Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 21 April 2026

C. Bonnell
DigiCert
J. Gray
Entrust
D. Hook
KeyFactor
T. Okubo
DigiCert
M. Ounsworth
Entrust

18 October 2025

A Mechanism for Encoding Differences in Paired Certificates
draft-bonnell-lamps-chameleon-certs-07

Abstract

This document specifies a method to efficiently convey the differences between two certificates in an X.509 version 3 extension. This method allows a relying party to extract information sufficient to reconstruct the paired certificate and perform certification path validation using the reconstructed certificate. In particular, this method is especially useful as part of a key or signature algorithm migration, where subjects may be issued multiple certificates containing different public keys or signed with different CA private keys or signature algorithms. This method does not require any changes to the certification path validation algorithm as described in RFC 5280. Additionally, this method does not violate the constraints of serial number uniqueness for certificates issued by a single certification authority.

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://CBonnell.github.io/chameleon-certs/draft-bonnell-lamps-chameleon-certs.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-bonnell-lamps-chameleon-certs/.

Discussion of this document takes place on the Limited Additional Mechanisms for PKIX and SMIME (lamps) Working Group mailing list (mailto:spasm@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/spasm/. Subscribe at https://www.ietf.org/mailman/listinfo/spasm/.

Source for this draft and an issue tracker can be found at https://github.com/CBonnell/chameleon-certs.

Status of This Memo

This Internet-Draft will expire on 21 April 2026.

Copyright Notice

Table of Contents

1. Introduction

In certain public key infrastructures, it is common to issue multiple certificates to a single subject. In particular, as part of an algorithm migration, multiple certificates may be issued to a single subject which convey public keys of different types or are signed with different signature algorithms. In cases where relying party systems cannot be immediately updated to support new algorithms, it is useful to issue certificates to subjects that convey public keys whose algorithm is being phased out to maintain interoperability. However, multiple certificates adds complexity to certificate management for relying parties and exposes limitations in applications and protocols that support a single certificate chain. For this reason, it is useful to efficiently convey information concerning the elements of two certificates within a single certificate. This information can then be used to construct the paired certificate as needed by relying parties.

This document specifies an X.509 v3 certificate extension that includes sufficient information for a relying party to construct both paired certificates with a single certificate. This method does not require any changes to the certification path validation algorithm as described in [RFC5280]. Additionally, this method does not violate the constraints of serial number uniqueness for certificates issued by a single certification authority.

This mechanism is particularly relevant for the migration to quantum-resistant algorithms. Similar migration mechanisms have been proposed in the literature, such as the mechanism proposed in [TRANSQRPKI], where encoding the entire paired certificate in a non-critical extension is proposed. This specification builds on this idea by specifying a mechanism that requires only the differences between two paired certificates to be encoded, thus realizing a space savings.

In addition to the certificate extension, this document specifies two PKCS #10 Certificate Signing Request attributes that can be used by applicants to request Paired Certificates using a single PKCS #10 Certificate Signing Request.

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 document defines several terms that are frequently used throughout.

Base Certificate: A X.509 v3 certificate which contains a delta certificate descriptor extension.

DCD: An acronym meaning "Delta Certificate descriptor", which is a reference to the X.509 v3 certificate extension defined in this document.

Delta Certificate: A X.509 v3 certificate which can be reconstructed by incorporating the fields and extensions contained in a Base Certificate.

Paired Certificates: A Base Certificate and the corresponding Delta Certificate whose information is encoded in the Base Certificate's DCD extension.

3. Relationship between Base Certificates and Delta Certificates

In some public key infrastructures, it may be common to issue multiple certificates to the same subject. For example, these certificates generally contain the same (or substantially similar) identity information and generally have identical validity periods. The differences in certificate content generally stem from the certification of different keys, where the named subject may have multiple keys of different algorithms certified by separate certificates. The use of different keys allows for the subject to use the key that is most appropriate for a given operation and intended recipient. For example, as part of an ongoing algorithm migration, it is useful to use stronger algorithms when both of the systems utilized by the subscriber/sender and recipient have been upgraded. However, in the case where systems have not yet been updated, the use of a legacy key algorithm may be required. Additionally, multiple certificates may be issued to the same subject that certify keys for different purposes, such as one key for signing and another key for encryption.

The management of multiple certificates may be complex, and there may be limitations in protocols regarding the handling of multiple certificate chains. To account for these concerns, this document proposes a method to efficiently encode the differences between two certificates with sufficient information such that a relying party can derive the complete certificate from another. For the purposes of this document, the "Base Certificate" contains its own fields and extensions and additionally includes an extension that conveys all differences contained within the paired certificate. The certificate whose elements which differ from the Base Certificate and are captured in the Delta Certificate descriptor extension of the Base Certificate is known as the "Delta Certificate".

Delta Certificates are reconstructed from the Base Certificate either on the sender's side or the recipient's side depending on the protocol and application(s) in use. The sender may elect to send the Base Certificate or the Delta Certificate based on information that it has about what the recipient can process. Similarly, the client may send either the Base Certificate or the Delta Certificate based on what the server can process. This assures backwards compatibility as the certificate sent to the peer (server or client) is chosen based on what it can process. The negotiation on which certificate to use is out-of-scope of this document and is deferred to each protocol and application.

In the absence of information concerning the capabilities of the peer, it is unknown whether it understands the DCD extension in the Base Certificate. When the recipient does not understand the DCD extension, it only processes the information within the Base Certificate and ignores the information found in a non-critical DCD extension. If the recipient receives a Base Certificate and is capable of processing the DCD extension, then it may reconstruct the Delta Certificate to be used for processing.

In a protocol, the sender may perform a cryptographic operation with the key conveyed within the Base Certificate. If it understands the DCD extension, then it may reconstruct the Delta Certificate and choose to perform the same operation with the key conveyed within the DCD extension. Alternatively, if the sender understands the DCD extension and knows that the receiver will only process the Delta Certificate, the sender can reconstruct and send only the Delta Certificate. This behavior is deferred to the software in use.

4. Delta certificate descriptor extension

The Delta Certificate descriptor ("DCD") extension is used to reconstruct the Delta Certificate by incorporating both the fields and extensions present in the Base Certificate as well as the information contained within the extension itself.

Certification authorities SHOULD NOT mark this extension as critical so that applications that do not understand the extension will still be able to process the Base Certificate.

The inclusion of the DCD extension within a Base Certificate is not a statement from the issuing Certification Authority of the Base Certificate that the contents of the Delta Certificate have been verified. Conversely, the DCD extension is merely a mechanism to encode the differences between two Paired Certificates. Given this, it is possible for the Base Certificate to expire prior to the Delta Certificate, and vice versa. However, the policies governing a public key infrastructure may add additional requirements for the content of the DCD extension or alignment of validity periods for Base Certificates and Delta Certificates. For example, a policy may require that the validity periods of the Base Certificate and Delta Certificate be identical, or that if the Delta Certificate is revoked, the Base Certificate must also be revoked.

4.1. Delta certificate descriptor content

The DCD extension is identified with the following object identifier:

(TODO: replace this temporary OID)

```
id-ce-deltaCertificateDescriptor OBJECT IDENTIFIER ::= {
   joint-iso-itu-t(2) country(16) us(840) organization(1)
   entrust(114027) 80 6 1
}
```

The ASN.1 syntax of the extension is as follows:

```
DeltaCertificateDescriptor ::= SEQUENCE {
  serialNumber          CertificateSerialNumber,
  signature             [0] EXPLICIT AlgorithmIdentifier
       {SIGNATURE_ALGORITHM, {...}} OPTIONAL,
  issuer                [1] EXPLICIT Name OPTIONAL,
  validity              [2] EXPLICIT Validity OPTIONAL,
  subject               [3] EXPLICIT Name OPTIONAL,
  subjectPublicKeyInfo  SubjectPublicKeyInfo,
  extensions            [4] EXPLICIT Extensions{CertExtensions}
       OPTIONAL,
  signatureValue        BIT STRING
}
```

The serialNumber field MUST be present and contain the serial number of the Delta Certificate.

The signature field specifies the signature algorithm used by the issuing certification authority to sign the Delta Certificate. If the DER encoding of the value of the signature field of the Base Certificate and Delta Certificate is the same, then this field MUST be absent. Otherwise, it MUST contain the DER encoding of the value of the signature field of the Delta Certificate.

The issuer field specifies the distinguished name of the issuing certification authority which signed the Delta Certificate. If the DER encoding of the value of the issuer field of the Base Certificate and Delta Certificate is the same, then this field MUST be absent. Otherwise, it MUST contain the DER encoding of the value of the issuer field of the Delta Certificate.

The validity field specifies the validity period of the Delta Certificate. If the DER encoding of the value of the validity field of the Base Certificate and Delta Certificate is the same, then this field MUST be absent. Otherwise, it MUST contain the DER encoding of the value of the validity field of the Delta Certificate.

The subject field specifies the distinguished name of the named subject as encoded in the Delta Certificate. If the DER encoding of the value of the subject field of the Base Certificate and Delta Certificate is the same, then this field MUST be absent. Otherwise, it MUST contain the DER encoding of the value of the subject field of the Delta Certificate.

The subjectPublicKeyInfo field contains the public key certified in the Delta Certificate. The value of this field MUST differ from the value of the subjectPublicKeyInfo field of the Base Certificate. In other words, the Base Certificate and Delta Certificate MUST certify different keys.

The extensions field contains the extensions whose criticality and/or DER-encoded value are different in the Delta Certificate compared to the Base Certificate with the exception of the DCD extension itself. If the extensions field is absent, then all extensions in the Delta Certificate MUST have the same criticality and DER-encoded value as the Base Certificate (except for the DCD extension, which MUST be absent from the Delta Certificate). This field MUST NOT contain any extension:

* which has the same criticality and DER-encoded value as encoded in the Base Certificate,

* whose type does not appear in the Base Certificate, or

* which is of the DCD extension type (recursive Delta Certificates are not permitted).

Additionally, the Base Certificate SHALL NOT include any extensions which are not included in the Delta Certificate, with the exception of the DCD extension itself. Likewise, there is no mechanism to remove extensions from the Delta Certificate that are present in the Base Certificate. Therefore, it is not possible to add or remove extensions using the DCD extension. The ordering of extensions in this field MUST be relative to the ordering of the extensions as they are encoded in the Delta Certificate. Maintaining this relative ordering ensures that the Delta Certificate’s extensions can be reconstructed with a single pass.

The signatureValue field contains the value of the signature field of the Delta Certificate. It MUST be present.

4.2. Issuing a Base Certificate

The signature of the Delta Certificate must be known so that its value can be included in the signatureValue field of the delta certificate descriptor extension. Given this, Delta Certificate will necessarily need to be issued prior to the issuance of the Base Certificate. To simplify reconstruction of the Delta Certificate, the signatures for Base and Delta Certificates MUST be calculated over the DER encoding of the TBSCertificate structure.

After the Delta Certificate is issued, the certification authority compares the signature, issuer, validity, subject, subjectPublicKeyInfo, and extensions fields of the Delta Certificate and the to-be-signed certificate which will contain the DCD extension. The certification authority then populates the DCD extension with the values of the fields which differ from the Base Certificate. The CA MUST encode extensions in the Base Certificate in the same order used for the Delta Certificate, with the exception of the DCD extension itself.

The certification authority then adds the computed DCD extension to the to-be-signed Base Certificate and signs the Base Certificate.

4.3. Reconstructing a Delta Certificate from a Base Certificate

The following procedure describes how to reconstruct a Delta Certificate from a Base Certificate:

1. Create an initial Delta Certificate template by copying the Base Certificate excluding the DCD extension.

2. Replace the value of the serialNumber field of the Delta Certificate template with the value of the DCD extension’s serialNumber field.

3. If the DCD extension contains a value for the signature field, then replace the value of the signature field and the signatureAlgorithm field of the Delta Certificate template with the value of the DCD extension’s signature field.

4. If the DCD extension contains a value for the issuer field, then replace the value of the issuer field of the Delta Certificate template with the value of the DCD extension’s issuer field.

5. If the DCD extension contains a value for the validity field, then replace the value of the validity field of the Delta Certificate template with the value of the DCD extension’s validity field.

6. Replace the value of the subjectPublicKeyInfo field of the Delta Certificate template with the value of the DCD extension’s subjectPublicKeyInfo field.

7. If the DCD extension contains a value for the subject field, then replace the value of the subject field of the Delta Certificate template with the value of the DCD extension’s subject field.

8. If the DCD extension contains a value for the extensions field, then iterate over the DCD extension’s "extensions" field, replacing the criticality and/or extension value of each identified extension in the Delta Certificate template. If any extension is present in the field that does not appear in the Delta Certificate template, then this reconstruction process MUST fail.

9. Replace the value of the signature field of the Delta Certificate template with the value of the DCD extension’s signatureValue field.

As part of testing implementations of this specification, implementers are encouraged to verify the signature of the reconstructed Delta Certificate using the issuing Certification Authority’s public key to ensure that the Delta Certificate was reconstructed correctly.

5. Delta certificate request content and semantics

Using the two attributes that are defined below, it is possible to create Certificate Signing Requests for both Base and Delta Certificates within a single PKCS #10 Certificate Signing Request. The mechanism presented in this section need not be used exclusively by requestors for the issuance of Paired Certificates; other mechanisms (such as the submission of two Certificate Signing Requests, etc.) are also acceptable. Additionally, this document does not place any restriction on the amount of time that may elapse between the issuance of a Delta Certificate and the request of a Base Certificate; such restrictions should be defined by the policy of a particular public key infrastructure.

The delta certificate request attribute is used to convey the requested differences between the request for issuance of the Base Certificate and the requested Delta Certificate. Similar to the semantics of Certificate Signing Requests in general, the Certification Authority MAY add, modify, or selectively ignore information conveyed in the attribute when issuing the corresponding Delta Certificate.

The attribute is identified with the following object identifier:

(TODO: replace this temporary OID)

```
id-at-deltaCertificateRequest OBJECT IDENTIFIER ::= {
   joint-iso-itu-t(2) country(16) us(840) organization(1)
   entrust(114027) 80 6 2
}
```

The ASN.1 syntax of the attribute is as follows:

```
DeltaCertificateRequestValue ::= SEQUENCE {
  subject               [0] EXPLICIT Name OPTIONAL,
  subjectPKInfo         SubjectPublicKeyInfo,
  extensions            [1] EXPLICIT Extensions{CertExtensions}
       OPTIONAL,
  signatureAlgorithm    [2] EXPLICIT AlgorithmIdentifier
       {SIGNATURE_ALGORITHM, {...}} OPTIONAL
}

DeltaCertificateRequest ::= ATTRIBUTE {
   WITH SYNTAX DeltaCertificateRequestValue
   SINGLE VALUE TRUE
   ID id-at-deltaCertificateRequest
}
```

The delta certificate request signature attribute is used to convey the signature that is calculated over the CertificationRequestInfo using the signature algorithm and key that is specified in the delta certificate request attribute. Section 5.1 describes in detail how to determine the value of this attribute.

This attribute is identified with the following object identifier:

(TODO: replace this temporary OID)

```
id-at-deltaCertificateRequestSignature OBJECT IDENTIFIER ::= {
   joint-iso-itu-t(2) country(16) us(840) organization(1)
   entrust(114027) 80 6 3
}
```

The ASN.1 syntax of the attribute is as follows:

```
DeltaCertificateRequestSignatureValue ::= BIT STRING

deltaCertificateRequestSignature ATTRIBUTE ::= {
   WITH SYNTAX DeltaCertificateRequestSignatureValue
   SINGLE VALUE TRUE
   ID id-at-deltaCertificateRequestSignature
}
```

5.1. Creating a Certificate Signing Request for Paired Certificates

The following procedure is used by a certificate requestor to create a combined Certificate Signing Request for Paired Certificates.

1. Create a CertificationRequestInfo containing the subject, subjectPKInfo, and attributes for the Base Certificate.

2. Create a delta certificate request attribute that specifies the requested differences between the to-be-issued Base Certificate and Delta Certificate requests.

3. Add the delta certificate request attribute that was created by step 2 to the list of attributes in the CertificationRequestInfo.

4. Sign the CertificationRequestInfo using the private key of the delta certificate request subject.

5. Create a delta certificate request signature attribute that contains the signature value calculated by step 4.

6. Add the delta certificate request signature attribute that was created by step 5 to the list of attributes.

7. Sign the CertificationRequestInfo using the private key of the base certificate request subject.

5.2. Verifying a Certificate Signing Request for Paired Certificates

The following procedure is used by a Certification Authority to verify a Certificate Signing Request for Paired Certificates that was created using the process outlined in Section 5.1.

1. Create a CertificationRequest template by copying the CertificationRequest submitted by the certificate requestor.

2. Verify the signature of the base certificate request using the public key associated with the base certificate request subject and the signature algorithm specified in the signatureAlgorithm field of the CertificationRequest template. If signature verification fails, then the Certification Authority MUST treat the Certificate Signing Request as invalid.

3. Remove the delta certificate request signature attribute from the CertificationRequest template.

4. Replace the value of the signature field of the CertificationRequest template with the value of the delta certificate request attribute that was removed in step 3.

