Stream: Internet Engineering Task Force (IETF)
RFC: 9883
Category: Standards Track
Published: October 2025
ISSN: 2070-1721
Author: R. Housley
Vigil Security

RFC 9883
An Attribute for Statement of Possession of a Private Key

Abstract

This document specifies an attribute for a statement of possession of a private key by a certificate subject. As part of X.509 certificate enrollment, a Certification Authority (CA) typically demands proof that the subject possesses the private key that corresponds to the to-be-certified public key. In some cases, a CA might accept a signed statement from the certificate subject. For example, when a certificate subject needs separate certificates for signature and key establishment, a statement that can be validated with the previously issued signature certificate for the same subject might be adequate for subsequent issuance of the key establishment certificate.

Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9883.

Copyright Notice

Table of Contents

1. Introduction 2
1.1. ASN.1 3
1.2. Terminology 3
2. Overview 3
3. Attribute for Statement of Possession of a Private Key 4
4. Conventions for PKCS#10 5
5. Conventions for CRMF 5
6. Security Considerations 6
7. IANA Considerations 7
8. References 7
8.1. Normative References 7
8.2. Informative References 8
Appendix A. ASN.1 Module 8
Appendix B. Example Use of the privateKeyPossessionStatement Attribute 9
Acknowledgements 17
Author's Address 17

1. Introduction

This document specifies an attribute for a statement of possession of a private key by a certificate subject. X.509 certificate [RFC5280] enrollment often depends on PKCS#10 [RFC2986] or the Certificate Request Message Format (CRMF) [RFC4211]. As part of enrollment, a Certification Authority (CA) typically demands proof that the subject possesses the private key that corresponds to the to-be-certified public key. Alternatively, a CA may accept a signed statement from the certificate subject claiming knowledge of that private key. When a certificate subject needs separate certificates for signature and key establishment, a signed statement that can be validated with the previously issued signature certificate for the same subject might be adequate for subsequent issuance of the key establishment certificate.

For example, a subject may need a signature certificate that contains an ML-DSA (Module-Lattice-Based Digital Signature Algorithm) public key and a key establishment certificate that contains an ML-KEM (Module-Lattice-Based Key-Encapsulation Mechanism) public key. For another example, a subject may need a signature certificate that contains an ECDSA (Elliptic Curve Digital Signature Algorithm) public key and a key establishment certificate that contains an ECDH (Elliptic Curve Diffie-Hellman) public key.

A statement of possession may be used in lieu of the usual proof-of-possession mechanisms. The statement is simply a signed assertion that the requestor of a key establishment certificate has possession of the key establishment private key and that statement is signed using a signature private key that was previously shown to be in the possession of the same certificate subject. If allowed by the Certificate Policy [RFC3647], the CA is permitted to accept this statement in lieu of proof that the requestor has possession of the private key, such as [RFC6955].

Note that [RFC6955] offers some algorithms that provide proof of possession for Diffie-Hellman private keys; however, these algorithms are not suitable for use with PKCS#10 [RFC2986]. In addition, the algorithms in [RFC6955] do not support key encapsulation mechanism algorithms, such as ML-KEM. The attribute specified in this document, on the other hand, is suitable for use with both PKCS#10 and the CRMF [RFC4211].

1.1. ASN.1

The attribute defined in this document is generated using ASN.1 [X680], using the Distinguished Encoding Rules (DER) [X690].

1.2. Terminology

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. Overview

When using the attribute defined in this document to make a statement about the possession of the key establishment private key, the process to obtain two certificates with PKCS#10 is as follows:

1. The subject generates the signature key pair.
2. The subject composes a PKCS#10 Certificate Signing Request (CSR) in the usual manner. It includes a signature that is produced with the private key from step 1.
3. The subject sends the CSR to the CA, and it gets back a signature certificate. The signature certificate includes a key usage of digitalSignature, nonRepudiation, or both (see Section 4.2.1.3 of [RFC5280]).
4. The subject generates the key establishment key pair.

5. The subject composes a PKCS#10 CSR containing the key establishment public key. The CSR attributes include the attribute specified in Section 3 of this document. The subject name matches the one from step 3. The CSR includes a signature that is produced with the private key from step 1.
6. The subject sends the CSR to the CA, and it gets back a key establishment certificate. The key establishment certificate includes a key usage of keyEncipherment or keyAgreement (see Section 4.2.1.3 of [RFC5280]).

In general, the issuer of the key establishment certificate will be the same as the issuer of the signature certificate. If the issuers of the two certificates will be different, then the certificate policy of the issuer of the key establishment certificate **MUST** explain the procedure that is used to verify the subject and subject alternative names.

