Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

T. Okubo
Penguin Securities Pte. Ltd.
C. Bonnell
DigiCert, Inc.
J. Gray
M. Ounsworth
Entrust
J. Mandel
AKAYLA, Inc.
7 July 2025

A Mechanism for X.509 Certificate Discovery
draft-ietf-lamps-certdiscovery-01

Abstract

This document specifies a method to discover a secondary X.509 certificate associated with an X.509 certificate to enable efficient multi-certificate handling in protocols. The objective is threefold: to enhance cryptographic agility, improve operational availability, and accommodate multi-key/certificate usage. The proposed method aims to maximize compatibility with existing systems and is designed to be legacy-friendly, making it suitable for environments with a mix of legacy and new implementations. It includes mechanisms to provide information about the target certificate's signature algorithm, public key algorithm and the location of the secondary X.509 certificate, empowering relying parties to make informed decisions on whether to fetch the Secondary Certificate.

The primary motivation for this method is to address the limitations of traditional certificate management approaches, which often lack flexibility, scalability, and seamless update capabilities. By leveraging this mechanism, subscribers can achieve cryptographic agility by facilitating the transition between different algorithms or X.509 certificate types. Operational redundancy is enhanced by enabling the use of backup certificates and minimizing the impact of Primary Certificate expiration or CA infrastructure failures.

The approach ensures backward compatibility with existing systems and leverages established mechanisms, such as the subjectInfoAccess extension, to enable seamless integration.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://lamps-wg.github.io/certificatediscovery/draft-ietf-lamps-certdiscovery.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-lamps-certdiscovery/.

Source for this draft and an issue tracker can be found at https://github.com/lamps-wg/certificatediscovery.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The efficient discovery of X.509 certificates play a critical role in modern cryptographic systems. Traditional certificate management approaches often face challenges in terms of flexibility, scalability, and seamless updates. To address these limitations, this document proposes a novel approach to certificate discovery utilizing the Subject Information Access extension within X.509 certificates.

The primary objective of this approach is to enable efficient multi-certificate handling in protocols, offering several key benefits. First, it enhances cryptographic agility by facilitating smooth transitions between different algorithms or X.509 certificate types. This is particularly valuable in scenarios where subscribers need to upgrade their cryptographic algorithms or adopt new certificate types while maintaining backward compatibility with existing systems.

Second, the proposed method improves operational availability by introducing redundancy in certificate usage. It enables the use of secondary certificates that can serve as backups, ensuring seamless continuity of services even in the event of Primary Certificate expiration or disruptions in the CA infrastructure.

Finally, the approach accommodates multi-key/certificate usage, allowing for a relying party to obtain certificates to perform cryptographic operations that are not certified by a single certificate.

The proposed method is designed to maximize compatibility with existing systems, including legacy implementations. It leverages the subjectInfoAccess extension, which is already established in X.509 certificates, and does not require modifications to the referring certificates. This ensures ease of adoption and avoids disruptions to current certificate management practices.

In the following sections, we will outline the details of the proposed approach, including the structure of the SIA extension, the modes of operation, and the considerations for secure implementation and deployment.

By leveraging the capabilities of the SIA extension for certificate discovery, organizations can enhance cryptographic agility, improve operational availability, and accommodate complex multi-key/ certificate scenarios, leading to more secure and resilient cryptographic systems.

1.1. Use Case 1: Algorithm Agility

The first use case is improving algorithm agility. For example, the Primary Certificate uses a widely adopted cryptographic algorithm while the Secondary Certificate uses the algorithm that is new and not widely adopted yet. The relying party will be presented with the opportunity to try the new algorithms and certificate types. This will be particularly useful when transitioning from one algorithm to another or to a new certificate/credential type.

In addition, the server may look at the logs to determine how ready the client side is to shift to completely rollover to the new algorithm. This allows the subscriber to gather the metrics necessary to make an informed decision on the best timing to do an algorithm rollover without relying on third parties or security researchers. This is particularly useful for PKIs that have a wide array of client software and requires careful consideration.

1.2. Use Case 2: Operational Redundancy

The second use case is where the Primary and Secondary Certificate adopts the same cryptographic algorithms but for instance, uses certificates issued by two different CAs or two certificates that have different validity periods. The Secondary Certificate may be used as a backup certificate in case the Primary Certificate validity is about to expire.

A common issue is when the intermediate CA certificate expires, and the subscriber forgets to update the intermediate CA configured on the server. Similar to when some software collects the parent

certificate through authorityInfoAccess CA Issuer access method when the intermediate certificate is absent, the peer certificate can be obtained.