5. Verify the signature of the delta certificate request using the public key associated with the delta certificate request subject. If the signatureAlgorithm field of the delta certificate request attribute is present, then the Certification Authority MUST perform signature verification using the algorithm specified in this field. Otherwise, the Certification Authority MUST perform signature verification using the algorithm specified in the signatureAlgorithm field of the CertificationRequest template. If signature verification fails, then the Certification Authority MUST treat the Certificate Signing Request as invalid.

6. Security Considerations

The validation of Base Certificates and Delta Certificates follows the certification path validation algorithm defined in [RFC5280]. In particular, the certification path validation algorithm defined in [RFC5280] MUST be performed prior to using a Base or Delta Certificate; it is not sufficient to reconstruct a Delta Certificate and use it for any purpose without performing certification path validation. If a use case requires it, a Delta Certificate can be reconstructed specifically for the purposes of validation to ensure that the Delta Certificate is valid for its intended purpose on final reconstruction. That being said, some form of validation such as revocation checking, and signature verification MUST always be assured at the point the certificate is used.

There are some additional considerations for the software to handle the Base Certificate and Delta Certificate. The Base Certificate and Delta Certificate may have different security properties such as different signing algorithms, different key types or the same key types with different key sizes or signing algorithms. The preference on which certificate to be used or using both when available is deferred to the server or client software.

The software is expected to make choices depending on the certificate's security properties or a policy set for the particular PKI. One example of handling two certificates is "fallback" where if the validation of the first certificate fails, it attempts to validate the second certificate. Another example to handle two certificate is "upgrade", where the validation of the first certificate succeeds but still attempts the validation of the second certificate. While this document provides a vehicle to convey information of two certificates in one, it does not address the rules that are expected to be set by the policy of a PKI on how to issue Paired Certificates. Likewise, this document does not establish how Paired Certificates are processed by certificate-consuming applications.

The algorithms that are used for the Base Certificate and Delta Certificate respectively should be carefully set by the policy of each PKI reflecting the best current practices in usage of cryptography. The behavior of the server or client software is expected to be well-defined in accordance with the policy in order to avoid downgrade attacks or substitution attacks.

7. IANA Considerations

For the Delta Certificate descriptor extension as defined in Section 4.1, IANA is requested to assign an object identifier (OID) for the certificate extension. The OID for the certificate extension should be allocated in the "SMI Security for PKIX Certificate Extension" registry (1.3.6.1.5.5.7.1).

For the Delta Certificate Request and Delta Certificate Request Signature attributes as defined in Section 5, IANA is requested to create a new registry under SMI Security Codes and assign two object identifiers (OID).

For the ASN.1 Module for the extension and attributes defined in Appendix A, IANA is requested to assign an object identifier (OID). The OID for the module should be allocated in the "SMI Security for PKIX Module Identifier" registry (1.3.6.1.5.5.7.0).

8. References

8.1. 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/rfc/rfc2119>.

[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>.

[X.680] ITU-T, "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation", ISO/IEC 8824-1:2015, November 2015.

8.2. Informative References

[TRANSQRPKI]
Bindel, N., Herath, U., McKague, M., and D. Stebila, "Transitioning to a Quantum-Resistant Public Key Infrastructure", May 2017, <https://d1kjwivbowugqa.cloudfront.net/files/research/papers/PQCrypto-BHMS17-full.pdf>.

Appendix A. ASN.1 Module

The following ASN.1 [X.680] module provides the complete definition of the extensions, attributes, and associated identifiers specified in this document.

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

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

EXPORTS ALL;

IMPORTS
  AlgorithmIdentifier{}, SIGNATURE-ALGORITHM
  FROM AlgorithmInformation-2009  -- RFC 5912
  { iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) id-mod(0)
    id-mod-algorithmInformation-02(58) }

  EXTENSION, ATTRIBUTE, Extensions{}
  FROM PKIX-CommonTypes-2009  -- 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) }

    CertificateSerialNumber, Name, Validity, SubjectPublicKeyInfo,
    CertExtensions FROM PKIX1Explicit-2009  -- RFC 5912
    { iso(1) identified-organization(3) dod(6) internet(1) security(5)
      mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-explicit-02(51) };

  -- Temporary OID arc --

  id-temporaryArc OBJECT IDENTIFIER ::= {
    joint-iso-itu-t(2) country(16) us(840) organization(1)
    entrust(114027) 80 6
  }

  -- Extension --

  id-ce-deltaCertificateDescriptor OBJECT IDENTIFIER ::= {
         id-temporaryArc 1 }

  DeltaCertificateDescriptor ::= SEQUENCE {
    serialNumber          CertificateSerialNumber,
    signature             [0] EXPLICIT AlgorithmIdentifier
         {SIGNATURE_ALGORITHM, {...}} OPTIONAL,
    issuer                [1] EXPLICIT Name OPTIONAL,
    validity              [2] EXPLICIT Validity OPTIONAL,
    subject               [3] EXPLICIT Name OPTIONAL,
    subjectPublicKeyInfo  SubjectPublicKeyInfo,
    extensions            [4] EXPLICIT Extensions{CertExtensions}
         OPTIONAL,
    signatureValue        BIT STRING
  }

  ext-deltaCertificateDescriptor EXTENSION ::= {
    SYNTAX DeltaCertificateDescriptor
    IDENTIFIED BY id-ce-deltaCertificateDescriptor
    CRITICALITY { FALSE }
  }

  -- Request Attributes --

  id-at-deltaCertificateRequest OBJECT IDENTIFIER ::= {
         id-temporaryArc 2 }

  DeltaCertificateRequestValue ::= SEQUENCE {
    subject               [0] EXPLICIT Name OPTIONAL,
    subjectPKInfo         SubjectPublicKeyInfo,
```

```
     extensions            [1] EXPLICIT Extensions{CertExtensions}
          OPTIONAL,
     signatureAlgorithm    [2] EXPLICIT AlgorithmIdentifier
          {SIGNATURE_ALGORITHM, {...}} OPTIONAL
   }

   DeltaCertificateRequest ::= ATTRIBUTE {
      WITH SYNTAX DeltaCertificateRequestValue
      SINGLE VALUE TRUE
      ID id-at-deltaCertificateRequest
   }

   id-at-deltaCertificateRequestSignature OBJECT IDENTIFIER ::= {
          id-temporaryArc 3 }

   DeltaCertificateRequestSignatureValue ::= BIT STRING

   DeltaCertificateRequestSignature ::= ATTRIBUTE {
      WITH SYNTAX DeltaCertificateRequestSignatureValue
      SINGLE VALUE TRUE
      ID id-at-deltaCertificateRequestSignature
   }

   END
```

Appendix B. Examples

This appendix includes some example certificates which demonstrate the use of the mechanism specified in this document. Two use cases of this mechanism are demonstrated: algorithm migration and dual use. The PEM text and dumpasn1 output for each certificate is provided.

B.1. Root certificates

The two certificates in this section represent the two root Certification Authorities which issue the end-entity certificates in the following section.

B.1.1. EC P-521 root certificate

This is the EC root certificate.

```
   -----BEGIN CERTIFICATE-----
   MIIDBDCCAmagAwIBAgIUDCQO4j68JeS6tggSujZ2W/+5RMAwCgYIKoZIzj0EAwQw
   gYsxCzAJBgNVBAYTAlhYMTUwMwYDVQQKDCxSb3lhbCBJbnN0aXR1dGUgb2YgUHVi
   bGljIEtleSBJbmZyYXN0cnVjdHVyZTErMCkGA1UECwwiUG9zdC1IZWZmYWx1bXAg
   UmVzZWFyY2ggRGVwYXJ0bWVudDEYMBYGA1UEAwwPRUNEU0EgUm9vdCAtIEcxMB4X
   DTI0MTAxNzIzMzcyM1oXDTM0MTAxNTIzMzcyM1owgYsxCzAJBgNVBAYTAlhYMTUw
   MwYDVQQKDCxSb3lhbCBJbnN0aXR1dGUgb2YgUHVibGljIEtleSBJbmZyYXN0cnVj
   dHVyZTErMCkGA1UECwwiUG9zdC1IZWZmYWx1bXAgUmVzZWFyY2ggRGVwYXJ0bWVu
   dDEYMBYGA1UEAwwPRUNEU0EgUm9vdCAtIEcxMIGbMBAGByqGSM49AgEGBSuBBAAj
   A4GGAAQBAFYGp79DhDUnJ+euhbWIqRMPC/YJyMcXp5xEF96cQji2rOckvcqQkhqE
   K2upXcSLaclIkS16REFZgT0q3vO2m1wAhXxeKePsML2EiCMQIEArXsEwCDGu+qdx
   mN2lHUQNuiisrkigRdXILHaAXdfTtAvpopsAchnm+vUbHNavcxVRjK2jYzBhMA8G
   A1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBTro9CLUf4S
   3MwhZoeFD5jHZ3OINDAfBgNVHSMEGDAWgBTro9CLUf4S3MwhZoeFD5jHZ3OINDAK
   BggqhkjOPQQDBAOBiwAwgYcCQUnnSxI6X5NPGGetpBUkEh3HIDTrW24dPtx74wmW
   ANwrejsbS0SvbipnQJPQXjTv8aXDlDAMiPKHado5qCJXMvU3AkIAmDbRmevtaNUQ
   0k6e97CWc8tTPE7gXo5iqFD0NU9v20HV3z7voEU8fYD65A1Ay3VQ76nC8W8T4T1a
   fvRCLit6wo0=
   -----END CERTIFICATE-----

  0 772: SEQUENCE {
  4 614:   SEQUENCE {
  8   3:     [0] {
 10   1:       INTEGER 2
      :       }
 13  20:     INTEGER 0C 24 0E E2 3E BC 25 E4 BA B6 08 12 BA 36 76 5B FF B9 44 C0
 35  10:     SEQUENCE {
 37   8:       OBJECT IDENTIFIER ecdsaWithSHA512 (1 2 840 10045 4 3 4)
      :       }
 47 139:     SEQUENCE {
 50  11:       SET {
 52   9:         SEQUENCE {
 54   3:           OBJECT IDENTIFIER countryName (2 5 4 6)
 59   2:           PrintableString 'XX'
      :           }
      :         }
 63  53:       SET {
 65  51:         SEQUENCE {
 67   3:           OBJECT IDENTIFIER organizationName (2 5 4 10)
 72  44:           UTF8String
      :             'Royal Institute of Public Key Infrastructure'
      :           }
      :         }
118  43:       SET {
120  41:         SEQUENCE {
122   3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
127  34:           UTF8String 'Post-Heffalump Research Department'
      :           }
```

```
       :      }
163  24:    SET {
165  22:     SEQUENCE {
167   3:      OBJECT IDENTIFIER commonName (2 5 4 3)
172  15:      UTF8String 'ECDSA Root - G1'
       :      }
       :     }
       :    }
189  30:   SEQUENCE {
191  13:    UTCTime 17/10/2024 23:37:23 GMT
206  13:    UTCTime 15/10/2034 23:37:23 GMT
       :    }
221 139:   SEQUENCE {
224  11:    SET {
226   9:     SEQUENCE {
228   3:      OBJECT IDENTIFIER countryName (2 5 4 6)
233   2:      PrintableString 'XX'
       :      }
       :     }
237  53:    SET {
239  51:     SEQUENCE {
241   3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
246  44:      UTF8String
       :       'Royal Institute of Public Key Infrastructure'
       :      }
       :     }
292  43:    SET {
294  41:     SEQUENCE {
296   3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
301  34:      UTF8String 'Post-Heffalump Research Department'
       :      }
       :     }
337  24:    SET {
339  22:     SEQUENCE {
341   3:      OBJECT IDENTIFIER commonName (2 5 4 3)
346  15:      UTF8String 'ECDSA Root - G1'
       :      }
       :     }
       :    }
363 155:   SEQUENCE {
366  16:    SEQUENCE {
368   7:     OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
377   5:     OBJECT IDENTIFIER secp521r1 (1 3 132 0 35)
       :      }
384 134:    BIT STRING
       :     04 01 00 56 06 A7 BF 43 84 35 27 27 E7 AE 85 B5
       :     88 A9 13 0F 0B F6 09 C8 C7 17 A7 9C 44 17 DE 9C
       :     42 38 B6 AC E7 24 BD CA 90 92 1A 84 2B 6B A9 5D
```

```
       :     C4 8B 69 C9 48 91 2D 7A 44 41 59 81 3D 2A DE F3
       :     B6 9B 5C 00 85 7C 5E 29 E3 EC 30 BD 84 88 23 10
       :     20 40 2B 5E C1 30 08 31 AE FA A7 71 98 DD A5 1D
       :     44 0D BA 28 AC AE 48 A0 45 D5 C8 2C 76 80 5D D7
       :     D3 B4 0B E9 A2 9B 00 72 19 E6 FA F5 1B 1C D6 AF
       :     73 15 51 8C AD
       :     }
521  99:   [3] {
523  97:    SEQUENCE {
525  15:     SEQUENCE {
527   3:      OBJECT IDENTIFIER basicConstraints (2 5 29 19)
532   1:      BOOLEAN TRUE
535   5:      OCTET STRING, encapsulates {
537   3:       SEQUENCE {
539   1:        BOOLEAN TRUE
       :         }
       :        }
       :       }
542  14:     SEQUENCE {
544   3:      OBJECT IDENTIFIER keyUsage (2 5 29 15)
549   1:      BOOLEAN TRUE
552   4:      OCTET STRING, encapsulates {
554   2:       BIT STRING 1 unused bit
       :        '1100000'B
       :        }
       :       }
558  29:     SEQUENCE {
560   3:      OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
565  22:      OCTET STRING, encapsulates {
567  20:       OCTET STRING
       :        EB A3 D0 8B 51 FE 12 DC CC 21 66 87 85 0F 98 C7
       :        67 73 88 34
       :        }
       :       }
589  31:     SEQUENCE {
591   3:      OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
596  24:      OCTET STRING, encapsulates {
598  22:       SEQUENCE {
600  20:        [0]
       :         EB A3 D0 8B 51 FE 12 DC CC 21 66 87 85 0F 98 C7
       :         67 73 88 34
       :         }
       :        }
       :       }
       :      }
       :     }
       :    }
622  10:  SEQUENCE {
```

```
624   8:   OBJECT IDENTIFIER ecdsaWithSHA512 (1 2 840 10045 4 3 4)
       :    }
634 139:  BIT STRING, encapsulates {
638 135:   SEQUENCE {
641  65:    INTEGER
       :     49 E7 4B 12 3A 5F 93 4F 18 67 AD A4 15 24 12 1D
       :     C7 20 34 EB 5B 6E 1D 3E DC 7B E3 09 96 00 DC 2B
       :     7A 3B 1B 4B 44 AF 6E 2A 67 40 93 D0 5E 34 EF F1
       :     A5 C3 94 30 0C 88 F2 87 69 DA 39 A8 22 57 32 F5
       :     37
708  66:    INTEGER
       :     00 98 36 D1 99 EB ED 68 D5 10 D2 4E 9E F7 B0 96
       :     73 CB 53 3C 4E E0 5E 8E 62 A8 50 F4 35 4F 6F DB
       :     41 D5 DF 3E EF A0 45 3C 7D 80 FA E4 0D 40 CB 75
       :     50 EF A9 C2 F1 6F 13 E1 3D 5A 7E F4 42 2E 2B 7A
       :     C2 8D
       :     }
       :    }
       :   }
```

B.1.2. ML-DSA-65 root certificate

This is the ML-DSA-65 root certificate. It contains a Delta Certificate Descriptor extension which includes sufficient information to recreate the ECDSA P-521 root.