3. Attribute for Statement of Possession of a Private Key

The attribute for statement of possession of a private key is included in a certificate request to make the following statement:

> The subject of the signature certificate that is used to validate the signature on this certificate request states, without providing proof, that it has possession of the private key that corresponds to the public key in the certificate request.

The CA **MUST** perform certification path validation for the signature certificate as specified in Section 6 of [RFC5280]. If the certification path is not valid, then the CA **MUST** reject the request for the key establishment certificate.

The CA **MUST** validate the signature on the certificate request using the public key from the signature certificate. If the signature is not valid, then the CA **MUST** reject the certificate request.

The subject in the signature certificate **SHOULD** be the same as the subject name in the certificate request. If they are different, the certificate policy **MUST** describe how the CA can determine that the two subject names identify the same entity. If the CA is unable to determine that the two subject names identify the same entity, then the CA **MUST** reject the certificate request.

If subject alternative names are present in the certificate request, they **SHOULD** match subject alternative names in the signature certificate. If they are different, the certificate policy **MUST** describe how the CA can determine that the two subject alternative names identify the same entity. If the CA is unable to determine that each of subject alternative names identifies the same entity as is named in the signature certificate, then the CA **MUST** reject the certificate request.

When the CA rejects a certificate request for any of the reasons listed above, the CA should provide information to the requestor about the reason for the rejection to aid with diagnostic efforts. Likewise, the CA should log the rejection events.

The attribute for statement of possession of a private key has the following structure:

```
id-at-statementOfPossession OBJECT IDENTIFIER ::=
  { 1 3 6 1 4 1 22112 2 1 }

privateKeyPossessionStatement ATTRIBUTE ::= {
  TYPE PrivateKeyPossessionStatement
  IDENTIFIED BY id-at-statementOfPossession }

PrivateKeyPossessionStatement ::= SEQUENCE {
  signer  IssuerAndSerialNumber,
  cert    Certificate OPTIONAL }
```

The components of the PrivateKeyStatement SEQUENCE have the following semantics:

signer: The issuer name and certificate serial number of the signature certificate.

cert: The signature certificate. If the issuer of the key establishment certificate will be the same as the issuer of the signature certificate, then this component **MAY** be omitted. When the signature certificate is omitted, the signer is assuming that the CA has a mechanism to obtain all valid certificates that it issued.

4. Conventions for PKCS#10

This section specifies the conventions for using the attribute for statement of possession of a private key with PKCS#10 [RFC2986] when requesting a key establishment certificate.

The PKCS#10 CertificationRequest always has three components, as follows:

certificationRequestInfo: The subject name **SHOULD** be the same as the subject name in the signature certificate, the subjectPKInfo **MUST** contain the public key for the key establishment algorithm, and the attributes **MUST** include privateKeyPossessionStatement attribute as specified in Section 3 of this document.

signatureAlgorithm: The signature algorithm **MUST** be one that can be validated with the public key in the signature certificate.

signature: The signature over certificationRequestInfo **MUST** validate with the public key in the signature certificate, and certification path validation for the signature certificate **MUST** be successful as specified in Section 6 of [RFC5280].

5. Conventions for CRMF

This section specifies the conventions for using the attribute for statement of possession of a private key with the CRMF [RFC4211] when requesting a key establishment certificate.

The following ASN.1 types are defined for use with CRMF. They have exactly the same semantics and syntax as the attribute discussed above, but they offer a similar naming convention to the Registration Controls in [RFC4211].

```
regCtrl-privateKeyPossessionStatement ATTRIBUTE ::=
  privateKeyPossessionStatement

id-regCtrl-statementOfPossession OBJECT IDENTIFIER ::=
  id-at-statementOfPossession
```

The CRMF CertificationRequest always has three components, as follows:

certReq: The certTemplate **MUST** include the subject and the publicKey components. The same subject name **SHOULD** match the subject name in the signature certificate, and publicKey **MUST** contain the public key for the key establishment algorithm.

popo: The ProofOfPossession **MUST** use the signature CHOICE, the poposkInput **MUST** be present, POPOSigningKeyInput.authInfo **MUST** use the sender CHOICE, the sender **SHOULD** be set to the subject name that appears in the signature certificate, the publicKey **MUST** contain a copy of the public key from the certTemplate, the algorithmIdentifier **MUST** identify a signature algorithm that can be validated with the public key in the signature certificate, the signature over the poposkInput **MUST** validate with the public key in the signature certificate, and certification path validation for the signature certificate **MUST** be successful as specified in Section 6 of [RFC5280].

regInfo: The attributes **MUST** include the privateKeyPossessionStatement attribute as specified in Section 3 of this document.

6. Security Considerations

The privateKeyPossessionStatement attribute **MUST NOT** be used to obtain a signature certificate. Performing proof of possession of the signature private key is easily accomplished by signing the certificate request.