Due to increased adoption of the ACME protocol, the burden of maintaining the availability of a service is shifted to the CA issuance infrastructure and the availability would be dependent on the CA infrastructure. To increase the operational redundancy, this mechanism can be used to point to another set of certificates that are independent from the Primary Certificate to minimize the chance of a failed transaction.

1.3. Use Case 3: Dual Use

The third use case is where one certificate is used by the named subject for a particular cryptographic operation and a relying party wishes to obtain the public key of the named subject for a different cryptographic operation. For example, the recipient of an email message which was signed using a key that is certified by a single use signing S/MIME certificate may wish to send an encrypted email to the sender. In this case, the recipient will need the sender’s public key used for encryption. A pointer to the named subject’s encryption certificate will permit the recipient to send an encrypted reply.

2. Conventions and Definitions

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.

2.1. Definitions

For conciseness, this section defines several terms that are frequently used throughout this specification.

Primary Certificate: The X.509 certificate that has the subjectInfoAccess extension with the certDiscovery accessMethod pointing to a Secondary Certificate.

Secondary Certificate: The X.509 certificate that is referenced by the Primary Certificate in the subjectInfoAccess extension certDiscovery accessMethod. This certificate may also have a reference to the Primary Certificate in the subjectInfoAccess extension.

3. Certificate Discovery Access Method

This document specifies the new certDiscovery access method for X.509 Subject Information Access (SIA) extension defined in [RFC5280].

The syntax of subject information access extension syntax is repeated here for convenience:

```
SubjectInfoAccessSyntax  ::=
        SEQUENCE SIZE (1..MAX) OF AccessDescription

AccessDescription  ::=  SEQUENCE {
        accessMethod          OBJECT IDENTIFIER,
        accessLocation        GeneralName  }
```

This document defines a new access method id-ad-certDiscovery which is an OBJECT IDENTIFIER that indicates the accessMethod is for certificate discovery.

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

The 'accessLocation' is a GeneralName otherName type as defined in [RFC5280]. Recall that the otherName type is defined as AnotherName:

```
AnotherName ::= SEQUENCE {
     type-id    OBJECT IDENTIFIER,
     value      [0] EXPLICIT ANY DEFINED BY type-id }
```

Which this document defines as:

```
-- Other Name OID Arc --
id-on OBJECT IDENTIFIER ::= { id-pkix 8 }

-- Certificate Discovery Access Descriptor --
id-on-relatedCertificateDescriptor OBJECT IDENTIFIER ::= { id-on TBD }

on-RelatedCertificateDescriptor OTHER-NAME ::= {
      RelatedCertificateDescriptor IDENTIFIED BY id-on-relatedCertificateDescriptor
   }
```

Where id-on-relatedCertificateDescriptor is the OBJECT IDENTIFIER (type-id) and the value is RelatedCertificateDescriptor

RelatedCertificateDescriptor is defined as follows:

```
 RelatedCertificateDescriptor ::= SEQUENCE {
      certref CertReference,
      purpose DiscoveryPurposeId,
      signatureAlgorithm [0] IMPLICIT AlgorithmIdentifier OPTIONAL,
      publicKeyAlgorithm [1] IMPLICIT AlgorithmIdentifier OPTIONAL }
```

RelatedCertificateDescriptor is composed of 4 components which are defined below.

3.1. CertLocation

CertLocation is defined by the following:

```
CertLocation ::= IA5String
```

CertLocation is to specify the Uniform Resource Identifier ([RFC3986]) where the secondary certificate is located.

3.2. CertReference

CertReference is defined by the following:

```
 CertReference ::= CHOICE {
      direct Certificate,
      indirect [0] IMPLICIT CertIndirectReference
   }
```

Which is a CHOICE defining either a direct reference to a Certificate (meaning that the full Secondary Certificate is embedded within the Primary Certificate), or an indirect reference (meaning that information to fetch the Secondary Certificate is provided). The syntax of an indirect reference is described below.

3.3. CertIndirectReference

CertIndirectReference is defined by the following:

```
CertIndirectReference ::= SEQUENCE {
         location CertLocation,
         certHash [0] IMPLICIT CertHash OPTIONAL
      }
```

The certificate is referenced by an IA5String that has the URL reference to the Secondary Certificate. The indirect reference also includes an optional certHash value which can be used to include a cryptographic hash of the DER Encoded Secondary Certificate. The syntax of a certHash is described below.