```
-----BEGIN CERTIFICATE-----
MIIZCDCCDAWgAwIBAgIUFWd6hCxGhDNL+S1OL3UY7w+psbQwCwYJYIZIAWUDBAMS
MIGMMQswCQYDVQQGEwJYWDE1MDMGA1UECgwsUm95YWwgSW5zdGl0dXRlIG9mIFB1
YmxpYyBLZXkgSW5mcmFzdHJ1Y3R1cmUxKzApBgNVBAsMIlBvc3QtSGVmZmFsdW1w
IFJlc2VhcmNoIERlcGFydG1lbnQxGTAXBgNVBAMMEE1MLURTQSBSb290IC0gRzEw
HhcNMjQxMDE3MjMzNzIzWhcNMzQxMDE1MjMzNzIzWjAvMQswCQYDVQQGEwJYWDEP
MA0GA1UECgwGSGFuYWtvMQ8wDQYDVQQLDAZZYW1hZGEwggeyMAsGCWCGSAFlAwQD
EgOCB6EA/a6iHTzCfanvaHi8GU+U+oX5nDkvkSj/c/eGnGt0f70YDjvXoNmwXSxI
pFHz7mLnmJ09lEI2O1OGLgUFjAYdubQRMlvjj0OzZjD4gJhs/c6G8B2loKtd6aOW
t4KPPVpmmvXaOFwFeU3NVq+JYZh8Uk7dCQ6PNC6FqIirE+5X8EqoG1SvOe8jYDt+
KVu7Q9VKSNMEwZYoa9lE/JDlQ+CTr3LsC4XbpHGFzxlbXBz2hPv9Rq+K5JGKZ8Xe
WiEvJ0gwU9BuEJ1zwA/mKO/gmYcxFwVT24aaoW4nDwZ4eyUmDxdH6y1PK7Anwbbm
Izis+50FisFbSCv+FCg+M4Bwa+VFLcnG2RB31dwR6Qx7WOyMhYEXycQIPwD1GRLO
zJopFVAoeWFIUec28gSLPG/a6WtZthVf3AwNlEU9q5GVMyNXi4wikwr3oMW0JHea
Muv1dN+QEHPKFUP/pVdrux+39EdUP2Ydh0m4xB4cPTYD3b0PEtNvlwWvprP1S6yE
GbbFNCqXKoPiGLeTgEiKIzDr9b0DxFMQhjzHBEKxnTbpFpYTuFuUApUAnmm6DzUj
o1V1RdsIqG5B1ZEGaJxHkBZle7Xr6ALw9Pr8ypmaUNnjq9E4ZxoqBWhuWQLsomB3
gLXrXcTTZGo/Yr603KZacjL0fitoplczRu91crEqLV4d7s1jNZKKRC+0p2cYDcIn
ktjgf8ypkgElWo+GdgoMmUdqtzpORIbq3pWV3AJfL9PU8tB6KTCN86Rxb/syTG7Y
53TLW67qyY6xY/BGvdxtdjw7zNkkOhO6AUywFn8xi5+B9S3n11ICSXrn6Rpu+Qgt
z1GKl53ltc97Gv19eFn48u7uPwkqtHVyvRqCzEMBygZKcy2Weaij3gsXW+JWYxtP
zJgrn9fA3zGia9CWPe/+W0xGuIm0nXf4H85rrJ8Iwgh7lwlQeoWaXFIzxtj5YSks
```

ksZGy0YcW1QfmUGpeu5j6qLBuLfNyKgNpRxMbqC50UQh0hSJkOr9mwEPEL7A0LJG
SlRnqNo7gCmhxTRKyaNR3luLZbIJfj4j3ltlPCDCRBrIFXkTbAiE8tfillgWqnCK
phdQHtzLRvFPaFpCDMhXuAyjLXW/6JhERClap5dJFjYVH0YWvWffcRPChusVSSLh
lTGxD5QYe8S0EkECJ1ajlUCyRf1kOPejJSh/gsW0DLvYW3vIypDJhnDcD40CPqqz
B5YduFeFGkYLnJs6LA1JyosJGkblcUA5O+eqz7AIdK02GPLL5eDAeQ/HCXgIa8AN
nXa3eErpAv7V6TCIdUpUdAMzTSf1E5x30uas4PBbF85MZWTP5xH2eGHmaNvi2UQd
Zg3FVfWa4hv99IIlBPqDykd60RHKrWSXwB16R0zzzLy8vcIDPMrC9M2rOGkJF2w1
KZ6u86/PGUVvwESxz+7jpxkMFdsgQ79ffhJYxjJNBY0EEe9C3vz4+vIDm0PW7YhR
BZ11pWYH+4v8WzhT1HmGIfq6QQYh9Pz9uHsNc9Pl5kTQu1jTgdSeqFbshUNWuvGo
LeGhoIC6fn5cYKJhKEslFl4fcJr5eQfVzwhXncZkloHgrrFYprv35Odqo/2AAPu6
+ILsegIY+D809nrQWBsZLNs7gopYi68xHc/yA6vOFGu7hLk/6qeV8ZNr89aGQU3Q
WBLmFRCZ+c4325uNmr2uhZntOK2ELVDosHkMkxynQPKDXheUGhU2lP/3bT+aGsug
l+0eXYYrxR3qnc5fM8u9ntJUGDTV9z/CzcBovHdMJfR16gbs4Qi5tsrwp1Teu8cI
SKh1zgdlxXKk/F7D0q3xiMcS8RC4fp3wstOVzgaQNrJEgAcN3tNsblu1JYKoSwoe
Rg94FAvULY8PLlkntNvjWAqAQXr1UW/diWX20wSUG9qFCmyo/TFpUbXIUER1kwi4
j/Ll5tLfziGWWsnAehE8heeZSkeKwa63ZBjUAZzQ2Ycohd3I+yzf3sbL6q4v4qtT
tpRSJh4f25YJEg4DPYCUxpS1ZQyhZ7yG77ylKgw2+pA1Kv3wArO/3pWff+NZoP9O
LKDrOmNY2ythYRxjfvwerjAl93vcnqm/t84XY1PNWk4uvf5M3ZqBF4skeMAtTdsD
y1ROQuAMBvnuXdg58cBDe6RSInoJyn8sZYILyveHt9bK4E9O3icOfCMaCweeJ30x
SDOrW9bdB7jA3WHqzSrg1NqJJGUwCQTYdmbACzOsiDaduUZAB4BaD666bzcSUjHf
QEN5/930H4wFNyRBAcrjlyJ0s3SCDG+8RX0EVN8QFt+A+0VFPpkTeIqsOzHaSwjs
WhALemYVCFpuqCc9ILDh7LmGNPkAUFpVjCGOWiNTxP1jhiQWp/gR7fF3Y/8tpFF0
/Le6X7p4VysXETscgXtopK/eT4eRo08wI2bkR7V8E9eiV+djFJw9uiHxmYH2HN2X
xVNWRGnpZRUiHzHIm0mQwmaamkaz+v8D8zNa2EC8PNcE1UAjqiKEcxOm3pBcFozT
BnDx5gJ0zcOP29+DZWKDNNHwr+ldHK51ttXeBn44LVC7CsG8CcN+/hMK0oGs7ivE
eC4YVfpeimX5k4ZfG0BfQAEVSUkgQL5d0VyW3Ceiunzc4wbYHTOjggNDMIIDPzAP
BgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUmwe0pHXE
vJFdNeDJocFi4ndV1j8wHwYDVR0jBBgwFoAUmwe0pHXEvJFdNeDJocFi4ndV1j8w
ggLaBgpghkgBhvprUAYBBIICyjCCAsYCFAwkDuI+vCXkurYIEro2dlv/uUTAoAww
CgYIKoZIzj0EAwShgY4wgYsxCzAJBgNVBAYTAlhYMTUwMwYDVQQKDCxSb3lhbCBJ
bnN0aXR1dGUgb2YgUHVibGljIEtleSBJbmZyYXN0cnVjdHVyZTErMCkGA1UECwwi
UG9zdC1IZWZmYWx1bXAgUmVzZWFyY2ggRGVwYXJ0bWVudDEYMBYGA1UEAwwPRUNE
U0EgUm9vdCAtIEcxo4GOMIGLMQswCQYDVQQGEwJYWDE1MDMGA1UECgwsUm95YWwg
SW5zdGl0dXRlIG9mIFB1YmxpYyBLZXkgSW5mcmFzdHJ1Y3R1cmUxKzApBgNVBAsM
IlBvc3QtSGVmZmFsdW1wIFJlc2VhcmNoIERlcGFydG1lbnQxGDAWBgNVBAMMD0VD
RFNBIFJvb3QgLSBHMTCBmzAQBgcqhkjOPQIBBgUrgQQAIwOBhgAEAQBWBqe/Q4Q1
JyfnroW1iKkTDwv2CcjHF6ecRBfenEI4tqznJL3KkJIahCtrqV3Ei2nJSJEtekRB
WYE9Kt7ztptcAIV8Xinj7DC9hIgjECBAK17BMAgxrvqncZjdpR1EDboorK5IoEXV
yCx2gF3X07QL6aKbAHIZ5vr1GxzWr3MVUYytpFIwUDAOBgNVHQ8BAf8EBAMCAQYw
HQYDVR0OBBYEFOuj0ItR/hLczCFmh4UPmMdnc4g0MB8GA1UdIwQYMBaAFOuj0ItR
/hLczCFmh4UPmMdnc4g0A4GLADCBhwJBSedLEjpfk08YZ62kFSQSHccgNOtbbh0+
3HvjCZYA3Ct6OxtLRK9uKmdAk9BeNO/xpcOUMAyI8odp2jmoIlcy9TcCQgCYNtGZ
6+1o1RDSTp73sJZzy1M8TuBejmKoUPQ1T2/bQdXfPu+gRTx9gPrkDUDLdVDvqcLx
bxPhPVp+9EIuK3rCjTALBglghkgBZQMEAxIDggzuAKKWJqZR+Ce+zJlGjCzLJWmx
/c3mxMug0vo1NtheVQ4Id3Lnhv5yimxnGXCdtkCcRyQGef6ku2gpf6AeSTBaA9sa
C3d8sB1HLTlPnwBXTWJ0xgp0kGVqCfwrbeRdCsoFrRoz1V2ETBecFehgQFHYLWtO
Q/VZrXPpwUxjIJRpSdxIw3SCsADjT/cQI9CQl5riMFOsZedRRPxSpw6lAGo557Ul
IgsxXA90mZQvRONY3mh8YkDT5uxQV0xRlyLLcZ3dxkzKgZrndz9F1yPrDGA0ixdk

F+3u4mxt8BJYZyRA2UeLEPG0OAIORmCeNuuixnFytGkctTd83GHeAszUtSanOyIU
JSomB/9G/omW3TU8uI8zWFuW+Fac8kpcNIECMujWUpB4RlxTWC/eTqrSiH7leE41
aCaZC8Hd1jwTEK2xvPC8KCI/apk71rnmX4KFOCgG8FirIw/Usv6mnqt2G65nNlde
7gWS8XoMFJDtYWDUpkgwRwLHvBNvKK/WikRDG3pSI8E1UyU1MU5R1JQnQRgzjazR
5lWhsMLrYoC+wi+XKNkFjeKVbgoiVoAD3ZKGnkJvIEueilGEWTYoTiobKBHzDjI3
b9x0GuBU0Z/MiRsycsDJUrl8hK8kpgBAS7ENTaowGdE9KhiqzY4xX83vgjy0bDL4
X9+2CXuQtFR5QotYapjA1HJUM5UuB5ix7i32dbZfnrJOZqcgocWuFRkShsAAu/wH
wFeeiJodlAlFVh3onDgh76316MewB/VLpM3G4EctzI9i5xqJzKXjlKWXP9ZjW/Sd
50R42fe6AaUqN4DaNbwOQaBs48auupU+EK6NytQl9rCeCJ4gicnM1no4QQbcwecZ
VvrrCs8egd3hawzMuCRhKeNzaYVeqnWhsRWdLxJANa07ANQqgXLhSx7vDBLiIQLA
UNqNYndYjEG3jirFxITUtHcz4ekZC7hpGkbl5qabQvrlk6lJ7yfuNuoO5jcCKY/C
BbOluVgjOeYeBfqJ9u9MwMcTUs26DkpW9h5YXIXATBj9dI3Tzp6fb9qtg9zBTHap
JKwmbUbgbPJSabtFMwh5XUpIpsJ+Mujek4uQ2GNaA6YfitHeDP91DM9ZUVFM7/5/
0BysQZLfD3SDO2Q/TnCug4XlCy0Z8ChnCBRGldqU1cMKPszXviLauBoM6aiLx7Lf
feuMxZVWJ1xC3crrP4OXpEO7IkIsDeKoVBBdkhV9RuD394h7jSCljMEVCYKCRUWV
mvffcmJNxfIfTgeLboHiAu0vDjE6s9vrcUASTDTgXuooy3rQ4q4fkWvV+cLPCZH9
sRZQvGcziMWrubTPa3WuVG9FHmejYcTsRTIxl1IlU8IjgW5hzwdmiZKI9cnCPqxy
a+uSk7K/by9rG4msxklsijdQ8CRg3V4mgNW5FTa6hXpfabmYWEZetYzJ7CISueCH
RtevtYz5HTARC5KB2CcYMV3Cz1AaIxd29lZpWJFd8sWLN50P/1KBanqeFfqXgIuW
RVE26I1IyMrrxzPUN6F+C7AWEEFGI5rue3M2EdIAAP3Sv6paLut9cESLS8zQBz8J
nL+RXsvL3ctYW5XbcuDIFJaIAENJArveEI4u4Oyvxyz0bp5JmS8tpRTWUoRWBt9B
hAp4axTi2eUWn6XHCBTG5DIyi4i+Tz8yt+zkBLdCh/b/UgxVtH6O5jcrOc9j0kb7
IJGacW71sE/sM0xCLFvqo2pEaKrt2dF+3/sK9TsT8TCisrEC4iCeSWt/I6Jb1RIc
TjBIt+yP/0u+B/pZVs9sUQpNkgkUmxAgJsuhH9PkIV+26vijDppKSgod4CrlCrkE
gvUe55DhUTx7bi2TNNUwQNRY8KeGsJMEOap6TIV0MdEnvRqpnVpWlYWSIpC4lJFH
uEPkkfumLKphH+ZrComFg/8Ys0ApNiGpAIIDngUkOvUggT51vyk5EnzJs7n6XQoI
cnJ+nax/+bLC9UHo6llqmggUXOUnRupclyJUo99uafwyoCw/8y43OI4s7CffQZ9U
JJLOT5WvoFXLqZlLl99CmQRZ/+0CEObOxsf5zmt2YbKKLVh1xZCyOyglODG5fU1O
a+droWi/K/Y8EN2TFYC9278AC9TVFhY7QoMhn1eG3Lc5cNfL5G5aPcbIybJUJ1XT
17wHKrExmRG8/itNHB7Yh4uetwumgls+vWL/QpCepU3kTMc4yXMnuYkmqmA4vW/B
vy4CCSarPjUCoF7hyfPSEv2fI1iTleJl5lDNsII8uz2zW2kNGK9RVKNPkMAW5EqR
nVGdt6TzpNlRtBo8ZZbV6g00+ZRKrW0iXfYRv6jT0+02/BjCA4sr+KwwTS70oU0s
MIsgfLFRWUP2swV3oUsGrxL+r8OckzqTmRzMMk0MCiIm6Tg2DSBhiMF9Sn8cajIW
PVpBZoPlmPz3S0LqMH4KFRHqx8HIcRUKdRSItOn7h7kZEl4dkrY6HehbZd/s0h4Z
EIUotKN/Ls8NbS+NqxvKkiu/udt+tGjWzlauHR1I8b4uyJMoE1A+pJS5A5AdXPsr
q0Nj3cHhvZCSdbsFH4pWd7m90EWc4e54EX6NHnFUlvypjmQ5rXAIdNn3uMQ9hD9F
BQjDa9p5KI8bTvQBojS1fuOKR2wTqjCFVAjnnTgQ1xvDtIdBDPpfFwB/fIvh2wRi
tnu/9bs4suGX7yXWIAg+KopJUL5AjF8QUyOPzDVAVHQk+rRjg2rcb4b72dR56o8F
Y8GvdTAJGKApxFFKXk+YRYXUBczoaV6XG3J4+pieHv10P/wNrua2nKotTNgQ3X3e
Lg8BYMejeUAFyaLSn/taMsCwDAQfEIExx8wCB3ySp7oa0l70V2IjXReTijM8M2Yf
osYQ6KX01GyP01BGPiLj4Z2CeAr7y7QH5qkWBIQNOPEOMpC/3lKfWgF0GUrW1qzD
lxRSRcqYrmvWpT3tZYftGt1WqpFHi6N4R09KaLrEEgF57CYfKrQJisBKehm63v5A
2Pbc9mRsbCz1KSP0dtKSaXhaMBSbs/AflAOEniA45ZozFQHIp2VIgvYVxe0kV+i4
0UOGBb4eckx8jfXr9zG1MyLCmc3tBEDZIIicJSaqGA4lRpKQoGbHOxGEEgvWi46e
pRUpsCh7xP9k4bwrAGiFUOsq+NneS0DL5/iosSwjVlrF1CKjpG6zvTNgtSgQ0WL0
5/1f/gH9KJDyzVukKsl8xfK/EQw7q9zNxlw3DdqX0Gh74+AwLpUCMIen4+XgKxTD
9qLphL/gsWtncpaK2buRmQii2++Px8+8XbrReAENjLfuEH4m1Pnqo/tVtGpt2JOv
KrtM7Abj9ZRcxVthhy+K6cZwC9qcFSeuswcU81bqpJ4pmEcgCbd+DnqMBBMJsJHs

```
   wKrhY4VaOpKt15lWQzqaHV2+S9uVmhS48aIUw4GWhfuwWLVwTACEtUvqPR8eK1n/
   MLP0tmLlc2igdIQ0+Rj+27GJOm4sWx58kEnOlO1E5AlaMmoclJuVtwn7LSjdLY6/
   HVBYST10of4OwXTk6reKpZnfzwI4Jb1Ml6MKyujGrBHfFpF0ezY8gGnhUif+GeCa
   A5qySvwYq7dHpbhBsWFZXf5coK5xCUX7MblWY1kzDXqKr0dTiSHL0AxKD5DORpal
   poSnF1auJcqwyoPDvtJxG/k2pWZD6206v6o/Ed2VItKgzeRQBznFCLhAbKSZKSjQ
   VDT8HpX1xoakWvXrmE5g0/2UVQu39MuHkoXV3MwG45N/03fyUsR+MAAvyjfOqlmb
   8UFQYGdAeA/P5+60DRCtC/v/9jlWj+uDDbyUD5rYlO9kLKlxmKgdH2LHerJ63Y4a
   0P4PvBNYiPevrvN9WO5f0Z5N/fMwz88gm8Q2TVt3kJqHPj9SFI1e/KVOCZQlNTLh
   ewnrNZZAjP9yQGbxnIgeiftnSmBM/30VxCgDv1+hE1FeG2jm4URgsiIu53rX6Ac7
   HUl48/QWIXHMUQxSWXeXxGKI+SaVFRrM+5DCjWxW4bM190Joj21R7dBRfCbmdLrI
   y6uhLJqKtZ+imkZnGpd33lWBJp9wuOVSEJWiS2Jwj4sgdrywD1mSPkyG2oXZOLUL
   A3e8SFVGQaI7aFwG7wiRibgnuLz1omU/n46f02Gjj/0kSkqlhWIT7rqKkRYimMhE
   6PTVRRLDmFK6OhK2WSCDIXGkQxv/pD5TOL0bjchj/FF9sQhDhG7W8IPOLMrtRsmK
   PHfQ4l6+SUMo+tVNfo87m6uQismTwSu6nt4X+CNFvb5bY9Apltg5HAkTInJ41ed+
   Shu2siOaLEbfQ3rjeeyTwuCnn8NPOe3VBLfhEvznBR+wCQA7W6/UiUoJCZgZwQkP
   pGbnhaBP+YJF9ypE/mS0RGhti6UGKS9QWXaH9h4hVlt3krr4IiVYgpGetb/vCA5d
   fYAjMD9K/AAAAAAAAAAAAAAAAAAAAAUNFR4jKA==
   -----END CERTIFICATE-----