The subject is signing the privateKeyPossessionStatement attribute to tell the CA that it has possession of the key establishment private key. This is being done instead of providing technical proof of possession. If the subject has lost control of the signature private key, then the signed privateKeyPossessionStatement attribute could be generated by some other party. Timely revocation of the compromised signature certificate is the only protection against such loss of control.

If the CA revokes a compromised signature certificate, then the CA **SHOULD** also revoke all key establishment certificates that were obtained with privateKeyPossessionStatement attributes signed by that compromised signature certificate.

The signature key pair and the key establishment key pair are expected to have roughly the same security strength. To ensure that the signature on the statement is not the weakest part of the certificate enrollment, the signature key pair **SHOULD** be at least as strong as the key establishment key pair.

If a CA allows a subject in the key establishment certificate to be different than the subject name in the signature certificate, then certificate policy **MUST** describe how to determine that the two subject names identify the same entity. Likewise, if a CA allows subject alternative names in the key establishment certificate that are not present in the signature certificate, then certificate policy **MUST** describe how to determine that the subject alternative names identify the same entity as is named in the signature certificate.

7. IANA Considerations

For the ASN.1 Module in Appendix A of this document, IANA has assigned an object identifier (OID) for the module identifier (118) with a Description of "id-mod-private-key-possession-stmt-2025" 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/info/rfc2119>.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, DOI 10.17487/RFC2986, November 2000, <https://www.rfc-editor.org/info/rfc2986>.

[RFC4211] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, DOI 10.17487/RFC4211, September 2005, <https://www.rfc-editor.org/info/rfc4211>.

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

[RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <https://www.rfc-editor.org/info/rfc5912>.

[RFC6268] Schaad, J. and S. Turner, "Additional New ASN.1 Modules for the Cryptographic Message Syntax (CMS) and the Public Key Infrastructure Using X.509 (PKIX)", RFC 6268, DOI 10.17487/RFC6268, July 2011, <https://www.rfc-editor.org/info/rfc6268>.

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

[X680] ITU-T, "Information technology -- Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T Recommendation X.680, ISO/IEC 8824-1:2021, February 2021, <https://www.itu.int/rec/T-REC-X.680>.

[X690] ITU-T, "Information technology -- ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, ISO/IEC 8825-1:2021, February 2021, <https://www.itu.int/rec/T-REC-X.690>.

8.2. Informative References

[RFC3647] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC 3647, DOI 10.17487/RFC3647, November 2003, <https://www.rfc-editor.org/info/rfc3647>.

[RFC6955] Schaad, J. and H. Prafullchandra, "Diffie-Hellman Proof-of-Possession Algorithms", RFC 6955, DOI 10.17487/RFC6955, May 2013, <https://www.rfc-editor.org/info/rfc6955>.

Appendix A. ASN.1 Module

This ASN.1 Module uses the conventions established by [RFC5912] and [RFC6268].

```
<CODE BEGINS>
PrivateKeyPossessionStatement-2025
  { iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-private-key-possession-stmt-2025(118) }

DEFINITIONS IMPLICIT TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS
  ATTRIBUTE
  FROM PKIX-CommonTypes-2009 -- in [RFC5912]
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-pkixCommon-02(57) }

  Certificate
  FROM PKIX1Explicit-2009 -- in [RFC5912]
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-pkix1-explicit-02(51) }
```

```
  IssuerAndSerialNumber
  FROM CryptographicMessageSyntax-2010 -- [RFC6268]
    { iso(1) member-body(2) us(840) rsadsi(113549)
       pkcs(1) pkcs-9(9) smime(16) modules(0)
       id-mod-cms-2009(58) } ;

--
-- Private Key Possession Statement Attribute
--

id-at-statementOfPossession OBJECT IDENTIFIER ::=
  { 1 3 6 1 4 1 22112 2 1 }

privateKeyPossessionStatement ATTRIBUTE ::= {
  TYPE PrivateKeyPossessionStatement
  IDENTIFIED BY id-at-statementOfPossession }

PrivateKeyPossessionStatement ::= SEQUENCE {
  signer  IssuerAndSerialNumber,
  cert    Certificate OPTIONAL }

--
-- Registration Control Support
--

RegControlSet ATTRIBUTE ::=
  { regCtrl-privateKeyPossessionStatement, ... }

regCtrl-privateKeyPossessionStatement ATTRIBUTE ::=
  privateKeyPossessionStatement

id-regCtrl-statementOfPossession OBJECT IDENTIFIER ::=
  id-at-statementOfPossession

END
<CODE ENDS>
```