3.4. CertHash

CertHash is defined by the following:

```
CertHash ::= SEQUENCE {
   value OCTET STRING,
   -- TODO Add IssuerAndSerialNumber?
   hashAlgorithm AlgorithmIdentifier DEFAULT {algorithm sha-256}
}
```

certHash is defined as a SEQUENCE containing the OCTET STRING value which is the hash of the DER Encoded reference certificate as well as the hashAlgorithm which contains the AlgorithmIdentifier for the chosen Hash value. All implementations MUST support SHA-256 via id-sha256, and other hash functions MAY be supported.

3.5. Purpose

The purpose describes the purpose of the discovery method. Currently the following purpose ids are defined:

```
 -- Purpose OBJECT IDENTIFIER
id-rcd-alg-agility OBJECT IDENTIFIER ::=
                                {id-on-relatedCertificateDescriptor 1}

id-rcd-redundancy OBJECT IDENTIFIER ::=
                                {id-on-relatedCertificateDescriptor 2}

id-rcd-dual OBJECT IDENTIFIER ::=
                                {id-on-relatedCertificateDescriptor 3}

id-rcd-priv-key-stmt OBJECT IDENTIFIER ::=
                                {id-on-relatedCertificateDescriptor 4}

id-rcd-self OBJECT IDENTIFIER ::=
                                {id-on-relatedCertificateDescriptor 5}
```

3.5.1. Algorithm Agility

This purpose indicates the referenced certificate’s purpose is to provide algorithm agility; i.e. the two certificates will use different cryptographic algorithms for the same key operations. The two certificates SHOULD be equivalent except for cryptographic algorithm; i.e. the key usages SHOULD match.

3.5.2. Redundancy

This purpose indicates the referenced certificate’s purpose is to provide operational redundancy; i.e. the Secondary Certificate could be issued by a different CA or has a different validity period which can be used as a backup if the Primary set of certificates is about to expire.

3.5.3. Dual Usage

This purpose indicates the referenced certificate’s purpose is for dual usage; i.e. the related certificates belong to the same entity and one provides a signing-type key while the other provides an encryption-type key. The two certificates SHOULD have matching identifiers.

3.5.4. Statement of Possession of a Private Key

This purpose indicates that the Primary Certificate did not not do a full proof-of-possession at enrollment time, but instead it provided a statement of possession as per [I-D.ietf-lamps-private-key-stmt-attr] signed by the Secondary Certificate.

The reason for carrying a RelatedCertificateDescriptor of this type is to track that the Primary Certificate had a trust dependency on the Secondary Certificate at the time of issuance and that presumably the two private keys are co-located on the same key storage. Therefore if one certificate is revoked, they SHOULD both be revoked.

3.5.5. Self reference

This purpose indicates the Uniform Resource Identifier where this certificate is located. Applications which retrieve this certificate can then compare the retrieved certificate with this value to ensure that the correct certificate was retrieved.

This purpose can be used to bind the subjects of Primary and Secondary Certificates. The Primary Certificate contains a self-reference to its location, as well as a reference to the Secondary Certificate. The Secondary Certificate contains a self-reference to its location, and a reference to the Primary Certificate. Provided that policy requires subject equivalence when this mechanism is used, then the consuming application can treat both certificates as certifying the same entity.

3.6. Signature Algorithm and Public Key Algorithm fields

The signatureAlgorithm is used to indicate the signature algorithm used in the Secondary Certificate and is an optional field. The publicKeyAlgorithm indicates the public key algorithm used in the Secondary Certificate and is an optional field.

When the validation of the Primary Certificate fails, the software that understands the SIA extension and the certDiscovery access method uses the information to determine whether to fetch the Secondary Certificate. The software will look at the signatureAlgorithm and publicKeyAlgorithm to determine whether the Secondary Certificate has the signature algorithm and certificate public key algorithm it can process. If the software understands the signature algorithm and certificate public key algorithm, the software fetches the certificate from the URI specified in the relatedCertificateLocation and attempts another validation. Otherwise, the validation simply fails.

The semantics of other id-ad-certDiscovery accessLocation name forms are not defined.

Note: For a description of uniformResourceIdentifier consult section 4.2.2.1 of [!RFC5280].

4. Security Considerations

Retrieval of the Secondary Certificate is not sufficient to consider the Secondary Certificate trustworthy. The certification path validation algorithm as defined in section 6 of [RFC5280] MUST be performed for the Secondary Certificate.

The use of the self-reference purpose can be used to provide a subject binding between the Primary and Secondary Certificates. However, the procedure for validating subject equivalence MUST be defined by policy. As a result, validation of subject equivalence is out of scope of this document.