   0 6408: SEQUENCE {
   4 3077:  SEQUENCE {
   8    3:   [0] {
  10    1:    INTEGER 2
        :     }
  13   20:   INTEGER 15 67 7A 84 2C 46 84 33 4B F9 2D 4E 2F 75 18 EF 0F A9 B1 B4
  35   11:   SEQUENCE {
  37    9:    OBJECT IDENTIFIER '2 16 840 1 101 3 4 3 18'
        :     }
  48  140:   SEQUENCE {
  51   11:    SET {
  53    9:     SEQUENCE {
  55    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
  60    2:      PrintableString 'XX'
        :       }
        :      }
  64   53:    SET {
  66   51:     SEQUENCE {
  68    3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
  73   44:      UTF8String
        :       'Royal Institute of Public Key Infrastructure'
        :       }
        :      }
 119   43:    SET {
 121   41:     SEQUENCE {
 123    3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
 128   34:      UTF8String 'Post-Heffalump Research Department'
        :       }
        :      }
```

```
 164   25:    SET {
 166   23:     SEQUENCE {
 168    3:      OBJECT IDENTIFIER commonName (2 5 4 3)
 173   16:      UTF8String 'ML-DSA Root - G1'
         :       }
         :      }
         :     }
 191   30:   SEQUENCE {
 193   13:    UTCTime 17/10/2024 23:37:23 GMT
 208   13:    UTCTime 15/10/2034 23:37:23 GMT
         :     }
 223   47:   SEQUENCE {
 225   11:    SET {
 227    9:     SEQUENCE {
 229    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
 234    2:      PrintableString 'XX'
         :       }
         :      }
 238   15:    SET {
 240   13:     SEQUENCE {
 242    3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
 247    6:      UTF8String 'Hanako'
         :       }
         :      }
 255   15:    SET {
 257   13:     SEQUENCE {
 259    3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
 264    6:      UTF8String 'Yamada'
         :       }
         :      }
         :     }
 272 1970:   SEQUENCE {
 276   11:    SEQUENCE {
 278    9:     OBJECT IDENTIFIER '2 16 840 1 101 3 4 3 18'
         :      }
 289 1953:    BIT STRING
         :     FD AE A2 1D 3C C2 7D A9 EF 68 78 BC 19 4F 94 FA
         :     85 F9 9C 39 2F 91 28 FF 73 F7 86 9C 6B 74 7F BD
         :     18 0E 3B D7 A0 D9 B0 5D 2C 48 A4 51 F3 EE 62 E7
         :     98 9D 3D 94 42 36 3B 53 86 2E 05 05 8C 06 1D B9
         :     B4 11 32 5B E3 8F 43 B3 66 30 F8 80 98 6C FD CE
         :     86 F0 1D A5 A0 AB 5D E9 A3 96 B7 82 8F 3D 5A 66
         :     9A F5 DA 38 5C 05 79 4D CD 56 AF 89 61 98 7C 52
         :     4E DD 09 0E 8F 34 2E 85 A8 88 AB 13 EE 57 F0 4A
         :         [ Another 1824 bytes skipped ]
         :     }
2246  835:   [3] {
2250  831:    SEQUENCE {
```

```
2254   15:     SEQUENCE {
2256    3:      OBJECT IDENTIFIER basicConstraints (2 5 29 19)
2261    1:      BOOLEAN TRUE
2264    5:      OCTET STRING, encapsulates {
2266    3:       SEQUENCE {
2268    1:        BOOLEAN TRUE
         :         }
         :        }
         :       }
2271   14:     SEQUENCE {
2273    3:      OBJECT IDENTIFIER keyUsage (2 5 29 15)
2278    1:      BOOLEAN TRUE
2281    4:      OCTET STRING, encapsulates {
2283    2:       BIT STRING 1 unused bit
         :        '1100001'B
         :        }
         :       }
2287   29:     SEQUENCE {
2289    3:      OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
2294   22:      OCTET STRING, encapsulates {
2296   20:       OCTET STRING
         :        9B 07 B4 A4 75 C4 BC 91 5D 35 E0 C9 A1 C1 62 E2
         :        77 55 D6 3F
         :        }
         :       }
2318   31:     SEQUENCE {
2320    3:      OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
2325   24:      OCTET STRING, encapsulates {
2327   22:       SEQUENCE {
2329   20:        [0]
         :         9B 07 B4 A4 75 C4 BC 91 5D 35 E0 C9 A1 C1 62 E2
         :         77 55 D6 3F
         :         }
         :        }
         :       }
2351  730:     SEQUENCE {
2355   10:      OBJECT IDENTIFIER
         :       deltaCertificateDescriptor (2 16 840 1 114027 80 6 1)
2367  714:      OCTET STRING, encapsulates {
2371  710:       SEQUENCE {
2375   20:        INTEGER
         :         0C 24 0E E2 3E BC 25 E4 BA B6 08 12 BA 36 76 5B
         :         FF B9 44 C0
2397   12:        [0] {
2399   10:         SEQUENCE {
2401    8:          OBJECT IDENTIFIER
         :           ecdsaWithSHA512 (1 2 840 10045 4 3 4)
         :           }
```

```
        :          }
2411  142:        [1] {
2414  139:         SEQUENCE {
2417   11:          SET {
2419    9:           SEQUENCE {
2421    3:            OBJECT IDENTIFIER countryName (2 5 4 6)
2426    2:            PrintableString 'XX'
        :            }
        :           }
2430   53:          SET {
2432   51:           SEQUENCE {
2434    3:            OBJECT IDENTIFIER organizationName (2 5 4 10)
2439   44:            UTF8String
        :           'Royal Institute of Public Key Infrastructure'
        :            }
        :           }
2485   43:          SET {
2487   41:           SEQUENCE {
2489    3:            OBJECT IDENTIFIER
        :             organizationalUnitName (2 5 4 11)
2494   34:            UTF8String 'Post-Heffalump Research Department'
        :            }
        :           }
2530   24:          SET {
2532   22:           SEQUENCE {
2534    3:            OBJECT IDENTIFIER commonName (2 5 4 3)
2539   15:            UTF8String 'ECDSA Root - G1'
        :            }
        :           }
        :          }
        :         }
2556  142:        [3] {
2559  139:         SEQUENCE {
2562   11:          SET {
2564    9:           SEQUENCE {
2566    3:            OBJECT IDENTIFIER countryName (2 5 4 6)
2571    2:            PrintableString 'XX'
        :            }
        :           }
2575   53:          SET {
2577   51:           SEQUENCE {
2579    3:            OBJECT IDENTIFIER organizationName (2 5 4 10)
2584   44:            UTF8String
        :           'Royal Institute of Public Key Infrastructure'
        :            }
        :           }
2630   43:          SET {
2632   41:           SEQUENCE {
```

```
2634    3:            OBJECT IDENTIFIER
        :             organizationalUnitName (2 5 4 11)
2639   34:            UTF8String 'Post-Heffalump Research Department'
        :            }
        :           }
2675   24:          SET {
2677   22:           SEQUENCE {
2679    3:            OBJECT IDENTIFIER commonName (2 5 4 3)
2684   15:            UTF8String 'ECDSA Root - G1'
        :            }
        :           }
        :          }
        :         }
2701  155:        SEQUENCE {
2704   16:         SEQUENCE {
2706    7:          OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
2715    5:          OBJECT IDENTIFIER secp521r1 (1 3 132 0 35)
        :          }
2722  134:         BIT STRING
        :          04 01 00 56 06 A7 BF 43 84 35 27 27 E7 AE 85 B5
        :          88 A9 13 0F 0B F6 09 C8 C7 17 A7 9C 44 17 DE 9C
        :          42 38 B6 AC E7 24 BD CA 90 92 1A 84 2B 6B A9 5D
        :          C4 8B 69 C9 48 91 2D 7A 44 41 59 81 3D 2A DE F3
        :          B6 9B 5C 00 85 7C 5E 29 E3 EC 30 BD 84 88 23 10
        :          20 40 2B 5E C1 30 08 31 AE FA A7 71 98 DD A5 1D
        :          44 0D BA 28 AC AE 48 A0 45 D5 C8 2C 76 80 5D D7
        :          D3 B4 0B E9 A2 9B 00 72 19 E6 FA F5 1B 1C D6 AF
        :          73 15 51 8C AD
        :         }
2859   82:        [4] {
2861   80:         SEQUENCE {
2863   14:          SEQUENCE {
2865    3:           OBJECT IDENTIFIER keyUsage (2 5 29 15)
2870    1:           BOOLEAN TRUE
2873    4:           OCTET STRING, encapsulates {
2875    2:            BIT STRING 1 unused bit
        :             '1100000'B
        :            }
        :           }
2879   29:          SEQUENCE {
2881    3:           OBJECT IDENTIFIER
        :            subjectKeyIdentifier (2 5 29 14)
2886   22:           OCTET STRING, encapsulates {
2888   20:            OCTET STRING
        :            EB A3 D0 8B 51 FE 12 DC CC 21 66 87 85 0F 98 C7
        :            67 73 88 34
        :           }
        :          }
```

```
2910   31:          SEQUENCE {
2912    3:           OBJECT IDENTIFIER
         :            authorityKeyIdentifier (2 5 29 35)
2917   24:           OCTET STRING, encapsulates {
2919   22:            SEQUENCE {
2921   20:             [0]
         :             EB A3 D0 8B 51 FE 12 DC CC 21 66 87 85 0F 98 C7
         :             67 73 88 34
         :               }
         :              }
         :             }
         :            }
         :           }
2943  139:         BIT STRING, encapsulates {
2947  135:          SEQUENCE {
2950   65:           INTEGER
         :            49 E7 4B 12 3A 5F 93 4F 18 67 AD A4 15 24 12 1D
         :            C7 20 34 EB 5B 6E 1D 3E DC 7B E3 09 96 00 DC 2B
         :            7A 3B 1B 4B 44 AF 6E 2A 67 40 93 D0 5E 34 EF F1
         :            A5 C3 94 30 0C 88 F2 87 69 DA 39 A8 22 57 32 F5
         :            37
3017   66:           INTEGER
         :            00 98 36 D1 99 EB ED 68 D5 10 D2 4E 9E F7 B0 96
         :            73 CB 53 3C 4E E0 5E 8E 62 A8 50 F4 35 4F 6F DB
         :            41 D5 DF 3E EF A0 45 3C 7D 80 FA E4 0D 40 CB 75
         :            50 EF A9 C2 F1 6F 13 E1 3D 5A 7E F4 42 2E 2B 7A
         :            C2 8D
         :           }
         :          }
         :         }
         :        }
         :       }
         :      }
         :     }
         :    }
3085   11:  SEQUENCE {
3087    9:   OBJECT IDENTIFIER '2 16 840 1 101 3 4 3 18'
         :    }
3098 3310:  BIT STRING
         :   A2 96 26 A6 51 F8 27 BE CC 99 46 8C 2C CB 25 69
         :   B1 FD CD E6 C4 CB A0 D2 FA 35 36 D8 5E 55 0E 08
         :   77 72 E7 86 FE 72 8A 6C 67 19 70 9D B6 40 9C 47
         :   24 06 79 FE A4 BB 68 29 7F A0 1E 49 30 5A 03 DB
         :   1A 0B 77 7C B0 1D 47 2D 39 4F 9F 00 57 4D 62 74
         :   C6 0A 74 90 65 6A 09 FC 2B 6D E4 5D 0A CA 05 AD
         :   1A 33 D5 5D 84 4C 17 9C 15 E8 60 40 51 D8 2D 6B
         :   4E 43 F5 59 AD 73 E9 C1 4C 63 20 94 69 49 DC 48
         :       [ Another 3181 bytes skipped ]
```

```
        :   }
```

B.2. Algorithm migration example

B.2.1. ML-DSA-65 signing end-entity certificate

This is an end-entity signing certificate which certifies a ML-DSA-65 key.