Appendix B. Example Use of the privateKeyPossessionStatement Attribute

In this example, the self-signed certificate for the CA is as follows:

```
-----BEGIN CERTIFICATE-----
MIIB7DCCAXKgAwIBAgIUL149AUxHunELBZMELEQm+isgKCQwCgYIKoZIzj0EAwMw
NzELMAkGA1UEBhMCVVMxEzARBgNVBAoTCkV4YW1wbGUgQ0ExEzARBgNVBAMTCmNh
LmV4YW1wbGUwHhcNMjUwMTAzMjAyNzA5WhcNMzUwMTAzMjAyNzA5WjA3MQswCQYD
VQQGEwJVUzETMBEGA1UEChMKRXhhbXBsZSBDQTETMBEGA1UEAxMKY2EuZXhhbXBs
ZTB2MBAGByqGSM49AgEGBSuBBAAiA2IABDxZdB/Glcxdk1p6Jf1j5en6QfliY9OS
fjZbtje/w6M58PN8Sb3VFln1rPdvD17UXeazSG9Hr/Dq3enbsHHO0pPntcFOgb8n
r8R8LUGhxRzjlxkaEJN+pa6Nf7qk49JDeaM/MD0wDwYDVR0TAQH/BAUwAwEB/zAL
BgNVHQ8EBAMCAgQwHQYDVR0OBBYEFD6YvLLv3DQbvnGS0qP6bbzyZkCqMAoGCCqG
SM49BAMDA2gAMGUCMGfb61IigoJ3QDnlsRdoktREHe0Dpm6DKw3qOyLL6A0cFK9Z
g8m11xIwvptlran52gIxAK1VrOjzRsFiHRptO+gFXstTXnQkKBb2/3WQz2SqcIS/
BWEp+siJ19OXOlz6APDB7w==
-----END CERTIFICATE-----
```

Alice generates her ECDSA signature key pair. Then, Alice composes a PKCS#10 Certificate Signing Request (CSR) in the usual manner as specified in [RFC2986]. The CSR includes a signature that is produced with her ECDSA private key. The CSR is as follows:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBhTCCAQsCAQAwPDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAlZBMRAwDgYDVQQH
EwdIZXJuZG9uMQ4wDAYDVQQDEwVBbGljZTB2MBAGByqGSM49AgEGBSuBBAAiA2IA
BIAc+6lXN1MIM/82QeWNb55H0zr+lVgWVeF0bf4jzxCb5MCjVaM0eFEvcjXMV5p4
kzqiJTHC0V2JAoqYMX/DMFIcwZ7xP9uQd9ep6KZ+RXut211L8+W1QI1QJSDNxANR
saBQME4GCSqGSIb3DQEJDjFBMD8wDAYDVR0TAQH/BAIwADALBgNVHQ8EBAMCB4Aw
IgYDVR0RBBswGYEXYWxpY2VAZW1haWwuZXhhbXBsZS5jb20wCgYIKoZIzj0EAwMD
aAAwZQIwPa2rOCe60edAF43C/t57IW8liyy+69FE04hMAFgw3Ga+nR+8zDuUsVLw
xXGAHtcDAjEA6LbvNkZjo6j2z5xRIjrHzEbGgiV4MF4xtnpfSSRI4dB0zT52bWkj
TZsuS1YWIkjt
-----END CERTIFICATE REQUEST-----
```

The CA issues a signature certificate to Alice:

```
-----BEGIN CERTIFICATE-----
MIICJzCCAa6gAwIBAgIUf3Sj/ANs4hR4XFlhTm+N8kxHqHkwCgYIKoZIzj0EAwMw
NzELMAkGA1UEBhMCVVMxEzARBgNVBAoTCkV4YW1wbGUgQ0ExEzARBgNVBAMTCmNh
LmV4YW1wbGUwHhcNMjUwMTA5MTcwMzQ4WhcNMjYwMTA5MTcwMzQ4WjA8MQswCQYD
VQQGEwJVUzELMAkGA1UECBMCVkExEDAOBgNVBAcTB0hlcm5kb24xDjAMBgNVBAMT
BUFsaWNlMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEgBz7qVc3Uwgz/zZB5Y1vnkfT
Ov6VWBZV4XRt/iPPEJvkwKNVozR4US9yNcxXmniTOqIlMcLRXYkCipgxf8MwUhzB
nvE/25B316nopn5Fe63bXUvz5bVAjVAlIM3EA1Gxo3YwdDAMBgNVHRMBAf8EAjAA
MAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUIx0A0f7tCzkQEZgYzH3NcM2L05IwHwYD
VR0jBBgwFoAUPpi8su/cNBu+cZLSo/ptvPJmQKowFwYDVR0gBBAwDjAMBgpghkgB
ZQMCATAwMAoGCCqGSM49BAMDA2cAMGQCMGu/Uypd7BaVnUjB36UtX9m5ZmPi78y5
1RA8WhbOv0KQVrcYtj4qOdiMVKBcoVceyAIwRJ6U91048NAb3nicHcrGFf1UYrhb
DlytK4tCa5HBxD/qAgy4/eUzA5NZwVaLK78u
-----END CERTIFICATE-----
```

Alice generates her ECDH key establishment key pair. Then, Alice composes a PKCS#10 CSR. The CSR attributes include the privateKeyPossessionStatement attribute, which points to her ECDSA signature certificate. The CSR includes her ECDH public key and a signature that is produced with her ECDSA private key. The CSR is as follows:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIEMTCCA7gCAQAwPDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAlZBMRAwDgYDVQQH
EwdIZXJuZG9uMQ4wDAYDVQQDEwVBbGljZTB0MA4GBSuBBAEMBgUrgQQAIgNiAAQB
RyQTH+cq1s5F94uFqFe7l1LqGdEC8Tm+e5VYBCfKAC8MJySQMj1GixEEXL+1Wjtg
23XvnJouCDoxSpDCSMqf3kvp5+naM37uxa3ZYgD6DPY3me5EZvyZPvSRJTFl/Bag
ggL9MGcGCSqGSIb3DQEJDjFaMFgwDAYDVR0TAQH/BAIwADALBgNVHQ8EBAMCAwgw
IgYDVR0RBBswGYEXYWxpY2VAZW1haWwuZXhhbXBsZS5jb20wFwYDVR0gBBAwDjAM
BgpghkgBZQMCATAwMIICkAYKKwYBBAGBrGACATGCAoAwggJ8ME8wNzELMAkGA1UE
BhMCVVMxEzARBgNVBAoTCkV4YW1wbGUgQ0ExEzARBgNVBAMTCmNhLmV4YW1wbGUC
FH90o/wDbOIUeFxZYU5vjfJMR6h5MIICJzCCAa6gAwIBAgIUf3Sj/ANs4hR4XFlh
Tm+N8kxHqHkwCgYIKoZIzj0EAwMwNzELMAkGA1UEBhMCVVMxEzARBgNVBAoTCkV4
YW1wbGUgQ0ExEzARBgNVBAMTCmNhLmV4YW1wbGUwHhcNMjUwMTA5MTcwMzQ4WhcN
MjYwMTA5MTcwMzQ4WjA8MQswCQYDVQQGEwJVUzELMAkGA1UECBMCVkExEDAOBgNV
BAcTB0hlcm5kb24xDjAMBgNVBAMTBUFsaWNlMHYwEAYHKoZIzj0CAQYFK4EEACID
YgAEgBz7qVc3Uwgz/zZB5Y1vnkfTOv6VWBZV4XRt/iPPEJvkwKNVozR4US9yNcxX
mniTOqIlMcLRXYkCipgxf8MwUhzBnvE/25B316nopn5Fe63bXUvz5bVAjVAlIM3E
A1Gxo3YwdDAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUIx0A
0f7tCzkQEZgYzH3NcM2L05IwHwYDVR0jBBgwFoAUPpi8su/cNBu+cZLSo/ptvPJm
QKowFwYDVR0gBBAwDjAMBgpghkgBZQMCATAwMAoGCCqGSM49BAMDA2cAMGQCMGu/
Uypd7BaVnUjB36UtX9m5ZmPi78y51RA8WhbOv0KQVrcYtj4qOdiMVKBcoVceyAIw
RJ6U91048NAb3nicHcrGFf1UYrhbDlytK4tCa5HBxD/qAgy4/eUzA5NZwVaLK78u
MAoGCCqGSM49BAMDA2cAMGQCL2TNHPULWcCS2DqZCCiQeSwx2JPLMI14Vi977bzy
rImq5p0H3Bel6fAS8BnQ00WNAjEAhHDAlcbRuHhqdW6mOgDd5kWEGGqgixIuvEEc
fVbnNCEyEE4n0mQ99PHURnXoHwqF
-----END CERTIFICATE REQUEST-----
```

