RFC 9216
S/MIME Example Keys and Certificates

Abstract
The S/MIME development community benefits from sharing samples of signed or encrypted data. This document facilitates such collaboration by defining a small set of X.509v3 certificates and keys for use when generating such samples.

Status of This Memo
This document is not an Internet Standards Track specification; it is published for informational purposes.

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). Not all documents approved by the IESG are candidates for any level of Internet Standard; see 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/rfc9216.

Copyright Notice
Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.
Table of Contents

1. Introduction
   1.1. Terminology
   1.2. Prior Work

2. Background
   2.1. Certificate Usage
   2.2. Certificate Expiration
   2.3. Certificate Revocation
   2.4. Using the CA in Test Suites
   2.5. Certificate Chains
   2.6. Passwords
   2.7. Secret Key Origins

3. Example RSA Certification Authority
   3.1. RSA Certification Authority Root Certificate
   3.2. RSA Certification Authority Secret Key
   3.3. RSA Certification Authority Cross-Signed Certificate

4. Alice's Sample Certificates
   4.1. Alice's Signature Verification End-Entity Certificate
   4.2. Alice's Signing Private Key Material
   4.3. Alice's Encryption End-Entity Certificate
   4.4. Alice's Decryption Private Key Material
   4.5. PKCS #12 Object for Alice

5. Bob's Sample
   5.1. Bob's Signature Verification End-Entity Certificate
   5.2. Bob's Signing Private Key Material
   5.3. Bob's Encryption End-Entity Certificate
   5.4. Bob's Decryption Private Key Material
   5.5. PKCS #12 Object for Bob
1. **Introduction**

The S/MIME ([RFC8551]) development community, in particular the email development community, benefits from sharing samples of signed and/or encrypted data. Often, the exact key material used does not matter because the properties being tested pertain to implementation correctness, completeness, or interoperability of the overall system. However, without access to the relevant secret key material, a sample is useless.
This document defines a small set of X.509v3 certificates ([RFC5280]) and secret keys for use when generating or operating on such samples.

An example RSA Certification Authority is supplied, and sample RSA certificates are provided for two "personas", Alice and Bob.

Additionally, an Ed25519 ([RFC8032]) Certification Authority is supplied, along with sample Ed25519 certificates for two more "personas", Carlos and Dana.

This document focuses narrowly on functional, well-formed identity and key material. It is a starting point that other documents can use to develop sample signed or encrypted messages, test vectors, or other artifacts for improved interoperability.

1.1. Terminology

"Certification Authority" (or "CA"): a party capable of issuing X.509 certificates

"End Entity" (or "EE"): a party that is capable of using X.509 certificates (and their corresponding secret key material)

"Mail User Agent" (or "MUA"): a program that generates or handles email messages ([RFC5322])

1.2. Prior Work

[RFC4134] contains some sample certificates as well as messages of various S/MIME formats. That older work has unacceptably old algorithm choices that may introduce failures when testing modern systems: in 2019, some tools explicitly marked 1024-bit RSA and 1024-bit DSS as weak.

This earlier document also does not use the now widely accepted Privacy-Enhanced Mail (PEM) encoding (see [RFC7468]) for the objects and instead embeds runnable Perl code to extract them from the document.

It also includes examples of messages and other structures that are greater in ambition than this document intends to be.

[RFC8410] includes an example X25519 certificate that is certified with Ed25519, but it appears to be self issued, and it is not directly useful in testing an S/MIME-capable MUA.

2. Background

2.1. Certificate Usage

These X.509 certificates ([RFC5280]) are designed for use with S/MIME protections ([RFC8551]) for email ([RFC5322]).

In particular, they should be usable with signed and encrypted messages as part of test suites and interoperability frameworks.
All end-entity and intermediate CA certificates are marked with Certificate Policies from [TEST-POLICY] indicating that they are intended only for use in testing environments. End-entity certificates are marked with policy 2.16.840.1.101.3.2.1.48.1 and intermediate CAs are marked with policy 2.16.840.1.101.3.2.1.48.2.

2.2. Certificate Expiration

The certificates included in this document expire in 2052. This should be sufficiently far in the future that they will be useful for a few decades. However, when testing tools in the far future (or when playing with clock-skew scenarios), care should be taken to consider the certificate validity window.

Due to this lengthy expiration window, these certificates will not be particularly useful to test or evaluate the interaction between certificate expiration and protected messages.