```
-----BEGIN CERTIFICATE-----
MIIWJjCCCSOgAwIBAgIUQZG8jQpzWDji9fN14AOMsoG89SIwCwYJYIZIAWUDBAMS
MIGMMQswCQYDVQQGEwJYWDE1MDMGA1UECgwsUm95YWwgSW5zdGl0dXRlIG9mIFB1
YmxpYyBLZXkgSW5mcmFzdHJ1Y3R1cmUxKzApBgNVBAsMIlBvc3QtSGVmZmFsdW1w
IFJlc2VhcmNoIERlcGFydG1lbnQxGTAXBgNVBAMMEE1MLURTQSBSb290IC0gRzEw
HhcNMjQxMDE3MjMzNzIzWhcNMzQxMDE1MjMzNzIzWjAvMQswCQYDVQQGEwJYWDEP
MA0GA1UECgwGSGFuYWtvMQ8wDQYDVQQLDAZZYW1hZGEwggeyMAsGCWCGSAFlAwQD
EgOCB6EAh3C60Iowi3gHMtKvoDgZ1gHulpK4i8rX/+KOI9lKjMr4BUqYKeM80jQ9
odCo1B3pTpG+79xQVpZakl2VCdhDEw4cdp+JZ21lwVhO8EBwMVFPExk4F3Tz94+J
2y0XqVx4TSGbeJzaaqPVEsJV/+KjBGr1BUUMFGl4ZAKwe5+47EAK9TZWNgsgW0MI
x9G0q42s8WHh0LTymbi7zTcUEh8HcHDJPfxcH53AjXTlOdjlnzUcyzxobot+cah/
62AEiPUaAuPQsko9l1Lk5B6frofqTE7tpNBn3eMjXV2R4cPJo/Kj1wFrchdD5BMg
+MD19mDu9aT7BIMC3MZemWaynMRnbwfMkqDFpPKE1mISbDv2Ia0bqzHUNMUSc26X
aer2rcXdD7TM4kZfS1SL4QhnbD6chyfw9Z9ggPiSU2J4glEL1detpi2MwWwKtSiQ
5Sc0ksaX35U6ULTl+5zOwQd8WeXaS+7AUlTS25mDzQRXd+goSX2QZC/e2a2yh2Hs
CIDP8PAG/J6ClnuMasGwo+4v93/GKfGjJ85HqSeK/ybbQna/bi4dwabtSRPmflmk
1YDFov4IP9J9oI3NQx/JEMSgHWpSxGsdGOgNGHLTACbqLCqLhWCjiJyMU5bQo7Um
p9U1nzaZoIQkKbsTAE2FrTKA+PJtDxl3OkDzNnCKW2IijN68+msyuDTMdv4fLIBC
KCsFPHpLRZInBKwZJ8a2qm0G0a6R64/A/65KoVoByDkKsJbP0QtpZmaLW+r9G6yr
vOSBgorDYBcA1l6y6gh8SS/RhSgmBFtXog4l4pm2DwkiqGEqz9GOUgMMA08UocHW
TvfUlGJYM/INFqLc5xM3xJ+6ef37+dlNd86+HXVx1uzEOCDJHhlVwYKNhoKiGGeb
3IrOkzVxQ44tVmjfIn6RmlvFyAIKuJvRxPN53wFFI2j4AhyCaut8dcgFdKdoAfnQ
AuoE0i1vMcJXBn2XAlFmifCVYw6AtghqVkEVlOK0VwEcNeiCC3TzwqPLLNkgedGZ
fiVTcJSVsAN6Q1KjPhAWAaUSCMPYgHKVzSOu9eKRiesiqEjr44D7ATWi3VP3kzHH
lRmPakUAV9U7woOpNHbsIyEhrua9n/D5LoYbL2a0QLtcs2BW6pGfXLlIafW95QZL
rz1uUvSzb8MuLdFuk3Moh/ugHWvmcyoLloIGhf789/qdEldmSMe/PwM3ruHsoKMj
/yTW28wvnkJU6y8FoOaDhLP83MICvLc0jockJXvgG6Y5eEfQ+69Or20Gvneb+fmn
YDxYhIqLhcZ9EjDAFsgKO83Xk8lWa20AXUX6vbkpbimRCRQK8NLc+NHxVYLnKwmN
Y9av2LPuXxyCm15YEWOlZs+rXa3zj8fa2ptidCGHICHR3rKEiyPfCIMAFhs5VGva
3SZWFOjFVXzOl8q/CODaoJwPXVPe8hiEIfnswNKBsg6+mZcv0wD252QmrCAmGhIh
z5uBj23TL/wHoucrVfvDhjQWPiiXNRSVYf9abgxU/C2VfQFJSWH9SDTQIShzY2vq
7T/Nnb76+3w+oo+f7a40Z+B2vSJJ37wqQQmVnQxQHrDMgRGzeS2OHoQMcZrrwOwz
c9IeP+pa4qQcdrzIuAbKHcZfGWS49OMSdOLAizOwKxkOKmixmNndQBbXqOjH7ofl
IzsX+muGqhp+pX0ZjH5A+SgPc3IogUV7giY5ABzn4q2Z7yyMewTrQB2VRANdM3EZ
BLGK3fQ/OrSrPqSfulwCLc97vLuByBfTrpJOhnLzM9AL8U6NascJ0W9fcyLKtBj9
+iGKhq0MgRgK/IsA1J7SEczyfk2sf39jIbPzSvD8RrbVcqO2p5wkzpL2H/BuD2Wq
dw301hefg/eSdcO/cSHumLUU9ZyRfQuWAFsZQ5+OXkSaIMStJZOScluoQuY0NHib
5bwni5TX3G8TZ0tcmMFgrhmNmv+xUYNhhKYOEL/SAIbonl33KR6URa1R0eQMnhTs
Gff7wHfMX7ggynZ20886NWHLA30whGHPiLcn8QHJFdPvRaJvrF03rEGUPMKXkoV3
```

/vX0QoA10Y4VD6EwdDiIOrIiBDSYBB1ZteGn5kG16c4YZTbenYJ6QBjB7D6FrkG5
SjNzDlSL1GBe/rxbyuEYrfM6ImIrl+GPP4p/aFx/tAmPrgIj6gxeXrUMaRXOipht
HWHdWdhnHpq6I61t0KmcbFnMZ2AQICJxbVSCd0uVwvtdJA8xftakRiC5Miy5c3jb
OmcPsTiCk3cj3U+cuXFiuiNMkkIxpUineagyZVYhYnn2QlerFdJC2Lfw5DHekIA0
nt8byScNsZujwPvIuKeevA/0byH6QpizT0lUBW/ZioI3780b5VZn0BRBNG7cY94s
9EpjfbuaTDv2xW7AGVGgOUy8VOnX0Xf9GlDiXOUnXAUJZjIFNmd4D7fyGQccbXl4
V9xf1xlIL8JYlHSgpMuVaSMlgwQEbHQrPNzpdmXaNfmFo/63FLgs8CL5XvaY2Qyj
oWnGdF6BUL2BFVFw+7nNTEqIT30XfEDJsqcDQPNLFK9vph1M6Bc2mqMpw0d3lMvK
1BIKKD6cDVqb20HQdDl7UejWY7ZdzoH7n8FLvQGiMw3JFbwcNW+jYzBhMA8GA1Ud
EwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQWBBRJdGxRQiHln7uR
5KM1S+MBTZCQjjAfBgNVHSMEGDAWgBSbB7SkdcS8kV014MmhwWLid1XWPzALBglg
hkgBZQMEAxIDggzuAGCvh2/Uhmxi1UZdlJIESseE0B5d5l0j10+pP5DAzzK29T+C
dHCNTxH7u2QSzKVI7PlhuufvItGyi0P2v4dFgd1fvWWJ4Q8SSGEV6/Rz4KX8PV4x
Rs5e6/aa3993WnIiEffCjTmibP7EkiHkhGmyx2xm4p1119bjVym7U94Wk9iEpL87
RHRD93XTg/zYYU2IVFMVsOaSkAXRGAyZkjGtM/BrHfcvS5ahQ1Y8HvDCaIGsv8vw
haUJprsJInyZ1/yTjM6ExoWGDFylEiOXKKEM/l5Q67z+stvjO9LeVnJUw4rFCbLu
xS7RdQYFZ2Pusz0dqbmjYdu64Iv9bjbY7DqoABumgRnjru36dbsAXmrOj94QrIm2
c0MiZihVRCKIoHj0wUUllfbnNCtXB2Ax4vjJXDLIoMk9R+hFjlocpF0cdUkxCEmZ
1nTeDP6+4uNALuuesT2HViqJLEmFFrgl1Mh+SvrQ00Arcsifs31ScixcvLjRtunI
cE4sq9knko7TTgZKuYaju9h/+6xlLbn8SgCWGD1pAN1VjIMps9a4PEmgU8OxrjSA
0zbizFqOAtX08ovsqYuSCSCB0jocG30lmM5Tzl5zKUWkNhSuQiBv5KI9U1W1F8Hb
9uBr2pu1Q50FFnYVRWwka0w61l+ryOIV4jhvrGV74YxEz5K7V63HHBDI5MYwA2oF
PMi0syrftCTG+2HGi2ohX4N4xVQ3NqZqlEH8pEw//J8sEqchdG6fZvBFVEeH+hId
skDYeKH7kkSVQfZp7/dUFRFCve2ZJu1lOOfAfLAIkbkInJk6EwPzbm9cwBplL33L
BcDaqBnINZct7ptfpl9g6OuuL0qyjTqaO6jPBcFIBNGBy0eggm2D1h+3J89vu+TI
+n6PVhcHnBBT5pzt2M8cfPzr41+Yw6WNxltH/wu+66ooqGm7nozY07BTFuGEGbRL
bEmdkOK0IVMSTNb0RIf7E8TfkT38YsuMHuYEmbHHy9KgfMJ+6WmP3G1hMySrb5RL
pL51iI7dejSyx3ASDE351Hbm3rpZgdaeZNTSiiELDj0tAMVi//gTbNsBUGuCQTbg
5YfJ0Ee3CFCBjoahcCthVgrh+NeENb3NNneBUx4j8OYiJvJP3GQ5TnijtDmp8K78
lTQ3W9Z2gvZa8UVpJnMqJD6AbbHxVBeW2JrK4QmRUh4KNo61TbF6OjrSc/d+GRGT
hg2I4Ua4xwpdx1IBYVH4eGlQ642hvGrUEqWwYTPmQG1SPK+5XXHIGggARPNt3UpY
s67gfaYYX+KzCipI1oN9bvZeR4PFkrxyKdsZ4PzMzmlFdTSMibDKaYt9JA6YDinO
Rp9MMYbwGRbxtMeNSpjCdPFLRudfnn7WRK2WSo8g8SfsUiEN6TX2RQQBOyT0w6gG
ABGjiRV+ToRN50sYqz1YetXPUZooVgNxieT9XbF9d1WLFp1vUBq57YIestxYlYxZ
gah5dGDl2ki6+LFJFz/iYiuHV56v3gmnGozotxWrh1NK5RgWOB0iM5xSPciLcq2Y
xfm+fA/mXWtUaHl4Y1jk44qPDWUJCJIN6XLXE4WxXWdtjkswoYsGTGWfC0c5Lz0s
YIe5EDjUZMhDibULHofwcU3/rUks0X35HIDVzVZFtyvgkFAec1Tby+VyyD6ssrnj
gloKmFFN/RssM50//yaenfvkuIWSo3ZQBX/5BwdqOcPJvYWA/vxAIJyNg8IMNsyL
yL3sbIrpzQFcTg2Q4cBEEFnHmju6qFq4xG4PzJafeAt94Hp5LI1sdALTho0ijVbT
zIoK3KEkYvQw+7+q5YRp3rUD+4C7RxSUxamxyqwoq5ZqakAtXAgghUXBJiDnL8z6
eDQ8e0OIaCbz5W12M92Sp0sjyORgW++LnAdfTXVHpyvQyAOez1pcErhJUo78oXio
GlaWXaxjkvXf8YzER33+v97X3zSP9BpMd3l11X4xTrMhjTY1fMVxZMhYiJ+zcZuA
EMVc2ikWj2oFfcpC3v/neF0mdNPiPPa+Bmi67IypjA47M/VyLii/xV9MWgEmvI/2
KGtFjN1AmiXkxTCUxY+eagSZl09x8l8QQwHre1psUA4fXImFi6dLFWEL07GPnQk/
FupDUBngvwtyUVXqFQxWihviYC722JpaqemAdw91lxqxsFYkU2Dqj81jwaRrJbju
HI1uTboqZ+o9cshZSJUIyRhxC1Pm/LwxaPfgkK+i9SB2sYtglFzFAAITNGLslQ5l
N1ZDhcZhwaa4mKnOi0sNfWeBJBGhHU7ok2Vw4LjpiyvJSrQTIfdyAJByB8lD1nvG
cgudSs3W9p7cEFlKOyV8I+j7Z3khtG8mSim4ZkzQfSOiEq3NtCNxk6iGB1ii0Kmv

```
   Me+OmBY/hT0f68eI3hHNCv+9epB2dQ10yy84limjaZQjpC6Qb/m3UGn4h7d6lfzY
   qI0mdiuT3JYdxf5uZWMdXbImd+R6vm3wv++TH56XaYmrZGT8u3AUu2iRWTKlv9/f
   HouVvtAGRQkj6edX2VQ882P5bXUQTNoBfMWzlSDLkXomuHniOVnT+mhmSNR1K8cj
   V2vT5IUL859ShRQFxHD9+wd5WP8+jW65fcUyjtOAsfG/nqv/tzpwLUx06tGqx70S
   EtClL2Uib2QNEZs4vcEQ5oltj6bwSNHr9BROIHRL47zgBbMZK48EYVn4N0Iuj8TI
   YhmsulTxoxEliXx8jMyC3ASeaKUobhFUjGM4mZSb4TWqxKrXR1xCB074fOf0azcK
   0btbqj4XN4kCHT4smC5Qa82tv9+YNIiNx5E5Dk4uI1fRGrPL/iHRKg4JbFa03KUB
   7wldKL/OXObo2q5POHcs/Rc8FKplGiILYFRBg2zFCFHmZnWwTklvsSoUJg9/skAG
   pHvs3gw8G19WepBFiU29iwuN9931RErk+o46o7gS6EcxGXM7uqeoeGhk/TZVSKCg
   wLsMZCTjoMgHFjxUOcBFsDMbGxO//rYCEA9qwhDcTeoz2MgXIdoVcp3vxBLohqAN
   6MQPAdIC6oFhKKKDQhBGJs7UTxkkSEE1GHWeHv5anTgGzCg/G+rKAvKrcZCSx+YW
   KjnsJbDS1ONnmSu958RrYNAsqUWnBCn2uLY9aj/0VYkoN1Q77MDBff32y9eaQaKm
   +pWXwigaiRE+GOEwGvU0Wtcjwt7tYmuZIRNqemVdlkMqA4D9wIh/HWfbg8hlkq0m
   m1PIFgbawXHNlLz5o+D+Sxe67a+kVrlk39m5HGWml/r7QxqKuUWsaryQWkTucvcG
   fmonqfVUfw5iCtuQNOv5VAdGDCpO3SikIq08nhKSMcs13RxQIzVCrsKpY6F4Zlnv
   EVwG4SqWcVSicElcn3oIOgnMCp0C4tdHxV3To/1jrwz3p318rSa0J2bqa28quvuy
   k7Bda75WW1bOeURbshnMv3k1lcsp1H0xYdLsvr+h3EnHqhd1ekyAq70taXCUJbzz
   gAqMCiIgo7OB/aGI30PvEIcz1u5jX+6lOGhpy/6Ol54DnTHE/o/zQHpSPdQiSutn
   pZZeZRbz13ZSxGSh/PS6/Xzic7hGtI3yxi6XyEC+gJoUv2a6sMEoptMPs5LUhA1I
   S13HQYaibxu98DDsKVCUf1zLn1j5xaOfuNgois4RysxTfs3/hPEkVVGwymWq7zaU
   Gov8U1E/XjaNWq31dOLUS+2Qq0/NoEB94ryhWPYUl/6JrvW3FhtwUIAiNR1iDFw6
   lgAdhQBGDzRBC8kzKSorNWNDdXoXXCe7gIYBfHppvqq12kQ11zk/13x0EhpzJZTg
   8K2pENPwj00pnXvimQC6vwXecydCkvyYXk0PI2c8LjDpnDqypQJVkR3xY3r21XF0
   gFnh4yUC0C+zjrs5PWa/C/7TBfwQyuJruZcPEiijT8tdcP7lakap5dD1tmGz2p1F
   67ppqbTCip46CyHDCbv6Qbtnj7DCOMYGkRIeb0MYC5f/g/7GsMiv1hToxRLLq2K4
   7/+xUoPj0LVwMM9/jP0Dry2y1+9A6qXM59oAWxb4HyPKQSlp5k/j6iWscRX1Rpuh
   71LlCySCSvjJrufw4YHtLG6dCS/t1gNT0NzfMAeTV5DU0BcwBlp2tk7cAH1Ti5ya
   UXfFvn2oY88PYr26GDJkzKF2jaAlUxwsgDh+X31JFgHu/OFQvdRO/HwFOMiPkEn5
   qPpca4A70JxL2U1z4T1Mhed5mW2OFil/LiB7WUjjCnkZJ2cRBGoBfP8jwwOUYgz+
   yikFRBRLfBjEBkFqHa1kPgBsXt1YHF3Y5mviKsyewLPidq1CrToM4xuRNN1hMf3y
   9aaYDoZ9cFO3Bklrjzu9AzUTHb5FEs7fzBoz+8y/3I4cmfCNw0rWkV+D4xuxACNG
   VnehzuIFGDI1PlVcdJvNYWqgw/y1v/D3CzxJmqMAAAAAAAAAAAAAAAAAAAAAAAAA
   AAAAAAAIEhcbIA==
   -----END CERTIFICATE-----

   0 5670: SEQUENCE {
   4 2339:  SEQUENCE {
   8    3:   [0] {
  10    1:    INTEGER 2
        :     }
  13   20:   INTEGER 41 91 BC 8D 0A 73 58 38 E2 F5 F3 75 E0 03 8C B2 81 BC F5 22
  35   11:   SEQUENCE {
  37    9:    OBJECT IDENTIFIER '2 16 840 1 101 3 4 3 18'
        :     }
  48  140:   SEQUENCE {
  51   11:    SET {
  53    9:     SEQUENCE {
  55    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
```

```
  60    2:      PrintableString 'XX'
        :       }
        :      }
  64   53:    SET {
  66   51:     SEQUENCE {
  68    3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
  73   44:      UTF8String
        :       'Royal Institute of Public Key Infrastructure'
        :       }
        :      }
 119   43:    SET {
 121   41:     SEQUENCE {
 123    3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
 128   34:      UTF8String 'Post-Heffalump Research Department'
        :       }
        :      }
 164   25:    SET {
 166   23:     SEQUENCE {
 168    3:      OBJECT IDENTIFIER commonName (2 5 4 3)
 173   16:      UTF8String 'ML-DSA Root - G1'
        :       }
        :      }
        :     }
 191   30:   SEQUENCE {
 193   13:    UTCTime 17/10/2024 23:37:23 GMT
 208   13:    UTCTime 15/10/2034 23:37:23 GMT
        :     }
 223   47:   SEQUENCE {
 225   11:    SET {
 227    9:     SEQUENCE {
 229    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
 234    2:      PrintableString 'XX'
        :       }
        :      }
 238   15:    SET {
 240   13:     SEQUENCE {
 242    3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
 247    6:      UTF8String 'Hanako'
        :       }
        :      }
 255   15:    SET {
 257   13:     SEQUENCE {
 259    3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
 264    6:      UTF8String 'Yamada'
        :       }
        :      }
        :     }
 272 1970:   SEQUENCE {
```

```
 276   11:    SEQUENCE {
 278    9:     OBJECT IDENTIFIER '2 16 840 1 101 3 4 3 18'
        :      }
 289 1953:    BIT STRING
        :     87 70 BA D0 8A 30 8B 78 07 32 D2 AF A0 38 19 D6
        :     01 EE 96 92 B8 8B CA D7 FF E2 8E 23 D9 4A 8C CA
        :     F8 05 4A 98 29 E3 3C D2 34 3D A1 D0 A8 D4 1D E9
        :     4E 91 BE EF DC 50 56 96 5A 92 5D 95 09 D8 43 13
        :     0E 1C 76 9F 89 67 6D 65 C1 58 4E F0 40 70 31 51
        :     4F 13 19 38 17 74 F3 F7 8F 89 DB 2D 17 A9 5C 78
        :     4D 21 9B 78 9C DA 6A A3 D5 12 C2 55 FF E2 A3 04
        :     6A F5 05 45 0C 14 69 78 64 02 B0 7B 9F B8 EC 40
        :         [ Another 1824 bytes skipped ]
        :     }
2246   99:   [3] {
2248   97:    SEQUENCE {
2250   15:     SEQUENCE {
2252    3:      OBJECT IDENTIFIER basicConstraints (2 5 29 19)
2257    1:      BOOLEAN TRUE
2260    5:      OCTET STRING, encapsulates {
2262    3:       SEQUENCE {
2264    1:        BOOLEAN TRUE
        :         }
        :        }
        :       }
2267   14:     SEQUENCE {
2269    3:      OBJECT IDENTIFIER keyUsage (2 5 29 15)
2274    1:      BOOLEAN TRUE
2277    4:      OCTET STRING, encapsulates {
2279    2:       BIT STRING 1 unused bit
        :        '1100001'B
        :        }
        :       }
2283   29:     SEQUENCE {
2285    3:      OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
2290   22:      OCTET STRING, encapsulates {
2292   20:       OCTET STRING
        :        49 74 6C 51 42 21 E5 9F BB 91 E4 A3 35 4B E3 01
        :        4D 90 90 8E
        :        }
        :       }
2314   31:     SEQUENCE {
2316    3:      OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
2321   24:      OCTET STRING, encapsulates {
2323   22:       SEQUENCE {
2325   20:        [0]
        :         9B 07 B4 A4 75 C4 BC 91 5D 35 E0 C9 A1 C1 62 E2
        :         77 55 D6 3F
```

```
         :         }
         :        }
         :       }
         :      }
         :     }
         :    }
2347   11:  SEQUENCE {
2349    9:   OBJECT IDENTIFIER '2 16 840 1 101 3 4 3 18'
         :    }
2360 3310:  BIT STRING
         :   60 AF 87 6F D4 86 6C 62 D5 46 5D 94 92 04 4A C7
         :   84 D0 1E 5D E6 5D 23 D7 4F A9 3F 90 C0 CF 32 B6
         :   F5 3F 82 74 70 8D 4F 11 FB BB 64 12 CC A5 48 EC
         :   F9 61 BA E7 EF 22 D1 B2 8B 43 F6 BF 87 45 81 DD
         :   5F BD 65 89 E1 0F 12 48 61 15 EB F4 73 E0 A5 FC
         :   3D 5E 31 46 CE 5E EB F6 9A DF DF 77 5A 72 22 11
         :   F7 C2 8D 39 A2 6C FE C4 92 21 E4 84 69 B2 C7 6C
         :   66 E2 9D 75 D7 D6 E3 57 29 BB 53 DE 16 93 D8 84
         :       [ Another 3181 bytes skipped ]
         :   }
```

B.2.2. EC signing end-entity certificate with encoded Delta Certificate

This is an end-entity signing certificate which certifies an EC key. It contains a Delta Certificate Descriptor extension which includes sufficient information to recreate the ML-DSA-65 signing end-entity certificate.