The CSR decodes to the following:

```
   0 1073: SEQUENCE {
   4  952:  SEQUENCE {
   8    1:   INTEGER 0
  11   60:   SEQUENCE {
  13   11:    SET {
  15    9:     SEQUENCE {
  17    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
  22    2:      PrintableString 'US'
         :      }
         :     }
  26   11:    SET {
  28    9:     SEQUENCE {
  30    3:      OBJECT IDENTIFIER stateOrProvinceName (2 5 4 8)
  35    2:      PrintableString 'VA'
         :      }
         :     }
  39   16:    SET {
  41   14:     SEQUENCE {
  43    3:      OBJECT IDENTIFIER localityName (2 5 4 7)
  48    7:      PrintableString 'Herndon'
         :      }
         :     }
  57   14:    SET {
  59   12:     SEQUENCE {
  61    3:      OBJECT IDENTIFIER commonName (2 5 4 3)
  66    5:      PrintableString 'Alice'
         :      }
         :     }
         :    }
  73  116:   SEQUENCE {
  75   14:    SEQUENCE {
  77    5:     OBJECT IDENTIFIER ECDH (1 3 132 1 12)
  84    5:     OBJECT IDENTIFIER secp384r1 (1 3 132 0 34)
         :     }
  91   98:    BIT STRING
         :     04 01 47 24 13 1F E7 2A D6 CE 45 F7 8B 85 A8 57
         :     BB 97 52 EA 19 D1 02 F1 39 BE 7B 95 58 04 27 CA
         :     00 2F 0C 27 24 90 32 3D 46 8B 11 04 5C BF B5 5A
         :     3B 60 DB 75 EF 9C 9A 2E 08 3A 31 4A 90 C2 48 CA
         :     9F DE 4B E9 E7 E9 DA 33 7E EE C5 AD D9 62 00 FA
         :     0C F6 37 99 EE 44 66 FC 99 3E F4 91 25 31 65 FC
         :     16
         :    }
 191  765:   [0] {
 195  103:    SEQUENCE {
 197    9:     OBJECT IDENTIFIER
         :      extensionRequest (1 2 840 113549 1 9 14)
 208   90:     SET {
 210   88:      SEQUENCE {
 212   12:       SEQUENCE {
 214    3:        OBJECT IDENTIFIER
         :         basicConstraints (2 5 29 19)
 219    1:        BOOLEAN TRUE
 222    2:        OCTET STRING, encapsulates {
 224    0:         SEQUENCE {}
         :         }
         :        }
```

```
  226   11:       SEQUENCE {
  228    3:        OBJECT IDENTIFIER keyUsage (2 5 29 15)
  233    4:        OCTET STRING, encapsulates {
  235    2:         BIT STRING 3 unused bits
          :          '10000'B (bit 4)
          :         }
          :        }
  239   34:       SEQUENCE {
  241    3:        OBJECT IDENTIFIER subjectAltName (2 5 29 17)
  246   27:        OCTET STRING, encapsulates {
  248   25:         SEQUENCE {
  250   23:          [1] 'alice@email.example.com'
          :         }
          :        }
          :       }
  275   23:       SEQUENCE {
  277    3:        OBJECT IDENTIFIER
          :         certificatePolicies (2 5 29 32)
  282   16:        OCTET STRING, encapsulates {
  284   14:         SEQUENCE {
  286   12:          SEQUENCE {
  288   10:           OBJECT IDENTIFIER
          :            testCertPolicy (2 16 840 1 101 3 2 1 48 48)
          :           }
          :          }
          :         }
          :        }
          :       }
          :      }
  300  656:     SEQUENCE {
  304   10:      OBJECT IDENTIFIER
          :       statementOfPossession (1 3 6 1 4 1 22112 2 1)
  316  640:      SET {
  320  636:       SEQUENCE {
  324   79:        SEQUENCE {
  326   55:         SEQUENCE {
  328   11:          SET {
  330    9:           SEQUENCE {
  332    3:            OBJECT IDENTIFIER countryName (2 5 4 6)
  337    2:            PrintableString 'US'
          :           }
          :          }
  341   19:          SET {
  343   17:           SEQUENCE {
  345    3:            OBJECT IDENTIFIER
          :             organizationName (2 5 4 10)
  350   10:            PrintableString 'Example CA'
          :           }
          :          }
  362   19:          SET {
  364   17:           SEQUENCE {
  366    3:            OBJECT IDENTIFIER commonName (2 5 4 3)
  371   10:            PrintableString 'ca.example'
          :           }
          :          }
          :         }
  383   20:         INTEGER
```

```
          :      7F 74 A3 FC 03 6C E2 14 78 5C 59 61 4E 6F 8D F2
          :      4C 47 A8 79
          :         }
  405  551:       SEQUENCE {
  409  430:        SEQUENCE {
  413    3:         [0] {
  415    1:          INTEGER 2
          :           }
  418   20:         INTEGER
          :      7F 74 A3 FC 03 6C E2 14 78 5C 59 61 4E 6F 8D F2
          :      4C 47 A8 79
  440   10:         SEQUENCE {
  442    8:          OBJECT IDENTIFIER
          :           ecdsaWithSHA384 (1 2 840 10045 4 3 3)
          :           }
  452   55:         SEQUENCE {
  454   11:          SET {
  456    9:           SEQUENCE {
  458    3:            OBJECT IDENTIFIER
          :             countryName (2 5 4 6)
  463    2:            PrintableString 'US'
          :             }
          :            }
  467   19:          SET {
  469   17:           SEQUENCE {
  471    3:            OBJECT IDENTIFIER
          :             organizationName (2 5 4 10)
  476   10:            PrintableString 'Example CA'
          :             }
          :            }
  488   19:          SET {
  490   17:           SEQUENCE {
  492    3:            OBJECT IDENTIFIER
          :             commonName (2 5 4 3)
  497   10:            PrintableString 'ca.example'
          :             }
          :            }
          :           }
  509   30:         SEQUENCE {
  511   13:          UTCTime 09/01/2025 17:03:48 GMT