2.3. Certificate Revocation

Because these are expected to be used in test suites or examples, and we do not expect there to be online network services in these use cases, we do not expect these certificates to produce any revocation artifacts.

As a result, none of the certificates include either an Online Certificate Status Protocol (OCSP) indicator (see id-ad-ocsp as defined in the Authority Information Access X.509 extension in Section 4.2.2.1 of [RFC5280]) or a Certificate Revocation List (CRL) indicator (see the CRL Distribution Points X.509 extension as defined in Section 4.2.1.13 of [RFC5280]).

2.4. Using the CA in Test Suites

To use these end-entity certificates in a piece of software (for example, in a test suite or an interoperability matrix), most tools will need to accept either the example RSA CA (Section 3) or the example Ed25519 CA (Section 6) as a legitimate root authority.

Note that some tooling behaves differently for certificates validated by "locally installed root CAs" than for pre-installed "system-level" root CAs). For example, many common implementations of HTTP Public Key Pinning (HPKP) ([RFC7469]) only applied the designed protections when dealing with a certificate issued by a pre-installed "system-level" root CA and were disabled when dealing with a certificate issued by a "locally installed root CA".

To test some tooling specifically, it may be necessary to install the root CA as a "system-level" root CA.

2.5. Certificate Chains

In most real-world examples, X.509 certificates are deployed with a chain of more than one X.509 certificate. In particular, there is typically a long-lived root CA that users' software knows about upon installation, and the end-entity certificate is issued by an intermediate CA, which is in turn issued by the root CA.
The example end-entity certificates in this document can be used either with a simple two-link certificate chain (they are directly certified by their corresponding root CA) or in a three-link chain.

For example, Alice's encryption certificate (alice.encrypt.crt; see Section 4.3) can be validated by a peer that directly trusts the example RSA CA's root cert (ca.rsa.crt; see Section 3.1):

![Diagram](ca.rsa.crt alice.encrypt.crt)

*Figure 1: Validating Alice’s encryption certificate directly when the issuing CA is a trust anchor*

And it can also be validated by a peer that only directly trusts the example Ed25519 CA's root cert (ca.25519.crt; see Section 6.1) via an intermediate cross-signed CA cert (ca.rsa.cross.crt; see Section 3.3):

![Diagram](ca.25519.crt ca.rsa.cross.crt alice.encrypt.crt)

*Figure 2: Validating Alice’s cert from a different trust anchor via an intermediate cross-signed CA certificate*

By omitting the cross-signed CA certs, it should be possible to test a "transvalid" certificate (an end-entity certificate that is supplied without its intermediate certificate) in some configurations.

### 2.6. Passwords

Each secret key presented in this document is represented as a PEM-encoded PKCS #8 ([RFC5958]) object in cleartext form (it has no password).

As such, the secret key objects are not suitable for verifying interoperable password protection schemes.

However, the PKCS #12 ([RFC7292]) objects do have simple textual passwords, because tooling for dealing with passwordless PKCS #12 objects is underdeveloped at the time of this document.
2.7. Secret Key Origins

The secret RSA keys in this document are all deterministically derived using provable prime generation as found in [FIPS186-4] based on known seeds derived via SHA-256 ((SHA)) from simple strings. The validation parameters for these derivations are stored in the objects themselves as specified in [RFC8479].

The secret Ed25519 and X25519 keys in this document are all derived by hashing a simple string. The seeds and their derivation are included in the document for informational purposes and to allow recreation of the objects from appropriate tooling.

All RSA seeds used are 224 bits long (the first 224 bits of the SHA-256 digest of the origin string) and are represented in hexadecimal.

3. Example RSA Certification Authority

The example RSA Certification Authority has the following information:

Name: Sample LAMPS RSA Certification Authority

3.1. RSA Certification Authority Root Certificate

This certificate is used to verify certificates issued by the example RSA Certification Authority.

```
-----BEGIN CERTIFICATE-----
MIIDezCCAmOgAwIBAgITcBn0xb/zdaeCQlqp6yZUAGZUCDANBgkqhkiG9w0BAQ0FADBVMQ0wCwYDVQQKEwRJRVRGMRewDwYDVQQLEweMQU1QiUyBXRzExMCBGA1UEAxMoU2FtcGl1IEBTBVTIFJTQSBDZXJiaWZpY2fuO09uIEdhVcm1QeTeAgFw0xOTEExAjwU0MTghGA8MD0gMTYgNzAvMCAwHQYDVR0OBBYEwP6710Hv2Kw1A9y2HkU+L0wIwIBAgIARyvaB76BTeSgDf
-----END CERTIFICATE-----
```
3.2. RSA Certification Authority Secret Key

