one time for POP (as detailed in NIST SP 800-57 Part 1 Rev 5 Section 8.1.5.1.1.2)

The validation of this signature by the CA ensures that the owner of the CSR also owns the certificate indicated in the relatedCertRequest attribute.

3.2. CSR Processing

The information provided in the relatedCertRequest attribute allows the CA to locate a previously issued certificate that the requesting entity owns, and verify ownership by using the public key in that certificate to validate the signature in the relatedCertRequest attribute.

If a CA receives a CSR that includes the relatedCertRequest attribute is equipped to recognize and understand the attribute the CA:

* MUST retrieve and validate the certificate identified in the relatedCertRequest using the information provided in AccessDescription. The CA then extracts the IssuerAndSerialNumber from the indicated certificate and compares this value against the IssuerAndSerialNumber provided in the certID field of relatedCertRequest.

* MUST check that the BinaryTime indicated in the requestTime field is sufficiently fresh.

* MUST verify the signature field of the relatedCertRequest attribute. The CA validates the signature using the public key associated with the certificate it located via the info provided in the AccessDescription field. The details of the validation process are outside the scope of this document.

* SHOULD issue the new certificate containing the RelatedCertificate extension as specified in Section 4, which references the certificate indicated in the attribute's IssuerAndSerialNumber field.

The RA MUST only allow a previously issued certificate to be indicated in the relatedCertRequest attribute in order to enable the CA to perform the required signature verification.

The RA MAY send the CA a CSR containing a relatedCertRequest attribute that includes the IssuerAndSerialNumber of a certificate that was issued by a different CA.

4. Related Certificate

4.1. The RelatedCertificate Extension

This section profiles a new X.509v3 certificate extension, RelatedCertificate. RelatedCertificate creates an association between the certificate containing the RelatedCertificate extension and the certificate referenced within the extension. When two end-entity certificates are used in a protocol, where one of the certificates contains a RelatedCertificate extension that references another certificate, the authenticating entity is provided with additional assurance that all certificates belong to the same entity.

The RelatedCertificate extension is an octet string that contains the hash of a single end-entity certificate.

The RelatedCertificate extension has the following syntax:

```
--  Object Identifiers for certificate extension
  id-relatedCertificate OBJECT IDENTIFIER ::= { TBD }

--  X.509 Certificate extension
  RelatedCertificate ::= OCTET STRING
               -- hash of entire related certificate }
```

The extension is comprised of an octet string, which is the digest value obtained from hashing the entire related certificate identified in the CSR attribute defined above, relatedCertRequest. The algorithm used to hash the certificate in the RelatedCertificate extension MUST match the hash algorithm used to sign the certificate that contains the extension.

ED NOTE: We recognize the following SCVP structure from [RFC5055] may be preferable to defining a new extension, however, it adds extra bytes of options for the hash function that may be deemed unnecessary for the RelatedCertificates extension. The structure is repeated here for convenience:

```
SCVPCertID ::= SEQUENCE {
    certHash       OCTET STRING,
    IssuerSerial   SCVPIssuerSerial,
    hashAlgorithm  AlgorithmIdentifier DEFAULT {algorithm sha-1}}
```

This extension SHOULD NOT be marked critical. Marking this extension critical would severely impact interoperability.

For certificate chains, this extension MUST only be included in the end-entity certificate.

For the RelatedCertificate extension to be meaningful, a CA that issues a certificate with this extension:

* MUST only include a certificate in the extension that is listed and validated in the relatedCertRequest attribute of the CSR submitted by the requesting entity.

* MUST ensure that all certificates are intended for the same use case. For example, the CA must ensure that both certificates have the same key usage [RFC5280]. The intended purpose of the certificate may be determined by policy or other means (e.g KU, EKU OIDS) but this is outside the scope of this document.

* SHOULD determine that all certificates are valid at the time of issuance. The usable overlap of validity periods is a Subscriber concern.

4.2. Endpoint Protocol Multiple Authentication Processing

When the preference to use a non-composite hybrid authentication mode is expressed by an endpoint through the protocol itself (e.g., during negotiation), the use of the RelatedCertificate extension and its enforcement are left to the protocol's native authorization mechanism (along with other decisions endpoints make about whether to complete or drop a connection).

If an endpoint has indicated that it is willing to do non-composite hybrid authentication and receives the appropriate authentication data, it SHOULD check end-entity certificates for the RelatedCertificate extension. If present in one certificate, it SHOULD:

* Compute the appropriate hash of the other end-entity certificate received. The hash algorithm is the same as the one used to sign the certificate containing the extension.

* Verify that the hash value matches the hash entry in the RelatedCertificate extension.

It is outside the scope of this document how to proceed with authentication based on the outcome of this verification process. Different determinations may be made depending on each peer’s policy regarding whether both or at least one authentication needs to succeed.

5. CA Organization Considerations

The relatedCertRequest CSR attribute provides assertion of end entity control of the private key of a related certificate to the CA organization. There may arise scenarios where a public-facing CA organization is not configured to validate signatures associated with certificates that have been issued by a different CA organization. In this case, recognition of the contents in the relatedCertRequest attribute may be contingent upon a pre-arranged contract with pre-configured trust anchors from the other CA organization, and include agreements on certificate policy with regards to certificate application, issuance, and acceptance. Further, matching policies between CA organizations on protection of private key may be necessary in order for the whole assurance level from the other CA organization to be accepted.