  526   13:          UTCTime 09/01/2026 17:03:48 GMT
          :           }
  541   60:         SEQUENCE {
  543   11:          SET {
  545    9:           SEQUENCE {
  547    3:            OBJECT IDENTIFIER
          :             countryName (2 5 4 6)
  552    2:            PrintableString 'US'
          :             }
          :            }
  556   11:          SET {
  558    9:           SEQUENCE {
  560    3:            OBJECT IDENTIFIER
          :             stateOrProvinceName (2 5 4 8)
  565    2:            PrintableString 'VA'
          :             }
          :            }
  569   16:          SET {
```

```
  571   14:           SEQUENCE {
  573    3:            OBJECT IDENTIFIER
          :             localityName (2 5 4 7)
  578    7:            PrintableString 'Herndon'
          :            }
          :           }
  587   14:          SET {
  589   12:           SEQUENCE {
  591    3:            OBJECT IDENTIFIER
          :             commonName (2 5 4 3)
  596    5:            PrintableString 'Alice'
          :            }
          :           }
          :          }
  603  118:         SEQUENCE {
  605   16:          SEQUENCE {
  607    7:           OBJECT IDENTIFIER
          :            ecPublicKey (1 2 840 10045 2 1)
  616    5:           OBJECT IDENTIFIER
          :            secp384r1 (1 3 132 0 34)
          :           }
  623   98:          BIT STRING
          :      04 80 1C FB A9 57 37 53 08 33 FF 36 41 E5 8D 6F
          :      9E 47 D3 3A FE 95 58 16 55 E1 74 6D FE 23 CF 10
          :      9B E4 C0 A3 55 A3 34 78 51 2F 72 35 CC 57 9A 78
          :      93 3A A2 25 31 C2 D1 5D 89 02 8A 98 31 7F C3 30
          :      52 1C C1 9E F1 3F DB 90 77 D7 A9 E8 A6 7E 45 7B
          :      AD DB 5D 4B F3 E5 B5 40 8D 50 25 20 CD C4 03 51
          :      B1
          :         }
  723  118:         [3] {
  725  116:          SEQUENCE {
  727   12:           SEQUENCE {
  729    3:            OBJECT IDENTIFIER
          :             basicConstraints (2 5 29 19)
  734    1:            BOOLEAN TRUE
  737    2:            OCTET STRING, encapsulates {
  739    0:             SEQUENCE {}
          :             }
          :            }
  741   11:           SEQUENCE {
  743    3:            OBJECT IDENTIFIER
          :             keyUsage (2 5 29 15)
  748    4:            OCTET STRING, encapsulates {
  750    2:             BIT STRING 7 unused bits
          :              '1'B (bit 0)
          :             }
          :            }
  754   29:           SEQUENCE {
  756    3:            OBJECT IDENTIFIER
          :             subjectKeyIdentifier (2 5 29 14)
  761   22:            OCTET STRING, encapsulates {
  763   20:             OCTET STRING
          :      23 1D 00 D1 FE ED 0B 39 10 11 98 18 CC 7D CD 70
          :      CD 8B D3 92
          :             }
          :            }
  785   31:           SEQUENCE {
```

```
  787    3:            OBJECT IDENTIFIER
         :             authorityKeyIdentifier (2 5 29 35)
  792   24:            OCTET STRING, encapsulates {
  794   22:             SEQUENCE {
  796   20:              [0]
         :      3E 98 BC B2 EF DC 34 1B BE 71 92 D2 A3 FA 6D BC
         :      F2 66 40 AA
         :               }
         :             }
         :            }
  818   23:           SEQUENCE {
  820    3:            OBJECT IDENTIFIER
         :             certificatePolicies (2 5 29 32)
  825   16:            OCTET STRING, encapsulates {
  827   14:             SEQUENCE {
  829   12:              SEQUENCE {
  831   10:               OBJECT IDENTIFIER
         :                testCertPolicy (2 16 840 1 101 3 2 1 48 48)
         :               }
         :              }
         :             }
         :            }
         :           }
         :          }
         :         }
  843   10:        SEQUENCE {
  845    8:         OBJECT IDENTIFIER
         :          ecdsaWithSHA384 (1 2 840 10045 4 3 3)
         :         }
  855  103:        BIT STRING, encapsulates {
  858  100:         SEQUENCE {
  860   48:          INTEGER
         :      6B BF 53 2A 5D EC 16 95 9D 48 C1 DF A5 2D 5F D9
         :      B9 66 63 E2 EF CC B9 D5 10 3C 5A 16 CE BF 42 90
         :      56 B7 18 B6 3E 2A 39 D8 8C 54 A0 5C A1 57 1E C8
  910   48:          INTEGER
         :      44 9E 94 F7 5D 38 F0 D0 1B DE 78 9C 1D CA C6 15
         :      FD 54 62 B8 5B 0E 5C AD 2B 8B 42 6B 91 C1 C4 3F
         :      EA 02 0C B8 FD E5 33 03 93 59 C1 56 8B 2B BF 2E
         :           }
         :          }
         :         }
         :        }
         :       }
         :      }
         :     }
         :    }
  960   10:  SEQUENCE {
  962    8:   OBJECT IDENTIFIER ecdsaWithSHA384 (1 2 840 10045 4 3 3)
         :    }
  972  103:  BIT STRING, encapsulates {
  975  100:   SEQUENCE {
  977   47:    INTEGER
         :     64 CD 1C F5 0B 59 C0 92 D8 3A 99 08 28 90 79 2C
         :     31 D8 93 CB 30 8D 78 56 2F 7B ED BC F2 AC 89 AA
         :     E6 9D 07 DC 17 A5 E9 F0 12 F0 19 D0 D3 45 8D
 1026   49:    INTEGER
         :     00 84 70 C0 95 C6 D1 B8 78 6A 75 6E A6 3A 00 DD
```

```
          :     E6 45 84 18 6A A0 8B 12 2E BC 41 1C 7D 56 E7 34
          :     21 32 10 4E 27 D2 64 3D F4 F1 D4 46 75 E8 1F 0A
          :     85
          :     }
          :    }
          :   }
```