```
-----BEGIN CERTIFICATE-----
MIIYhDCCF+agAwIBAgIUQFy9NSVq9ZXG6QZyo14DJ/bew58wCgYIKoZIzj0EAwQw
gYsxCzAJBgNVBAYTAlhYMTUwMwYDVQQKDCxSb3lhbCBJbnN0aXR1dGUgb2YgUHVi
bGljIEtleSBJbmZyYXN0cnVjdHVyZTErMCkGA1UECwwiUG9zdC1IZWZmYWx1bXAg
UmVzZWFyY2ggRGVwYXJ0bWVudDEYMBYGA1UEAwwPRUNEU0EgUm9vdCAtIEcxMB4X
DTI0MTAxNzIzMzcyM1oXDTM0MTAxNTIzMzcyM1owLzELMAkGA1UEBhMCWFgxDzAN
BgNVBAoMBkhhbmFrbzEPMA0GA1UECwwGWWFtYWRhMIGbMBAGByqGSM49AgEGBSuB
BAAjA4GGAAQAFfoXF6AZPOkYTpb8vA2q+ZAtkE399B9BBz+q0A91vSeBvZbfat5V
hqVLtT+nEguQhlYhXf6CmCvFUERmQc8zfW4BaH1ZSd+kpuR5fJj6ibDbstHU3le4
Vq2qHR+aXvmccEtYVZ5BX3KE+gY/ezpY/BBXrd8vJuV72SPdsrNzjCz5z8OjghY+
MIIWOjAMBgNVHRMBAf8EAjAAMA4GA1UdDwEB/wQEAwIHgDAdBgNVHQ4EFgQUB4Ts
5OVjMVy4x3jV/GEY8FPDjK0wHwYDVR0jBBgwFoAU66PQi1H+EtzMIWaHhQ+Yx2dz
iDQwghXYBgpghkgBhvprUAYBBIIVyDCCFcQCFEGRvI0Kc1g44vXzdeADjLKBvPUi
oA0wCwYJYIZIAWUDBAMSoYGPMIGMMQswCQYDVQQGEwJYWDE1MDMGA1UECgwsUm95
YWwgSW5zdGl0dXRlIG9mIFB1YmxpYyBLZXkgSW5mcmFzdHJ1Y3R1cmUxKzApBgNV
BAsMIlBvc3QtSGVmZmFsdW1wIFJlc2VhcmNoIERlcGFydG1lbnQxGTAXBgNVBAMM
EE1MLURTQSBSb290IC0gRzEwggeyMAsGCWCGSAFlAwQDEgOCB6EAh3C60Iowi3gH
MtKvoDgZ1gHulpK4i8rX/+KOI9lKjMr4BUqYKeM80jQ9odCo1B3pTpG+79xQVpZa
kl2VCdhDEw4cdp+JZ21lwVhO8EBwMVFPExk4F3Tz94+J2y0XqVx4TSGbeJzaaqPV
EsJV/+KjBGr1BUUMFGl4ZAKwe5+47EAK9TZWNgsgW0MIx9G0q42s8WHh0LTymbi7
```

zTcUEh8HcHDJPfxcH53AjXTlOdjlnzUcyzxobot+cah/62AEiPUaAuPQsko9l1Lk
5B6frofqTE7tpNBn3eMjXV2R4cPJo/Kj1wFrchdD5BMg+MD19mDu9aT7BIMC3MZe
mWaynMRnbwfMkqDFpPKE1mISbDv2Ia0bqzHUNMUSc26Xaer2rcXdD7TM4kZfS1SL
4QhnbD6chyfw9Z9ggPiSU2J4glEL1detpi2MwWwKtSiQ5Sc0ksaX35U6ULTl+5zO
wQd8WeXaS+7AUlTS25mDzQRXd+goSX2QZC/e2a2yh2HsCIDP8PAG/J6ClnuMasGw
o+4v93/GKfGjJ85HqSeK/ybbQna/bi4dwabtSRPmflmk1YDFov4IP9J9oI3NQx/J
EMSgHWpSxGsdGOgNGHLTACbqLCqLhWCjiJyMU5bQo7Ump9U1nzaZoIQkKbsTAE2F
rTKA+PJtDxl3OkDzNnCKW2IijN68+msyuDTMdv4fLIBCKCsFPHpLRZInBKwZJ8a2
qm0G0a6R64/A/65KoVoByDkKsJbP0QtpZmaLW+r9G6yrvOSBgorDYBcA1l6y6gh8
SS/RhSgmBFtXog4l4pm2DwkiqGEqz9GOUgMMA08UocHWTvfUlGJYM/INFqLc5xM3
xJ+6ef37+dlNd86+HXVx1uzEOCDJHhlVwYKNhoKiGGeb3IrOkzVxQ44tVmjfIn6R
mlvFyAIKuJvRxPN53wFFI2j4AhyCaut8dcgFdKdoAfnQAuoE0i1vMcJXBn2XAlFm
ifCVYw6AtghqVkEVlOK0VwEcNeiCC3TzwqPLLNkgedGZfiVTcJSVsAN6Q1KjPhAW
AaUSCMPYgHKVzSOu9eKRiesiqEjr44D7ATWi3VP3kzHHlRmPakUAV9U7woOpNHbs
IyEhrua9n/D5LoYbL2a0QLtcs2BW6pGfXLlIafW95QZLrz1uUvSzb8MuLdFuk3Mo
h/ugHWvmcyoLloIGhf789/qdEldmSMe/PwM3ruHsoKMj/yTW28wvnkJU6y8FoOaD
hLP83MICvLc0jockJXvgG6Y5eEfQ+69Or20Gvneb+fmnYDxYhIqLhcZ9EjDAFsgK
O83Xk8lWa20AXUX6vbkpbimRCRQK8NLc+NHxVYLnKwmNY9av2LPuXxyCm15YEWOl
Zs+rXa3zj8fa2ptidCGHICHR3rKEiyPfCIMAFhs5VGva3SZWFOjFVXzOl8q/CODa
oJwPXVPe8hiEIfnswNKBsg6+mZcv0wD252QmrCAmGhIhz5uBj23TL/wHoucrVfvD
hjQWPiiXNRSVYf9abgxU/C2VfQFJSWH9SDTQIShzY2vq7T/Nnb76+3w+oo+f7a40
Z+B2vSJJ37wqQQmVnQxQHrDMgRGzeS2OHoQMcZrrwOwzc9IeP+pa4qQcdrzIuAbK
HcZfGWS49OMSdOLAizOwKxkOKmixmNndQBbXqOjH7oflIzsX+muGqhp+pX0ZjH5A
+SgPc3IogUV7giY5ABzn4q2Z7yyMewTrQB2VRANdM3EZBLGK3fQ/OrSrPqSfulwC
Lc97vLuByBfTrpJOhnLzM9AL8U6NascJ0W9fcyLKtBj9+iGKhq0MgRgK/IsA1J7S
Eczyfk2sf39jIbPzSvD8RrbVcqO2p5wkzpL2H/BuD2Wqdw301hefg/eSdcO/cSHu
mLUU9ZyRfQuWAFsZQ5+OXkSaIMStJZOScluoQuY0NHib5bwni5TX3G8TZ0tcmMFg
rhmNmv+xUYNhhKYOEL/SAIbonl33KR6URa1R0eQMnhTsGff7wHfMX7ggynZ20886
NWHLA30whGHPiLcn8QHJFdPvRaJvrF03rEGUPMKXkoV3/vX0QoA10Y4VD6EwdDiI
OrIiBDSYBB1ZteGn5kG16c4YZTbenYJ6QBjB7D6FrkG5SjNzDlSL1GBe/rxbyuEY
rfM6ImIrl+GPP4p/aFx/tAmPrgIj6gxeXrUMaRXOiphtHWHdWdhnHpq6I61t0Kmc
bFnMZ2AQICJxbVSCd0uVwvtdJA8xftakRiC5Miy5c3jbOmcPsTiCk3cj3U+cuXFi
uiNMkkIxpUineagyZVYhYnn2QlerFdJC2Lfw5DHekIA0nt8byScNsZujwPvIuKee
vA/0byH6QpizT0lUBW/ZioI3780b5VZn0BRBNG7cY94s9EpjfbuaTDv2xW7AGVGg
OUy8VOnX0Xf9GlDiXOUnXAUJZjIFNmd4D7fyGQccbXl4V9xf1xlIL8JYlHSgpMuV
aSMlgwQEbHQrPNzpdmXaNfmFo/63FLgs8CL5XvaY2QyjoWnGdF6BUL2BFVFw+7nN
TEqIT30XfEDJsqcDQPNLFK9vph1M6Bc2mqMpw0d3lMvK1BIKKD6cDVqb20HQdDl7
UejWY7ZdzoH7n8FLvQGiMw3JFbwcNW+kYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYD
VR0PAQH/BAQDAgGGMB0GA1UdDgQWBBRJdGxRQiHln7uR5KM1S+MBTZCQjjAfBgNV
HSMEGDAWgBSbB7SkdcS8kV014MmhwWLid1XWPwOCDO4AYK+Hb9SGbGLVRl2UkgRK
x4TQHl3mXSPXT6k/kMDPMrb1P4J0cI1PEfu7ZBLMpUjs+WG65+8i0bKLQ/a/h0WB
3V+9ZYnhDxJIYRXr9HPgpfw9XjFGzl7r9prf33daciIR98KNOaJs/sSSIeSEabLH
bGbinXXX1uNXKbtT3haT2ISkvztEdEP3ddOD/NhhTYhUUxWw5pKQBdEYDJmSMa0z
8Gsd9y9LlqFDVjwe8MJogay/y/CFpQmmuwkifJnX/JOMzoTGhYYMXKUSI5cooQz+
XlDrvP6y2+M70t5WclTDisUJsu7FLtF1BgVnY+6zPR2puaNh27rgi/1uNtjsOqgA
G6aBGeOu7fp1uwBeas6P3hCsibZzQyJmKFVEIoigePTBRSWV9uc0K1cHYDHi+Mlc
MsigyT1H6EWOWhykXRx1STEISZnWdN4M/r7i40Au656xPYdWKoksSYUWuCXUyH5K
+tDTQCtyyJ+zfVJyLFy8uNG26chwTiyr2SeSjtNOBkq5hqO72H/7rGUtufxKAJYY

PWkA3VWMgymz1rg8SaBTw7GuNIDTNuLMWo4C1fTyi+ypi5IJIIHSOhwbfSWYzlPO
XnMpRaQ2FK5CIG/koj1TVbUXwdv24Gvam7VDnQUWdhVFbCRrTDrWX6vI4hXiOG+s
ZXvhjETPkrtXrcccEMjkxjADagU8yLSzKt+0JMb7YcaLaiFfg3jFVDc2pmqUQfyk
TD/8nywSpyF0bp9m8EVUR4f6Eh2yQNh4ofuSRJVB9mnv91QVEUK97Zkm7WU458B8
sAiRuQicmToTA/Nub1zAGmUvfcsFwNqoGcg1ly3um1+mX2Do664vSrKNOpo7qM8F
wUgE0YHLR6CCbYPWH7cnz2+75Mj6fo9WFwecEFPmnO3Yzxx8/OvjX5jDpY3GW0f/
C77rqiioabuejNjTsFMW4YQZtEtsSZ2Q4rQhUxJM1vREh/sTxN+RPfxiy4we5gSZ
scfL0qB8wn7paY/cbWEzJKtvlEukvnWIjt16NLLHcBIMTfnUdubeulmB1p5k1NKK
IQsOPS0AxWL/+BNs2wFQa4JBNuDlh8nQR7cIUIGOhqFwK2FWCuH414Q1vc02d4FT
HiPw5iIm8k/cZDlOeKO0OanwrvyVNDdb1naC9lrxRWkmcyokPoBtsfFUF5bYmsrh
CZFSHgo2jrVNsXo6OtJz934ZEZOGDYjhRrjHCl3HUgFhUfh4aVDrjaG8atQSpbBh
M+ZAbVI8r7ldccgaCABE823dSlizruB9phhf4rMKKkjWg31u9l5Hg8WSvHIp2xng
/MzOaUV1NIyJsMppi30kDpgOKc5Gn0wxhvAZFvG0x41KmMJ08UtG51+eftZErZZK
jyDxJ+xSIQ3pNfZFBAE7JPTDqAYAEaOJFX5OhE3nSxirPVh61c9RmihWA3GJ5P1d
sX13VYsWnW9QGrntgh6y3FiVjFmBqHl0YOXaSLr4sUkXP+JiK4dXnq/eCacajOi3
FauHU0rlGBY4HSIznFI9yItyrZjF+b58D+Zda1RoeXhjWOTjio8NZQkIkg3pctcT
hbFdZ22OSzChiwZMZZ8LRzkvPSxgh7kQONRkyEOJtQseh/BxTf+tSSzRffkcgNXN
VkW3K+CQUB5zVNvL5XLIPqyyueOCWgqYUU39GywznT//Jp6d++S4hZKjdlAFf/kH
B2o5w8m9hYD+/EAgnI2Dwgw2zIvIvexsiunNAVxODZDhwEQQWceaO7qoWrjEbg/M
lp94C33genksjWx0AtOGjSKNVtPMigrcoSRi9DD7v6rlhGnetQP7gLtHFJTFqbHK
rCirlmpqQC1cCCCFRcEmIOcvzPp4NDx7Q4hoJvPlbXYz3ZKnSyPI5GBb74ucB19N
dUenK9DIA57PWlwSuElSjvyheKgaVpZdrGOS9d/xjMRHff6/3tffNI/0Gkx3eXXV
fjFOsyGNNjV8xXFkyFiIn7Nxm4AQxVzaKRaPagV9ykLe/+d4XSZ00+I89r4GaLrs
jKmMDjsz9XIuKL/FX0xaASa8j/Yoa0WM3UCaJeTFMJTFj55qBJmXT3HyXxBDAet7
WmxQDh9ciYWLp0sVYQvTsY+dCT8W6kNQGeC/C3JRVeoVDFaKG+JgLvbYmlqp6YB3
D3WXGrGwViRTYOqPzWPBpGsluO4cjW5Nuipn6j1yyFlIlQjJGHELU+b8vDFo9+CQ
r6L1IHaxi2CUXMUAAhM0YuyVDmU3VkOFxmHBpriYqc6LSw19Z4EkEaEdTuiTZXDg
uOmLK8lKtBMh93IAkHIHyUPWe8ZyC51Kzdb2ntwQWUo7JXwj6PtneSG0byZKKbhm
TNB9I6ISrc20I3GTqIYHWKLQqa8x746YFj+FPR/rx4jeEc0K/716kHZ1DXTLLziW
KaNplCOkLpBv+bdQafiHt3qV/NiojSZ2K5Pclh3F/m5lYx1dsiZ35Hq+bfC/75Mf
npdpiatkZPy7cBS7aJFZMqW/398ei5W+0AZFCSPp51fZVDzzY/ltdRBM2gF8xbOV
IMuReia4eeI5WdP6aGZI1HUrxyNXa9PkhQvzn1KFFAXEcP37B3lY/z6Nbrl9xTKO
04Cx8b+eq/+3OnAtTHTq0arHvRIS0KUvZSJvZA0Rmzi9wRDmiW2PpvBI0ev0FE4g
dEvjvOAFsxkrjwRhWfg3Qi6PxMhiGay6VPGjESWJfHyMzILcBJ5opShuEVSMYziZ
lJvhNarEqtdHXEIHTvh85/RrNwrRu1uqPhc3iQIdPiyYLlBrza2/35g0iI3HkTkO
Ti4jV9Eas8v+IdEqDglsVrTcpQHvCV0ov85c5ujark84dyz9FzwUqmUaIgtgVEGD
bMUIUeZmdbBOSW+xKhQmD3+yQAake+zeDDwbX1Z6kEWJTb2LC4333fVESuT6jjqj
uBLoRzEZczu6p6h4aGT9NlVIoKDAuwxkJOOgyAcWPFQ5wEWwMxsbE7/+tgIQD2rC
ENxN6jPYyBch2hVyne/EEuiGoA3oxA8B0gLqgWEoooNCEEYmztRPGSRIQTUYdZ4e
/lqdOAbMKD8b6soC8qtxkJLH5hYqOewlsNLU42eZK73nxGtg0CypRacEKfa4tj1q
P/RViSg3VDvswMF9/fbL15pBoqb6lZfCKBqJET4Y4TAa9TRa1yPC3u1ia5khE2p6
ZV2WQyoDgP3AiH8dZ9uDyGWSrSabU8gWBtrBcc2UvPmj4P5LF7rtr6RWuWTf2bkc
ZaaX+vtDGoq5RaxqvJBaRO5y9wZ+aiep9VR/DmIK25A06/lUB0YMKk7dKKQirTye
EpIxyzXdHFAjNUKuwqljoXhmWe8RXAbhKpZxVKJwSVyfegg6CcwKnQLi10fFXdOj
/WOvDPenfXytJrQnZuprbyq6+7KTsF1rvlZbVs55RFuyGcy/eTWVyynUfTFh0uy+
v6HcSceqF3V6TICrvS1pcJQlvPOACowKIiCjs4H9oYjfQ+8QhzPW7mNf7qU4aGnL
/o6XngOdMcT+j/NAelI91CJK62elll5lFvPXdlLEZKH89Lr9fOJzuEa0jfLGLpfI
QL6AmhS/ZrqwwSim0w+zktSEDUhLXcdBhqJvG73wMOwpUJR/XMufWPnFo5+42CiK