In a similar vein, if the CA organization issuing the PQ certificate can recognize the relatedCertRequest attribute in the CSR and wishes to issue the certificate with the RelatedCerts extension, it may be the case that a different CA organization issued the related certificate referenced in the CSR. In order to ensure that the certificates have been issued under homogeneous sets of security parameters, the certificate policies should be the same with regard to common security requirements. The CA organizations should have the same certificate policy, with the same identifier, or there should exist a certificate policy mapping between the two, to ensure that the policies for protection of private key are equivalent. The relatedCertRequest attribute and subsequent RelatedCertificate certificate extension are solely intended to provide assurance that both private keys are controlled by the same end entity.

6. Security Considerations

This document inherits security considerations identified in [RFC5280].

The mechanisms described in this document provide only a means to express that multiple certificates are related. They are intended for the interpretation of the recipient of the data in which they are embedded (i.e. a CSR or certificate). They do not by themselves effect any security function.

Authentication, unlike key establishment, is necessarily a one-way arrangement. That is, authentication is an assertion made by a claimant to a verifier. The means and strength of mechanism for authentication have to be to the satisfaction of the verifier. A system security designer needs to be aware of what authentication assurances are needed in various parts of the system and how to achieve that assurance. In a closed system (e.g. Company X distributing firmware to its own devices) the approach may be implicit. In an online protocol like IPsec where the peers are generally known, any mechanism selected from a pre-established set may be sufficient. For more promiscuous online protocols, like TLS, the ability for the verifier to express what is possible and what is preferred – and to assess that it got what it needed – is important.

A certificate is an assertion of binding between an identity and a public key. However, that assertion is based on several other assurances – specifically, that the identity belongs to a particular physical entity, and that that physical entity has control over the private key corresponding to the public. For any hybrid approach, it is important that there be evidence that the same entity controls all private keys at time of use (to the verifier) and at time of registration (to the CA).

All hybrid implementations are vulnerable to a downgrade attack in which a malicious peer does not express support for PQ algorithms, resulting in an exchange that can only rely upon traditional algorithms for security.

Implementors should be aware of risks that arise from the retrieval of a related certificate via the AccessDescription method provided in the relatedCertRequest CSR attribute, if the URI points to malicious code. Implementors should ensure the data is properly formed and validate the retrieved data fully.

7. IANA Considerations

This document defines an extension for use with X.509 certificates. IANA is requested to register an OID in the PKIX certificate extensions arc [RFC7299]:

```
id-relatedCert OBJECT IDENTIFIER ::= { id-pkix 1 tbd }
```

with this document as the Required Specification.

This document also defines a CSR attribute. IANA is requested to register an OID:

```
id-relatedCertRequest OBJECT IDENTIFIER ::= { tbd }
```

An additional OID for a specific accessMethod is requested:

```
id-ad-relatedCert OBJECT IDENTIFIER ::= { tbd }
```

8. References

8.1. Normative References

[I-D.draft-ounsworth-pq-composite-sigs] Ounsworth, M. and M. Pala, "Composite Signatures For Use In Internet PKI", February 2022, <https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-sigs/06/>.

[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>.

[RFC2397] Masinter, L., "The "data" URL scheme", RFC 2397, DOI 10.17487/RFC2397, August 1998, <https://www.rfc-editor.org/info/rfc2397>.

[RFC5055] Freeman, T., Housley, R., Malpani, A., Cooper, D., and W. Polk, "Server-Based Certificate Validation Protocol (SCVP)", RFC 5055, DOI 10.17487/RFC5055, December 2007, <https://www.rfc-editor.org/info/rfc5055>.

[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/info/rfc5280>.

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

[RFC6019] Housley, R., "BinaryTime: An Alternate Format for Representing Date and Time in ASN.1", RFC 6019, DOI 10.17487/RFC6019, September 2010, <https://www.rfc-editor.org/info/rfc6019>.

8.2. Informative References

[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>.

[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/info/rfc6268>.

[RFC7299] Housley, R., "Object Identifier Registry for the PKIX Working Group", RFC 7299, DOI 10.17487/RFC7299, July 2014, <https://www.rfc-editor.org/info/rfc7299>.

Appendix A. ASN.1 Module

The following RelatedCertificate ASN.1 module describes the RequesterCertificate type found in the relatedCertAttribute. It pulls definitions from modules defined in [RFC5912], and [RFC6268], and [RFC6019] for the AccessDescription type, IssuerAndSerialNumber type, and BinaryTime type, respectively.

```
RelatedCertificate {optional id value} DEFINITIONS ::=
  BEGIN
    {
     IMPORTS

      -- Imports from New PKIX ASN.1 [RFC5912]

        AccessDescription
         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) }

      -- Imports from Additional New ASN.1 Modules [RFC6268]

        IssuerAndSerialNumber
         CryptographicMessageSyntax-2010
           { iso(1) member-body(2) us(840) rsadsi(113549)
             pkcs(1) pkcs-9(9) smime(16) modules(0)
             id-mod-cms-2009(58) }

      -- Imports from BinaryTime [RFC6019]

        BinaryTime
         BinarySigningTimeModule
           { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
             pkcs-9(9) smime(16) modules(0) 27 }

      ;

      -- relatedCertRequest Attribute

      relatedCertRequest ATTRIBUTE ::=
        {
         WITH SYNTAX RequesterCertificate
         ID { TBD }
        }

      -- RequesterCertificate definition

      RequesterCertificate ::= SEQUENCE
        {
         certID        IssuerAndSerialNumber
         requestTime   BinaryTime
         locationInfo  AccessDescription
         signature     BIT STRING
        }
    }
  END
```

Authors' Addresses

Alison Becker
National Security Agency

Email: aebecke@uwe.nsa.gov

Rebecca Guthrie
National Security Agency

Email: rmguthr@uwe.nsa.gov

Michael Jenkins
National Security Agency

Email: mjjenki@cyber.nsa.gov