The CA issues a key establishment certificate to Alice:

```
-----BEGIN CERTIFICATE-----
MIICJTCCAaygAwIBAgIUf3Sj/ANs4hR4XFlhTm+N8kxHqHowCgYIKoZIzj0EAwMw
NzELMAkGA1UEBhMCVVMxEzARBgNVBAoTCkV4YW1wbGUgQ0ExEzARBgNVBAMTCmNh
LmV4YW1wbGUwHhcNMjUwMTA5MTcwNTAwWhcNMjYwMTA5MTcwNTAwWjA8MQswCQYD
VQQGEwJVUzELMAkGA1UECBMCVkExEDAOBgNVBAcTB0hlcm5kb24xDjAMBgNVBAMT
BUFsaWNlMHQwDgYFK4EEAQwGBSuBBAAiA2IABAFHJBMf5yrWzkX3i4WoV7uXUuoZ
0QLxOb57lVgEJ8oALwwnJJAyPUaLEQRcv7VaO2Dbde+cmi4IOjFKkMJIyp/eS+nn
6dozfu7FrdliAPoM9jeZ7kRm/Jk+9JElMWX8FqN2MHQwDAYDVR0TAQH/BAIwADAL
BgNVHQ8EBAMCAwgwHQYDVR0OBBYEFAnLfJvnEUcvLXaPUDZMZlQ/zZ3WMB8GA1Ud
IwQYMBaAFD6YvLLv3DQbvnGS0qP6bbzyZkCqMBcGA1UdIAQQMA4wDAYKYIZIAWUD
AgEwMDAKBggqhkjOPQQDAwNnADBkAjARQ5LuV6yz8A5DZCll1S/gfxZ+QSJl/pKc
cTL6Sdr1IS18U/zY8VUJeB2H0nBamLwCMBRQ6sEWpNoeeR8Bonpoot/zYD2luQ1V
2jevmYsnBihKF0debgfhGvh8WIgBR69DZg==
-----END CERTIFICATE-----
```

Acknowledgements

Thanks to Sean Turner, Joe Mandel, Mike StJohns, Mike Ounsworth, John Gray, Carl Wallace, Corey Bonnell, Hani Ezzadeen, Deb Cooley, Mohamed Boucadair, and Bron Gondwana for their constructive comments.

Author's Address

Russ Housley
Vigil Security, LLC
Herndon, VA
United States of America
Email: housley@vigilsec.com