```
   zhHKzFN+zf+E8SRVUbDKZarvNpQai/xTUT9eNo1arfV04tRL7ZCrT82gQH3ivKFY
   9hSX/omu9bcWG3BQgCI1HWIMXDqWAB2FAEYPNEELyTMpKis1Y0N1ehdcJ7uAhgF8
   emm+qrXaRDXXOT/XfHQSGnMllODwrakQ0/CPTSmde+KZALq/Bd5zJ0KS/JheTQ8j
   ZzwuMOmcOrKlAlWRHfFjevbVcXSAWeHjJQLQL7OOuzk9Zr8L/tMF/BDK4mu5lw8S
   KKNPy11w/uVqRqnl0PW2YbPanUXrummptMKKnjoLIcMJu/pBu2ePsMI4xgaREh5v
   QxgLl/+D/sawyK/WFOjFEsurYrjv/7FSg+PQtXAwz3+M/QOvLbLX70Dqpczn2gBb
   FvgfI8pBKWnmT+PqJaxxFfVGm6HvUuULJIJK+Mmu5/Dhge0sbp0JL+3WA1PQ3N8w
   B5NXkNTQFzAGWna2TtwAfVOLnJpRd8W+fahjzw9ivboYMmTMoXaNoCVTHCyAOH5f
   fUkWAe784VC91E78fAU4yI+QSfmo+lxrgDvQnEvZTXPhPUyF53mZbY4WKX8uIHtZ
   SOMKeRknZxEEagF8/yPDA5RiDP7KKQVEFEt8GMQGQWodrWQ+AGxe3VgcXdjma+Iq
   zJ7As+J2rUKtOgzjG5E03WEx/fL1ppgOhn1wU7cGSWuPO70DNRMdvkUSzt/MGjP7
   zL/cjhyZ8I3DStaRX4PjG7EAI0ZWd6HO4gUYMjU+VVx0m81haqDD/LW/8PcLPEma
   owAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAgSFxsgMAoGCCqGSM49BAMEA4GLADCB
   hwJCAP1Y2r26fxhSYmL7pjEF7aP9V4ZzoVfpDf75VxKTW6vCvz/CozYhzn6mZka5
   18GBRgmXC4Ye88toLOhdxjT319/lAkEVyxpodYAljpbkwVjT4a7b4yioPJvR6S44
   6dU955tbns3PFbzhOU8usFhyXsKRDH7MBzt+ew9EnPEel7ud4+F23A==
   -----END CERTIFICATE-----

   0 6276: SEQUENCE {
   4 6118:  SEQUENCE {
   8    3:   [0] {
  10    1:    INTEGER 2
        :     }
  13   20:   INTEGER 40 5C BD 35 25 6A F5 95 C6 E9 06 72 A3 5E 03 27 F6 DE C3 9F
  35   10:   SEQUENCE {
  37    8:    OBJECT IDENTIFIER ecdsaWithSHA512 (1 2 840 10045 4 3 4)
        :     }
  47  139:   SEQUENCE {
  50   11:    SET {
  52    9:     SEQUENCE {
  54    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
  59    2:      PrintableString 'XX'
        :       }
        :      }
  63   53:    SET {
  65   51:     SEQUENCE {
  67    3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
  72   44:      UTF8String
        :       'Royal Institute of Public Key Infrastructure'
        :       }
        :      }
 118   43:    SET {
 120   41:     SEQUENCE {
 122    3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
 127   34:      UTF8String 'Post-Heffalump Research Department'
        :       }
        :      }
 163   24:    SET {
```

```
 165   22:     SEQUENCE {
 167    3:      OBJECT IDENTIFIER commonName (2 5 4 3)
 172   15:      UTF8String 'ECDSA Root - G1'
         :       }
         :      }
         :     }
 189   30:   SEQUENCE {
 191   13:    UTCTime 17/10/2024 23:37:23 GMT
 206   13:    UTCTime 15/10/2034 23:37:23 GMT
         :     }
 221   47:   SEQUENCE {
 223   11:    SET {
 225    9:     SEQUENCE {
 227    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
 232    2:      PrintableString 'XX'
         :       }
         :      }
 236   15:    SET {
 238   13:     SEQUENCE {
 240    3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
 245    6:      UTF8String 'Hanako'
         :       }
         :      }
 253   15:    SET {
 255   13:     SEQUENCE {
 257    3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
 262    6:      UTF8String 'Yamada'
         :       }
         :      }
         :     }
 270  155:   SEQUENCE {
 273   16:    SEQUENCE {
 275    7:     OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
 284    5:     OBJECT IDENTIFIER secp521r1 (1 3 132 0 35)
         :      }
 291  134:    BIT STRING
         :     04 00 15 FA 17 17 A0 19 3C E9 18 4E 96 FC BC 0D
         :     AA F9 90 2D 90 4D FD F4 1F 41 07 3F AA D0 0F 75
         :     BD 27 81 BD 96 DF 6A DE 55 86 A5 4B B5 3F A7 12
         :     0B 90 86 56 21 5D FE 82 98 2B C5 50 44 66 41 CF
         :     33 7D 6E 01 68 7D 59 49 DF A4 A6 E4 79 7C 98 FA
         :     89 B0 DB B2 D1 D4 DE 57 B8 56 AD AA 1D 1F 9A 5E
         :     F9 9C 70 4B 58 55 9E 41 5F 72 84 FA 06 3F 7B 3A
         :     58 FC 10 57 AD DF 2F 26 E5 7B D9 23 DD B2 B3 73
         :     8C 2C F9 CF C3
         :     }
 428 5694:   [3] {
 432 5690:    SEQUENCE {
```

```
 436   12:     SEQUENCE {
 438    3:      OBJECT IDENTIFIER basicConstraints (2 5 29 19)
 443    1:      BOOLEAN TRUE
 446    2:      OCTET STRING, encapsulates {
 448    0:       SEQUENCE {}
         :        }
         :       }
 450   14:     SEQUENCE {
 452    3:      OBJECT IDENTIFIER keyUsage (2 5 29 15)
 457    1:      BOOLEAN TRUE
 460    4:      OCTET STRING, encapsulates {
 462    2:       BIT STRING 7 unused bits
         :        '1'B (bit 0)
         :        }
         :       }
 466   29:     SEQUENCE {
 468    3:      OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
 473   22:      OCTET STRING, encapsulates {
 475   20:       OCTET STRING
         :        07 84 EC E4 E5 63 31 5C B8 C7 78 D5 FC 61 18 F0
         :        53 C3 8C AD
         :        }
         :       }
 497   31:     SEQUENCE {
 499    3:      OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
 504   24:      OCTET STRING, encapsulates {
 506   22:       SEQUENCE {
 508   20:        [0]
         :         EB A3 D0 8B 51 FE 12 DC CC 21 66 87 85 0F 98 C7
         :         67 73 88 34
         :         }
         :        }
         :       }
 530 5592:     SEQUENCE {
 534   10:      OBJECT IDENTIFIER
         :       deltaCertificateDescriptor (2 16 840 1 114027 80 6 1)
 546 5576:      OCTET STRING, encapsulates {
 550 5572:       SEQUENCE {
 554   20:        INTEGER
         :         41 91 BC 8D 0A 73 58 38 E2 F5 F3 75 E0 03 8C B2
         :         81 BC F5 22
 576   13:        [0] {
 578   11:         SEQUENCE {
 580    9:          OBJECT IDENTIFIER '2 16 840 1 101 3 4 3 18'
         :           }
         :         }
 591  143:        [1] {
 594  140:         SEQUENCE {
```

```
 597   11:          SET {
 599    9:           SEQUENCE {
 601    3:            OBJECT IDENTIFIER countryName (2 5 4 6)
 606    2:            PrintableString 'XX'
        :             }
        :            }
 610   53:          SET {
 612   51:           SEQUENCE {
 614    3:            OBJECT IDENTIFIER organizationName (2 5 4 10)
 619   44:            UTF8String
        :           'Royal Institute of Public Key Infrastructure'
        :             }
        :            }
 665   43:          SET {
 667   41:           SEQUENCE {
 669    3:            OBJECT IDENTIFIER
        :             organizationalUnitName (2 5 4 11)
 674   34:            UTF8String 'Post-Heffalump Research Department'
        :             }
        :            }
 710   25:          SET {
 712   23:           SEQUENCE {
 714    3:            OBJECT IDENTIFIER commonName (2 5 4 3)
 719   16:            UTF8String 'ML-DSA Root - G1'
        :             }
        :            }
        :           }
        :          }
 737 1970:        SEQUENCE {
 741   11:         SEQUENCE {
 743    9:          OBJECT IDENTIFIER '2 16 840 1 101 3 4 3 18'
        :           }
 754 1953:         BIT STRING
        :          87 70 BA D0 8A 30 8B 78 07 32 D2 AF A0 38 19 D6
        :          01 EE 96 92 B8 8B CA D7 FF E2 8E 23 D9 4A 8C CA
        :          F8 05 4A 98 29 E3 3C D2 34 3D A1 D0 A8 D4 1D E9
        :          4E 91 BE EF DC 50 56 96 5A 92 5D 95 09 D8 43 13
        :          0E 1C 76 9F 89 67 6D 65 C1 58 4E F0 40 70 31 51
        :          4F 13 19 38 17 74 F3 F7 8F 89 DB 2D 17 A9 5C 78
        :          4D 21 9B 78 9C DA 6A A3 D5 12 C2 55 FF E2 A3 04
        :          6A F5 05 45 0C 14 69 78 64 02 B0 7B 9F B8 EC 40
        :              [ Another 1824 bytes skipped ]
        :          }
2711   99:        [4] {
2713   97:         SEQUENCE {
2715   15:          SEQUENCE {
2717    3:           OBJECT IDENTIFIER basicConstraints (2 5 29 19)
2722    1:           BOOLEAN TRUE
```

```
2725    5:           OCTET STRING, encapsulates {
2727    3:            SEQUENCE {
2729    1:             BOOLEAN TRUE
         :              }
         :             }
         :            }
2732   14:           SEQUENCE {
2734    3:            OBJECT IDENTIFIER keyUsage (2 5 29 15)
2739    1:            BOOLEAN TRUE
2742    4:            OCTET STRING, encapsulates {
2744    2:             BIT STRING 1 unused bit
         :              '1100001'B
         :             }
         :            }
2748   29:           SEQUENCE {
2750    3:            OBJECT IDENTIFIER
         :             subjectKeyIdentifier (2 5 29 14)
2755   22:            OCTET STRING, encapsulates {
2757   20:             OCTET STRING
         :             49 74 6C 51 42 21 E5 9F BB 91 E4 A3 35 4B E3 01
         :             4D 90 90 8E
         :              }
         :             }
2779   31:           SEQUENCE {
2781    3:            OBJECT IDENTIFIER
         :             authorityKeyIdentifier (2 5 29 35)
2786   24:            OCTET STRING, encapsulates {
2788   22:             SEQUENCE {
2790   20:              [0]
         :              9B 07 B4 A4 75 C4 BC 91 5D 35 E0 C9 A1 C1 62 E2
         :              77 55 D6 3F
         :                 }
         :                }
         :               }
         :              }
         :             }
2812 3310:           BIT STRING
         :            60 AF 87 6F D4 86 6C 62 D5 46 5D 94 92 04 4A C7
         :            84 D0 1E 5D E6 5D 23 D7 4F A9 3F 90 C0 CF 32 B6
         :            F5 3F 82 74 70 8D 4F 11 FB BB 64 12 CC A5 48 EC
         :            F9 61 BA E7 EF 22 D1 B2 8B 43 F6 BF 87 45 81 DD
         :            5F BD 65 89 E1 0F 12 48 61 15 EB F4 73 E0 A5 FC
         :            3D 5E 31 46 CE 5E EB F6 9A DF DF 77 5A 72 22 11
         :            F7 C2 8D 39 A2 6C FE C4 92 21 E4 84 69 B2 C7 6C
         :            66 E2 9D 75 D7 D6 E3 57 29 BB 53 DE 16 93 D8 84
         :               [ Another 3181 bytes skipped ]
         :            }
         :           }
```

```
         :       }
         :      }
         :     }
         :    }
6126   10:  SEQUENCE {
6128    8:   OBJECT IDENTIFIER ecdsaWithSHA512 (1 2 840 10045 4 3 4)
         :    }
6138  139:  BIT STRING, encapsulates {
6142  135:   SEQUENCE {
6145   66:    INTEGER
         :     00 FD 58 DA BD BA 7F 18 52 62 62 FB A6 31 05 ED
         :     A3 FD 57 86 73 A1 57 E9 0D FE F9 57 12 93 5B AB
         :     C2 BF 3F C2 A3 36 21 CE 7E A6 66 46 B9 D7 C1 81
         :     46 09 97 0B 86 1E F3 CB 68 2C E8 5D C6 34 F7 D7
         :     DF E5
6213   65:    INTEGER
         :     15 CB 1A 68 75 80 25 8E 96 E4 C1 58 D3 E1 AE DB
         :     E3 28 A8 3C 9B D1 E9 2E 38 E9 D5 3D E7 9B 5B 9E
         :     CD CF 15 BC E1 39 4F 2E B0 58 72 5E C2 91 0C 7E
         :     CC 07 3B 7E 7B 0F 44 9C F1 1E 97 BB 9D E3 E1 76
         :     DC
         :     }
         :    }
         :   }
```

B.3. Dual use example

B.3.1. EC signing end-entity certificate

This is an end-entity signing certificate which certifies an EC key.

