

---

Stream: Internet Engineering Task Force (IETF)  
RFC: [9763](#)  
Category: Standards Track  
Published: June 2025  
ISSN: 2070-1721  
Authors: A. Becker R. Guthrie M. Jenkins  
NSA NSA NSA

# RFC 9763

## Related Certificates for Use in Multiple Authentications within a Protocol

---

### Abstract

This document defines a new Certificate Signing Request (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

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9763>.

### Copyright Notice

Copyright (c) 2025 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.

## Table of Contents

1. Introduction	2
1.1. Overview	3
2. Requirements Language	4
3. CSR and Related Certificates	4
3.1. The relatedCertRequest Attribute	4
3.2. CSR Processing	6
4. Related Certificate	6
4.1. The RelatedCertificate Extension	6
4.2. Endpoint Protocol Multiple Authentication Processing	7
5. Use Case	8
6. CA Organization Considerations	8
7. Security Considerations	9
8. IANA Considerations	10
9. References	10
9.1. Normative References	10
9.2. Informative References	11
Appendix A. ASN.1 Module	11
Authors' Addresses	13

## 1. Introduction

The goal of this document is to define a method for providing assurance that two 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 the use of two certificates for authentication in a secure protocol 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 (defined here as the entity or organization that runs a CA and determines the policies and tools the CA will use) that is asked to issue a certificate with such an extension may want assurance from a registration authority (RA) that the private keys (corresponding to, for example, 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, called `relatedCertRequest`, is defined to permit an RA to make such an assertion.

## 1.1. Overview

The general roadmap of this design is best illustrated via an entity (e.g., a device, service, user token) that has an existing certificate (Cert A) and requests a new certificate (Cert B), perhaps as part of an organization's transition strategy to migrate their PKI from traditional cryptography to post-quantum cryptography (PQC).

- For protocols where authentication is not negotiated but instead is limited by what the signer offers, such as in Cryptographic Message Syntax (CMS) and S/MIME, either Cert A's signing key, Cert B's signing key, or both signing keys may be invoked, according to which verifiers the signer anticipates.
- For protocols where authentication is negotiated in-protocol, such as TLS and Internet Key Exchange Protocol Version 2 (IKEv2), either Cert A or Cert B's signing key may be invoked, according to the preference of the verifier. If the protocol is enabled to do so, peers may request that both Cert A and Cert B are used for authentication.

A verifier that prefers multiple authentication types may be assisted by the inclusion of relevant information in one of the signer's certificates; 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 (Cert A) 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 Cert A and then verifies ownership before generating the new certificate, Cert B.

## 2. Requirements Language

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 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:

```
id-aa-relatedCertRequest OBJECT IDENTIFIER ::= { id-aa 60 }

aa-relatedCertRequest ATTRIBUTE ::= {
  TYPE RequesterCertificate
  IDENTIFIED BY id-aa-relatedCertRequest}

RequesterCertificate ::= SEQUENCE {
  certID      IssuerAndSerialNumber,
  requestTime BinaryTime,
  locationInfo UniformResourceIdentifier,
  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 UniformResourceIdentifier to provide information on the location of the other certificate the requesting entity owns. UniformResourceIdentifier is defined as:

```
UniformResourceIdentifier ::= IA5String
```

The UniformResourceIdentifier is a pointer to a location via HTTP(S) or a data URL. This field can contain one of two acceptable values:

- If the request for (new) Cert B is to the CA organization that also issued (existing) Cert A, then the UniformResourceIdentifier value **SHOULD** be a URL that points to a file containing a certificate or certificate chain that the requesting entity owns, as detailed in [RFC5280]; the URL is made available via HTTP or HTTPS. The file must permit access to a CMS 'certs-only' message containing the end-entity certificate or the entire certificate chain. This option uses less data than a data URL. All certificates contained must be DER encoded.
  - If the request for (new) Cert B is to a CA organization different than the CA organization that issued the certificate (existing) Cert A referenced in the CSR, then the UniformResourceIdentifier value **SHOULD** be a data URL [RFC2397] containing inline degenerate PKCS#7 (see Sections 3.2.1 and 3.8 of [RFC8551]) consisting of all the certificates and CRLs required to validate Cert A. This allows the CA to perform validation (as described in Section 3.2) without 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 field. Specifically, the signature field contains a digital signature over the concatenation of DER-encoded IssuerAndSerialNumber and BinaryTime. 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 key establishment certificate (e.g., using RSA key transport or Elliptic Curve Cryptography (ECC) key agreement), the private key is used to sign one time for proof of possession (POP) (as detailed in Section 8.1.5.1.1.2 of [NIST-SP-800-57]).

The verification 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 confirm ownership by using the public key in that certificate to verify the signature in the `relatedCertRequest` attribute.

If a CA receives a CSR that includes the `relatedCertRequest` attribute and that CA supports the attribute, the CA:

- **MUST** retrieve the certificate identified in the `relatedCertRequest` attribute using the information provided in `UniformResourceIdentifier`, and validate it using certificate path validation rules defined in [\[RFC5280\]](#). 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. Note that sufficient freshness is defined by local policy and is out of the scope of this document.
- **MUST** verify the signature field of the `relatedCertRequest` attribute. The CA verifies the signature using the public key associated with the certificate identified by the `relatedCertRequest` attribute. The details of the verification 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 CA may also apply local policy regarding the suitability of the related certificate, such as validity period remaining.

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 specifies a new X.509 certificate extension, `RelatedCertificate`. `RelatedCertificate` creates an association between the certificate containing the `RelatedCertificate` extension (Cert B) and the certificate referenced within the extension (Cert A). When two end-entity certificates are used in a protocol, where one of the certificates contains a `RelatedCertificate` extension that references the other certificate, the authenticating entity is provided with additional assurance that both certificates belong to the same entity.

The `RelatedCertificate` extension contains the hash of a single end-entity certificate.

The `RelatedCertificate` extension has the following syntax:

```
-- Object Identifier for certificate extension
id-relatedCert OBJECT IDENTIFIER ::= { 36 }

-- X.509 Certificate extension
RelatedCertificate ::= SEQUENCE {
    hashAlgorithm DigestAlgorithmIdentifier,
    hashValue      OCTET STRING }
```

The extension is a SEQUENCE of two fields. The hashAlgorithm field identifies the hash algorithm used to compute hashValue, which is the digest value obtained from hashing the entire related certificate identified in the relatedCertRequest CSR attribute defined above. If there is a hash algorithm explicitly indicated by the related certificate's signature OID (e.g., ecdsa-with-SHA512), that hash algorithm **SHOULD** also be used for this extension.

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 in the relatedCertRequest attribute of the CSR submitted by the requesting entity.
- **MUST** ensure that the related certificate contains the key usage (KU) bits and extended key usage (EKU) OIDs [RFC5280] being asserted in the certificate being issued.
- **SHOULD** determine that the related certificate is valid at the time of issuance. The usable overlap with the validity period of the newly issued certificate 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 existing authorization mechanism (along with other decisions endpoints make about whether to complete or drop a connection).

If an endpoint has indicated that it supports non-composite hybrid authentication and receives the appropriate authentication data, it should check end-entity certificates (Cert A and Cert B) for the RelatedCertificate extension. If present in one certificate (e.g., Cert B), it should:

- Compute the appropriate hash of Cert A, the other end-entity certificate received.
- Confirm that the hash value matches the hash entry in the RelatedCertificate extension of Cert B.

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

## 5. Use Case

The general design of this method is best illustrated through specific use within a migration strategy to PQC via a non-composite hybrid authentication mechanism. The intent is for a CA issuing a new, post-quantum (PQ) certificate to add an X.509 extension that provides information about a previously issued, traditional certificate in which the private key is controlled by the same end entity as the PQ certificate.

In the following scenario, an entity currently has a traditional certificate and requests that a new, PQ certificate be issued containing a RelatedCertificate extension, which references the entity's traditional certificate.

The RA receives a CSR for a PQ certificate, where the CSR includes the relatedCertRequest attribute detailed in this document. The relatedCertRequest attribute includes a certID field that identifies the entity's previously issued traditional certificate and a signature field in which the requesting entity produces a digital signature over the concatenation of the IssuerAndSerialNumber and BinaryTime, using the private key of the certificate identified by the IssuerAndSerialNumber.

The purpose of the relatedCertRequest attribute 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 the referenced, previously issued traditional certificate.

Upon receipt and validation of the CSR, the CA issues a PQ certificate to the requesting entity that includes the RelatedCertificate extension detailed in this document; the extension includes a hash of the entire traditional certificate identified in the CSR. The X.509 extension creates an association between the PQ certificate and the traditional certificate via an assertion of 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.

## 6. CA Organization Considerations

The relatedCertRequest CSR attribute asserts end-entity control of the private key associated with a related certificate (Cert A) to the CA organization issuing a new certificate (Cert B). A public-facing CA organization may not be configured to validate certificates that have been issued by different CA organizations. In this case, recognition of the contents in the relatedCertRequest attribute may necessitate pre-arrangement, e.g., a contract with pre-configured trust anchors from another CA organization and agreements regarding policies concerning certificate application, issuance, and acceptance.



Continuing with this scenario, in order for the CA organization to ensure that Cert B is issued under security parameters comparable to Cert A, the issuing CA organization should match the issued certification policies to the related ones. The issuing CA organization, as part of its validation of Cert A, ascertains what policies are asserted in Cert A's certification path and determines which of their own certification policies will best ensure that the protection of the private key associated with Cert B is comparable to that of Cert A. 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.

## 7. 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 the authentication mechanism have to be satisfactory to 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 its requirements were met -- 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, especially that the identity belongs to a particular physical entity and that the physical entity has control over the private key corresponding to the public key. 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 the stronger algorithm, resulting in an exchange that can only rely upon a weaker algorithm for security.

Implementors should be aware of risks that arise from the retrieval of a related certificate via the `UniformResourceIdentifier` provided in the `relatedCertRequest` CSR attribute, as a URL can point to malicious code. Implementors should ensure the data is properly formed and validate the retrieved data fully.

CAs should be aware that retrieval of existing certificates may be subject to observation; if this is a concern, it is advisable to use the data URL option described in [Section 3.1](#).

## 8. IANA Considerations

This document defines an extension for use with X.509 certificates. IANA has registered the following OID in the "SMI Security for PKIX Certificate Extension" registry (1.3.6.1.5.5.7.1):

Decimal	Description	References
36	id-pe-relatedCert	RFC 9763

Table 1

The registration procedure is Specification Required [RFC8126].

This document defines a CSR attribute. IANA has registered the following OID in the "SMI Security for S/MIME Attributes (1.2.840.113549.1.9.16.2)" registry:

Decimal	Description	References
60	id-aa-relatedCertRequest	RFC 9763

Table 2

This document defines an ASN.1 module in [Appendix A](#). IANA has registered the following OID for the module identifier in the "SMI Security for PKIX Module Identifier" registry (1.3.6.1.5.5.7.0):

Decimal	Description	References
115	id-mod-related-cert-2023	RFC 9763

Table 3

## 9. References

### 9.1. 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>>.
- [RFC2397] Masinter, L., "The "data" URL scheme", RFC 2397, DOI 10.17487/RFC2397, August 1998, <<https://www.rfc-editor.org/info/rfc2397>>.
- [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>>.
- [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/info/rfc8174>>.

## 9.2. Informative References

- [NIST-SP-800-57] Barker, E., "Recommendation for Key Management: Part 1 - General", National Institute of Standards and Technology, NIST SP 800-57pt1r5, DOI 10.6028/NIST.SP.800-57pt1r5, May 2020, <<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.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>>.
- [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>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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>>.

## 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] for the IssuerAndSerialNumber type and in [RFC6019] for the BinaryTime type.