The Secondary Certificate may also have the certDiscovery access method. In order to avoid cyclic loops or infinite chaining, the validator should be mindful of how many fetching attempts it allows in one validation.

The same security considerations for caIssuers access method outlined in [RFC5280] applies to the certDiscovery access method. In order to avoid recursive certificate validations which involve online revocation checking, untrusted transport protocols (such as plaintext HTTP) are commonly used for serving certificate files. While the use

of such protocols avoids issues with recursive certification path validations and associated online revocation checking, it also enables an attacker to tamper with data and perform substitution attacks. Clients fetching certificates using the mechanism specified in this document MUST treat downloaded certificate data as untrusted and perform requisite checks to ensure that the downloaded data is not malicious.

5. IANA Considerations

TBD

6. Normative References

[I-D.ietf-lamps-private-key-stmt-attr]
Housley, R., "An Attribute for Statement of Possession of a Private Key", Work in Progress, Internet-Draft, draft-ietf-lamps-private-key-stmt-attr-09, 26 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-private-key-stmt-attr-09>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/rfc/rfc3986>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

Acknowledgments

TODO acknowledge.

Appendix A. ASN.1 Module

The following ASN.1 module provides the complete definition of the Certificate Discovery access descriptor.

```
CertDiscovery { iso(1) identified-organization(3) dod(6) internet(1)
   security(5) mechanisms(5) pkix(7) id-mod(0) id-mod-CertDiscovery(TBD1) }

   DEFINITIONS EXPLICIT TAGS ::=

   BEGIN

-- EXPORTS ALL --

   IMPORTS
    OTHER-NAME
    FROM PKIX1Implicit-2009
      { iso(1) identified-organization(3) dod(6) internet(1) security(5)
      mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-implicit-02(59) }

    id-pkix, id-ad
    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) } ;

   id-ad-certDiscovery OBJECT IDENTIFIER ::= { id-ad TBD }

   -- Other Name OID Arc --

   id-on OBJECT IDENTIFIER ::= { id-pkix 8 }

   -- Certificate Discovery Access Descriptor --

   id-on-relatedCertificateDescriptor OBJECT IDENTIFIER ::= { id-on TBD }

   on-RelatedCertificateDescriptor OTHER-NAME ::= {
      RelatedCertificateDescriptor IDENTIFIED BY id-on-relatedCertificateDescriptor
   }

   -- Purpose OBJECT IDENTIFIER
   id-rcd-agility OBJECT IDENTIFIER ::= {id-on-relatedCertificateDescriptor 1}
   id-rcd-redundency OBJECT IDENTIFIER ::= {id-on-relatedCertificateDescriptor 2}
   id-rcd-dual OBJECT IDENTIFIER ::= {id-on-relatedCertificateDescriptor 3}
   id-rcd-priv-key-stmt OBJECT IDENTIFIER ::= {id-on-relatedCertificateDescriptor 4}
   id-rcd-self OBJECT IDENTIFIER ::= {id-on-relatedCertificateDescriptor 5}

   DiscoveryPurposeId ::= OBJECT IDENTIFIER
```

```
   RelatedCertificateDescriptor ::= SEQUENCE {
      certref CertReference,
      purpose DiscoveryPurposeId,
      signatureAlgorithm [0] IMPLICIT AlgorithmIdentifier OPTIONAL,
      publicKeyAlgorithm [1] IMPLICIT AlgorithmIdentifier OPTIONAL
   }

    CertReference ::= CHOICE {
      direct Certificate,
      indirect [0] IMPLICIT CertIndirectReference
   }

   CertIndirectReference ::= SEQUENCE {
      uniformResourceIdentifier IA5String,
      certHash [0] IMPLICIT CertHash OPTIONAL
   }

   CertHash ::= SEQUENCE {
      value OCTET STRING,
      -- TODO Add IssuerAndSerialNumber?
      hashAlgorithm AlgorithmIdentifier DEFAULT {algorithm sha-256}
   }

   END
```

Authors' Addresses

Tomofumi Okubo
Penguin Securities Pte. Ltd.
Email: tomofumi.okubo+ietf@gmail.com

Corey Bonnell
DigiCert, Inc.
Email: corey.bonnell@digicert.com

John Gray
Entrust
Email: john.gray@entrust.com

Mike Ounsworth
Entrust
Email: mike.ounsworth@entrust.com

Joe Mandel
AKAYLA, Inc.
Email: joe@akayla.com