```
-----BEGIN CERTIFICATE-----
MIICYTCCAcOgAwIBAgIUVcVNficoipRs4c6JBiF731VtDLAwCgYIKoZIzj0EAwQw
gYsxCzAJBgNVBAYTAlhYMTUwMwYDVQQKDCxSb3lhbCBJbnN0aXR1dGUgb2YgUHVi
bGljIEtleSBJbmZyYXN0cnVjdHVyZTErMCkGA1UECwwiUG9zdC1IZWZmYWx1bXAg
UmVzZWFyY2ggRGVwYXJ0bWVudDEYMBYGA1UEAwwPRUNEU0EgUm9vdCAtIEcxMB4X
DTI0MTAxNzIzMzcyM1oXDTM0MTAxNTIzMzcyM1owLzELMAkGA1UEBhMCWFgxDzAN
BgNVBAoMBkhhbmFrbzEPMA0GA1UECwwGWWFtYWRhMFkwEwYHKoZIzj0CAQYIKoZI
zj0DAQcDQgAEbg5mK9aDw+9pIASgzCANcYRugXSfaWtTH3Kg6th/m8hybPvXHsFG
Enm4Zu3a+S/5RPmIw78UoBMpIqR+Tfno16NgMF4wDAYDVR0TAQH/BAIwADAOBgNV
HQ8BAf8EBAMCB4AwHQYDVR0OBBYEFKjGwfjydnErtBzOVMiLz5lP9Jq/MB8GA1Ud
IwQYMBaAFOuj0ItR/hLczCFmh4UPmMdnc4g0MAoGCCqGSM49BAMEA4GLADCBhwJB
O3d8oj0thpSmSI85xLuvA97w/QKRhdGXwPtzO7VceH3seMiORoCLPKO8Gfd1liRL
tznhz7IbmVbS64WbxQe4QawCQgFeT1babH2MEBLT+NGXIKA0azitP11LA/rynYoD
bindtP08txIa8w9O2MhG1706nrLc+z+PstQqXgQQ5ha/fn97PA==
-----END CERTIFICATE-----
```

```
  0 609: SEQUENCE {
  4 451:  SEQUENCE {
  8   3:   [0] {
 10   1:    INTEGER 2
      :     }
 13  20:   INTEGER 55 C5 4D 7E 27 28 8A 94 6C E1 CE 89 06 21 7B DF 55 6D 0C B0
 35  10:   SEQUENCE {
 37   8:    OBJECT IDENTIFIER ecdsaWithSHA512 (1 2 840 10045 4 3 4)
      :     }
 47 139:   SEQUENCE {
 50  11:    SET {
 52   9:     SEQUENCE {
 54   3:      OBJECT IDENTIFIER countryName (2 5 4 6)
 59   2:      PrintableString 'XX'
      :       }
      :      }
 63  53:    SET {
 65  51:     SEQUENCE {
 67   3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
 72  44:      UTF8String
      :       'Royal Institute of Public Key Infrastructure'
      :       }
      :      }
118  43:    SET {
120  41:     SEQUENCE {
122   3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
127  34:      UTF8String 'Post-Heffalump Research Department'
      :       }
      :      }
163  24:    SET {
165  22:     SEQUENCE {
167   3:      OBJECT IDENTIFIER commonName (2 5 4 3)
172  15:      UTF8String 'ECDSA Root - G1'
      :       }
      :      }
      :     }
189  30:   SEQUENCE {
191  13:    UTCTime 17/10/2024 23:37:23 GMT
206  13:    UTCTime 15/10/2034 23:37:23 GMT
      :     }
221  47:   SEQUENCE {
223  11:    SET {
225   9:     SEQUENCE {
227   3:      OBJECT IDENTIFIER countryName (2 5 4 6)
232   2:      PrintableString 'XX'
      :       }
      :      }
236  15:    SET {
```

```
238  13:     SEQUENCE {
240   3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
245   6:      UTF8String 'Hanako'
       :      }
       :     }
253  15:    SET {
255  13:     SEQUENCE {
257   3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
262   6:      UTF8String 'Yamada'
       :      }
       :     }
       :    }
270  89:   SEQUENCE {
272  19:    SEQUENCE {
274   7:     OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
283   8:     OBJECT IDENTIFIER prime256v1 (1 2 840 10045 3 1 7)
       :     }
293  66:    BIT STRING
       :     04 6E 0E 66 2B D6 83 C3 EF 69 20 04 A0 CC 20 0D
       :     71 84 6E 81 74 9F 69 6B 53 1F 72 A0 EA D8 7F 9B
       :     C8 72 6C FB D7 1E C1 46 12 79 B8 66 ED DA F9 2F
       :     F9 44 F9 88 C3 BF 14 A0 13 29 22 A4 7E 4D F9 E8
       :     D7
       :     }
361  96:   [3] {
363  94:    SEQUENCE {
365  12:     SEQUENCE {
367   3:      OBJECT IDENTIFIER basicConstraints (2 5 29 19)
372   1:      BOOLEAN TRUE
375   2:      OCTET STRING, encapsulates {
377   0:       SEQUENCE {}
       :       }
       :      }
379  14:     SEQUENCE {
381   3:      OBJECT IDENTIFIER keyUsage (2 5 29 15)
386   1:      BOOLEAN TRUE
389   4:      OCTET STRING, encapsulates {
391   2:       BIT STRING 7 unused bits
       :        '1'B (bit 0)
       :       }
       :      }
395  29:     SEQUENCE {
397   3:      OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
402  22:      OCTET STRING, encapsulates {
404  20:       OCTET STRING
       :        A8 C6 C1 F8 F2 76 71 2B B4 1C CE 54 C8 8B CF 99
       :        4F F4 9A BF
       :       }
```

```
       :       }
426  31:     SEQUENCE {
428   3:      OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
433  24:      OCTET STRING, encapsulates {
435  22:       SEQUENCE {
437  20:        [0]
       :         EB A3 D0 8B 51 FE 12 DC CC 21 66 87 85 0F 98 C7
       :         67 73 88 34
       :         }
       :        }
       :       }
       :      }
       :     }
       :    }
459  10:  SEQUENCE {
461   8:   OBJECT IDENTIFIER ecdsaWithSHA512 (1 2 840 10045 4 3 4)
       :    }
471 139:  BIT STRING, encapsulates {
475 135:   SEQUENCE {
478  65:    INTEGER
       :     3B 77 7C A2 3D 2D 86 94 A6 48 8F 39 C4 BB AF 03
       :     DE F0 FD 02 91 85 D1 97 C0 FB 73 3B B5 5C 78 7D
       :     EC 78 C8 8E 46 80 8B 3C A3 BC 19 F7 75 96 24 4B
       :     B7 39 E1 CF B2 1B 99 56 D2 EB 85 9B C5 07 B8 41
       :     AC
545  66:    INTEGER
       :     01 5E 4F 56 DA 6C 7D 8C 10 12 D3 F8 D1 97 20 A0
       :     34 6B 38 AD 3F 5D 4B 03 FA F2 9D 8A 03 6E 29 DD
       :     B4 FD 3C B7 12 1A F3 0F 4E D8 C8 46 D7 BD 3A 9E
       :     B2 DC FB 3F 8F B2 D4 2A 5E 04 10 E6 16 BF 7E 7F
       :     7B 3C
       :     }
       :    }
       :   }
```

B.3.2. EC dual use end-entity certificate with encoded Delta Certificate

This is an end-entity key exchange certificate which certifies an EC key. It contains a Delta Certificate Descriptor extension which includes sufficient information to the recreate the EC signing end-entity certificate.

```
   -----BEGIN CERTIFICATE-----
   MIIDzTCCAy6gAwIBAgIUczxcVsNa7M9uSs598vuGatGLDuIwCgYIKoZIzj0EAwQw
   gYsxCzAJBgNVBAYTAlhYMTUwMwYDVQQKDCxSb3lhbCBJbnN0aXR1dGUgb2YgUHVi
   bGljIEtleSBJbmZyYXN0cnVjdHVyZTErMCkGA1UECwwiUG9zdC1IZWZmYWx1bXAg
   UmVzZWFyY2ggRGVwYXJ0bWVudDEYMBYGA1UEAwwPRUNEU0EgUm9vdCAtIEcxMB4X
   DTI0MTAxNzIzMzcyM1oXDTM0MTAxNTIzMzcyM1owLzELMAkGA1UEBhMCWFgxDzAN
   BgNVBAoMBkhhbmFrbzEPMA0GA1UECwwGWWFtYWRhMHYwEAYHKoZIzj0CAQYFK4EE
   ACIDYgAE+qm8IaZ5hVFufLvTuniWWnQoa9d0YCyNiOmQ2OrrcukSy0FgozyJq7hc
   g8o2pJ5uRRLVysU1gHNfxL+TvwRRr6eWUJE8v0dCUccuCFPAVbxwf7Hjcp5NSsFn
   J2lIrvzgo4IBrDCCAagwDAYDVR0TAQH/BAIwADAOBgNVHQ8BAf8EBAMCAwgwHQYD
   VR0OBBYEFAHprr1J3zZ7gG1ksEzN8BHM7tCzMB8GA1UdIwQYMBaAFOuj0ItR/hLc
   zCFmh4UPmMdnc4g0MIIBRgYKYIZIAYb6a1AGAQSCATYwggEyAhRVxU1+JyiKlGzh
   zokGIXvfVW0MsDBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABG4OZivWg8PvaSAE
   oMwgDXGEboF0n2lrUx9yoOrYf5vIcmz71x7BRhJ5uGbt2vkv+UT5iMO/FKATKSKk
   fk356NekMTAvMA4GA1UdDwEB/wQEAwIHgDAdBgNVHQ4EFgQUqMbB+PJ2cSu0HM5U
   yIvPmU/0mr8DgYsAMIGHAkE7d3yiPS2GlKZIjznEu68D3vD9ApGF0ZfA+3M7tVx4
   fex4yI5GgIs8o7wZ93WWJEu3OeHPshuZVtLrhZvFB7hBrAJCAV5PVtpsfYwQEtP4
   0ZcgoDRrOK0/XUsD+vKdigNuKd20/Ty3EhrzD07YyEbXvTqestz7P4+y1CpeBBDm
   Fr9+f3s8MAoGCCqGSM49BAMEA4GMADCBiAJCAXrIaCetU/F7+TDkYBjEaHRZEujy
   DL2Ic08Eu+iDBRvzuYjxulQKCJaRFrcbegcW8D8MTkrJW8b0j9PkIXuLB51wAkIB
   0/4Tx4hhUQ6SCBNx70mG2kOeHpgZB62K3b3PtypOJtUWTZS5XgBhljUUTmdsaQtA
   wi1V+cwAnegmu168l43lQz0=
   -----END CERTIFICATE-----

  0 973: SEQUENCE {
  4 814:   SEQUENCE {
  8   3:     [0] {
 10   1:       INTEGER 2
       :       }
 13  20:     INTEGER 73 3C 5C 56 C3 5A EC CF 6E 4A CE 7D F2 FB 86 6A D1 8B 0E E2
 35  10:     SEQUENCE {
 37   8:       OBJECT IDENTIFIER ecdsaWithSHA512 (1 2 840 10045 4 3 4)
       :       }
 47 139:     SEQUENCE {
 50  11:       SET {
 52   9:         SEQUENCE {
 54   3:           OBJECT IDENTIFIER countryName (2 5 4 6)
 59   2:           PrintableString 'XX'
       :           }
       :         }
 63  53:       SET {
 65  51:         SEQUENCE {
 67   3:           OBJECT IDENTIFIER organizationName (2 5 4 10)
 72  44:           UTF8String
       :             'Royal Institute of Public Key Infrastructure'
       :           }
       :         }
118  43:       SET {
```

```
120  41:     SEQUENCE {
122   3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
127  34:      UTF8String 'Post-Heffalump Research Department'
       :      }
       :     }
163  24:    SET {
165  22:     SEQUENCE {
167   3:      OBJECT IDENTIFIER commonName (2 5 4 3)
172  15:      UTF8String 'ECDSA Root - G1'
       :      }
       :     }
       :    }
189  30:   SEQUENCE {
191  13:    UTCTime 17/10/2024 23:37:23 GMT
206  13:    UTCTime 15/10/2034 23:37:23 GMT
       :     }
221  47:   SEQUENCE {
223  11:    SET {
225   9:     SEQUENCE {
227   3:      OBJECT IDENTIFIER countryName (2 5 4 6)
232   2:      PrintableString 'XX'
       :      }
       :     }
236  15:    SET {
238  13:     SEQUENCE {
240   3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
245   6:      UTF8String 'Hanako'
       :      }
       :     }
253  15:    SET {
255  13:     SEQUENCE {
257   3:      OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
262   6:      UTF8String 'Yamada'
       :      }
       :     }
       :    }
270 118:   SEQUENCE {
272  16:    SEQUENCE {
274   7:     OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
283   5:     OBJECT IDENTIFIER secp384r1 (1 3 132 0 34)
       :      }
290  98:    BIT STRING
       :     04 FA A9 BC 21 A6 79 85 51 6E 7C BB D3 BA 78 96
       :     5A 74 28 6B D7 74 60 2C 8D 88 E9 90 D8 EA EB 72
       :     E9 12 CB 41 60 A3 3C 89 AB B8 5C 83 CA 36 A4 9E
       :     6E 45 12 D5 CA C5 35 80 73 5F C4 BF 93 BF 04 51
       :     AF A7 96 50 91 3C BF 47 42 51 C7 2E 08 53 C0 55
       :     BC 70 7F B1 E3 72 9E 4D 4A C1 67 27 69 48 AE FC
```

```
       :     E0
       :     }
390 428:   [3] {
394 424:    SEQUENCE {
398  12:     SEQUENCE {
400   3:      OBJECT IDENTIFIER basicConstraints (2 5 29 19)
405   1:      BOOLEAN TRUE
408   2:      OCTET STRING, encapsulates {
410   0:       SEQUENCE {}
       :       }
       :      }
412  14:     SEQUENCE {
414   3:      OBJECT IDENTIFIER keyUsage (2 5 29 15)
419   1:      BOOLEAN TRUE
422   4:      OCTET STRING, encapsulates {
424   2:       BIT STRING 3 unused bits
       :        '10000'B (bit 4)
       :       }
       :      }
428  29:     SEQUENCE {
430   3:      OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
435  22:      OCTET STRING, encapsulates {
437  20:       OCTET STRING
       :        01 E9 AE BD 49 DF 36 7B 80 6D 64 B0 4C CD F0 11
       :        CC EE D0 B3
       :       }
       :      }
459  31:     SEQUENCE {
461   3:      OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
466  24:      OCTET STRING, encapsulates {
468  22:       SEQUENCE {
470  20:        [0]
       :         EB A3 D0 8B 51 FE 12 DC CC 21 66 87 85 0F 98 C7
       :         67 73 88 34
       :        }
       :       }
       :      }
492 326:     SEQUENCE {
496  10:      OBJECT IDENTIFIER
       :       deltaCertificateDescriptor (2 16 840 1 114027 80 6 1)
508 310:      OCTET STRING, encapsulates {
512 306:       SEQUENCE {
516  20:        INTEGER
       :         55 C5 4D 7E 27 28 8A 94 6C E1 CE 89 06 21 7B DF
       :         55 6D 0C B0
538  89:        SEQUENCE {
540  19:         SEQUENCE {
542   7:          OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
```

```
551   8:          OBJECT IDENTIFIER prime256v1 (1 2 840 10045 3 1 7)
       :          }
561  66:         BIT STRING
       :          04 6E 0E 66 2B D6 83 C3 EF 69 20 04 A0 CC 20 0D
       :          71 84 6E 81 74 9F 69 6B 53 1F 72 A0 EA D8 7F 9B
       :          C8 72 6C FB D7 1E C1 46 12 79 B8 66 ED DA F9 2F
       :          F9 44 F9 88 C3 BF 14 A0 13 29 22 A4 7E 4D F9 E8
       :          D7
       :          }
629  49:        [4] {
631  47:         SEQUENCE {
633  14:          SEQUENCE {
635   3:           OBJECT IDENTIFIER keyUsage (2 5 29 15)
640   1:           BOOLEAN TRUE
643   4:           OCTET STRING, encapsulates {
645   2:            BIT STRING 7 unused bits
       :             '1'B (bit 0)
       :            }
       :           }
649  29:          SEQUENCE {
651   3:           OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
656  22:           OCTET STRING, encapsulates {
658  20:            OCTET STRING
       :            A8 C6 C1 F8 F2 76 71 2B B4 1C CE 54 C8 8B CF 99
       :            4F F4 9A BF
       :            }
       :           }
       :          }
       :         }
680 139:        BIT STRING, encapsulates {
684 135:         SEQUENCE {
687  65:          INTEGER
       :           3B 77 7C A2 3D 2D 86 94 A6 48 8F 39 C4 BB AF 03
       :           DE F0 FD 02 91 85 D1 97 C0 FB 73 3B B5 5C 78 7D
       :           EC 78 C8 8E 46 80 8B 3C A3 BC 19 F7 75 96 24 4B
       :           B7 39 E1 CF B2 1B 99 56 D2 EB 85 9B C5 07 B8 41
       :           AC
754  66:          INTEGER
       :           01 5E 4F 56 DA 6C 7D 8C 10 12 D3 F8 D1 97 20 A0
       :           34 6B 38 AD 3F 5D 4B 03 FA F2 9D 8A 03 6E 29 DD
       :           B4 FD 3C B7 12 1A F3 0F 4E D8 C8 46 D7 BD 3A 9E
       :           B2 DC FB 3F 8F B2 D4 2A 5E 04 10 E6 16 BF 7E 7F
       :           7B 3C
       :           }
       :          }
       :         }
       :        }
       :       }
```

```
       :      }
       :     }
       :    }
822  10:  SEQUENCE {
824   8:   OBJECT IDENTIFIER ecdsaWithSHA512 (1 2 840 10045 4 3 4)
       :    }
834 140:  BIT STRING, encapsulates {
838 136:   SEQUENCE {
841  66:    INTEGER
       :     01 7A C8 68 27 AD 53 F1 7B F9 30 E4 60 18 C4 68
       :     74 59 12 E8 F2 0C BD 88 73 4F 04 BB E8 83 05 1B
       :     F3 B9 88 F1 BA 54 0A 08 96 91 16 B7 1B 7A 07 16
       :     F0 3F 0C 4E 4A C9 5B C6 F4 8F D3 E4 21 7B 8B 07
       :     9D 70
909  66:    INTEGER
       :     01 D3 FE 13 C7 88 61 51 0E 92 08 13 71 EF 49 86
       :     DA 43 9E 1E 98 19 07 AD 8A DD BD CF B7 2A 4E 26
       :     D5 16 4D 94 B9 5E 00 61 96 35 14 4E 67 6C 69 0B
       :     40 C2 2D 55 F9 CC 00 9D E8 26 BB 5E BC 97 8D E5
       :     43 3D
       :     }
       :    }
       :   }
```

Acknowledgments

TODO acknowledge.

Authors' Addresses

C. Bonnell
DigiCert
Email: corey.bonnell@digicert.com

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

D. Hook
KeyFactor
Email: david.hook@keyfactor.com

T. Okubo
DigiCert
Email: tomofumi.okubo@digicert.com

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