```
RelatedCertificate { iso(1) identified-organization(3) dod(6)
  internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
  id-mod-related-cert-2023(115) }
```

```
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
```

```
IMPORTS
```

```

ATTRIBUTE, EXTENSION
  FROM PKIX-CommonTypes-2009 -- in RFC 5912
  { iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-pkixCommon-02(57) }

IssuerAndSerialNumber, DigestAlgorithmIdentifier
  FROM CryptographicMessageSyntax-2010 -- in RFC 6268
  { iso(1) member-body(2) us(840) rsadsi(113549)
    pkcs(1) pkcs-9(9) smime(16) modules(0)
    id-mod-cms-2009(58) }

BinaryTime
  FROM BinarySigningTimeModule -- in RFC 6019
  { iso(1) member-body(2) us(840) rsadsi(113549)
    pkcs(1) pkcs-9(9) smime(16) modules(0)
    id-mod-binarySigningTime(27) } ;

-- Object identifier arcs

id-pe OBJECT IDENTIFIER ::= { iso(1) identified-organization(3)
  dod(6) internet(1) security(5) mechanisms(5) pkix(7) 1 }

id-aa OBJECT IDENTIFIER ::= { iso(1) member-body(2) usa(840)
  rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) 2 }

-- relatedCertificate Extension

id-pe-relatedCert OBJECT IDENTIFIER ::= { id-pe 36 }

RelatedCertificate ::= SEQUENCE {
  hashAlgorithm DigestAlgorithmIdentifier,
  hashValue     OCTET STRING }

ext-relatedCertificate EXTENSION ::= {
  SYNTAX RelatedCertificate
  IDENTIFIED BY id-pe-relatedCert }

-- relatedCertRequest Attribute

id-aa-relatedCertRequest OBJECT IDENTIFIER ::= { id-aa 60 }

RequesterCertificate ::= SEQUENCE {
  certID      IssuerAndSerialNumber,
  requestTime BinaryTime,
  locationInfo UniformResourceIdentifiers,
  signature    BIT STRING }

UniformResourceIdentifiers ::= SEQUENCE SIZE (1..MAX) OF URI

URI ::= IA5String

aa-relatedCertRequest ATTRIBUTE ::= {
  TYPE RequesterCertificate

```

```
IDENTIFIED BY id-aa-relatedCertRequest }
```

```
END
```

## Authors' Addresses

### **Alison Becker**

National Security Agency

Email: [aebecke@uwe.nsa.gov](mailto:aebecke@uwe.nsa.gov)

### **Rebecca Guthrie**

National Security Agency

Email: [rmguthr@uwe.nsa.gov](mailto:rmguthr@uwe.nsa.gov)

### **Michael Jenkins**

National Security Agency

Email: [mjjenki@cyber.nsa.gov](mailto:mjjenki@cyber.nsa.gov)