This secret key material is used by the example RSA Certification Authority to issue new certificates.

```
-----BEGIN PRIVATE KEY-----
MIIe+wIBADANBgkqhkiG9w0BAQEFAA SocketgSkAgEAAoIBAQC2GGPTFVNd1oL
s1079A8MzQ2G+LRLjbXzVoNo8StIbNyQ9VzFrGJHjUhRX/OmrQP3rCDB2SYFPWd
0CdC6z9qfJckxVdc1kz+K5v9Kmcl0IPyU1kJwWuMpaXa1Elez+zcU+vgyV83Uvn6w
Tn39Mcmym7uNPZlu6OnMYOCdMmz80Df89+hS04P3o8H23Vh7Se37
WafFx57kBiOuNjKEyPUk9tUkGxG9Jr95PPEHNh1mH5ybhBz1cmgyzRSEs
rKax9XeJK9U41WGaZ8zb44kCur/eiMoCk3YV68L3slyvcMGq1UDCAakx6FZFE7h
E9RN6L3bAgMBAAECgqEA3Etfhs7Dqgrl0+1K1kJbHhsR4soBhb4rrP6c18P0
678TuWbCj1DzoYkNTzaM3aXai4vuxVjB1sberKnhzTFQyxbX3ceSEIOCT3aoyq
5fipR8L6bA1yvg8R7TvNCAIApHn4a4pVsKBh8wq+h7m1UA0YGbiefU508/92qWjcz
+zcheyYXJS/iuqO/2F0ihE0GcxBmoc8D+++n7mKstz2kHAD4w1P2NgMvnpagmpBz
gobFNMmCzyZpD+S+PPTtQZ1XvdGF5SoGcf+Zz+jpWun1kqxDHE4UIZDa/HAaB0gRb
mZaVs0s9ZExqotqu2FPB7zF/1JKdRk4UJOUxS0QKBqDQJwonP5Rwv00syoCiW
zuFcYTMn/hI3R3viKuxr19CH6+mvuI85ooIHf6TiuozWhk+6+Kv7rcxsS5D4T54
2BrvR/x/51/M0zx8c81lWoZJlAsLz+xx8F4n6hxyVhV65bXN7ABojMh2dt8tPZM/R
VEFska4mm06yKuzyAfjziciCnCQKBqDnH9UYUIPq50PSviQKFJCBF9Bjzfd1d2
pIgoziw/JZJ3M3W1U0KW7Ux0T3xmn31X6xmWW4vX1/088ybObZWP0ed61Gm
I9D0i15qndLgDwv0L2FPUhZ5pqcQ09DE+cpJW4nNoudqTmCrjnhmNCGKgjD8z
/0KScvvywWKdDB8eajRuiEjDxj2UbzkX8lZJsX4KIs2G2IdHqSRCVly8Q0a
5WN3ULnIyB350HCP69wDFMXVym5rJoQjPvhGhluHkYV4V8fffxKv5k5sUXwVXJ7
Zvz+xM8B3qLvy+pkyWLyf7hKymdibzB+oTXwF7r4ueLJtaxngxn39a0GABKPr
r9R9PnoKhube/StrUNZFlvznwPvDv6l08T978t0NL372pUT95Kj8eN31DaMpoQpQc
BuqpsOqQLtInDysv2Kv9Rwl0ZwAcE9C8RmVj6+RdUSQ51Q8yjVLGanKsAHk
PTkv8CGY0V1CHL8t3pXf8wXaHxtuVCV8EyCvAoRBAIzelnVh0YTOjUadZ+0
vSOzAarg5k2CYPCTGF7+zi+j5rbrM7k+yX6DmWmjTOKVLDsVmxMBbA7GhL7TKy5
cEpBH1PVwxE1l8dQ+nUoeJeBpHo/cj8jC9R/aMjZl+iqU03M0DR+U9H9N1dKn
i75GRVLaEWA0tZ9E0md90jOsdwQYKkWYBAGSBCBiIAETeMKGCWGCSAFLwQc
AgOlpc3GlHY7U7WyawUlNRQotfwnYcZmotmTAti6Q==
-----END PRIVATE KEY-----
```

This secret key was generated using provable prime generation found in [FIPS186-4] using the seed a5c1b7847614ed61a6b05235142b44b7f09d8cca2d99302dd6e9. This seed is the first 224 bits of the SHA-256 (SHA) draft-lamps-sample-certs-keygen.ca.rsa.seed.

3.3. RSA Certification Authority Cross-Signed Certificate

If an email client only trusts the Ed25519 Certification Authority Root Certificate found in Section 6.1, they can use this intermediate CA certificate to verify any end-entity certificate issued by the example RSA Certification Authority.
4. Alice's Sample Certificates

Alice has the following information:

Name: Alice Lovelace

Email Address: alice@smime.example

4.1. Alice's Signature Verification End-Entity Certificate

This certificate is used for verification of signatures made by Alice.
4.2. Alice's Signing Private Key Material

This private key material is used by Alice to create signatures.
This secret key was generated using provable prime generation found in [FIPS186-4] using the seed 92c89d4333d3de8e13d4fe9bde0fe699fc142141dd65a45e5b436f05. This seed is the first 224 bits of the SHA-256 ([SHA]) digest of the string draft-lamps-sample-certs-keygen.alice.sign.seed.

4.3. Alice's Encryption End-Entity Certificate

This certificate is used to encrypt messages to Alice.
4.4. Alice's Decryption Private Key Material

This private key material is used by Alice to decrypt messages.
This secret key was generated using provable prime generation found in [FIPS186-4] using the seed 1cf74849f7445f466c4272251f5f96b77fa0698b3e98bf3f1ee8207bf. This seed is the first 224 bits of the SHA-256 ([SHA]) digest of the string draft-lamps-sample-certs-keygen.alice.encrypt.seed.

4.5. PKCS #12 Object for Alice

This PKCS #12 ([RFC7292]) object contains the same information as presented in Sections 3.3, 4.1, 4.2, 4.3, and 4.4.

It is locked with the simple five-letter password alice.
-----BEGIN PKCS12-----
MIIX+AIBAzCCF8AGCSqGSIb3DQEHAQCCqGSIb3DQEMAQAwDQKjMB977r7RL/VSxSBdLJv8Sv5n++/o3f7r5NkyBwuky33ySy3HvHz2c2TooyFedvR8x2z8MeVaM7pW4z4og/IVR0TlzCMjWooGoEO0Rim6y2G+iRZ3ePBUq0+8SENYW+jlWov9abdfqgj9jibQkJ/Hrdje2Tcd16a9sT1FyVxWBudWlZDwcvQqwc1WMw1e6T9fEpzldzr5+n+ZfSh0wRAwRGR0jXsX9AUesm9fDxqozV8s3eqgqFY6aJcZOxe171IECAiqKh8z1lKCrywNvAkwXeFw0j1WmnW41tDaqcC6GeupuINiqqdf94YAOHxRL1l1kU4ulLm5Jkn9S7D67Wa14f9p85k6QpaDzQheu6BBq+dvd4d4sQ3DmwqVDY07UjyjIEPv39E9fEm641CR0m1CvUyxn3a3kJhjBKTbgYsKtueP9744DRIPARmS2eBoy3Cgzf7c20qEIvCglRatN6eSjLBf8jBOWp5ZGc08AwTM1pK3RrDziheA8DBBB=KT4J2ZBO6upr1jC6Yb6nCXa5F4e4b57Cdb80+zuJuubBX9f4yG/j0CSF59w9c2c6Qb14EfKt6ai9j1PvojgLjCfXeur7wvV3e9yj1Av9Pfjwzq/RENALFW41D30WIBRc2mshhJ8s83FNPcvIY3EKnZwM14rZFAjbtLoCck5rcfFSSxvucAvCSX95605xExcdOysnupLH77u+CWWXj3C4141SKZTuBwc1a0B6Y6RdvojUMKzdSxzyyxJnJG5QhmFQryTALcWgCsp/rAf5SxPh2p2+9Qul8yb1IIZwKKNKRQKL+YLCyvYTh1bjbfurfLVyzzvy9CLcX9yLziMfL0Mu6Aw6IWX2BjhtGu6VThRmHufoQsNgq/ZrNCTyAe77wGwSUc3ReszSzdA+50Te657e/vaN4o4A3h8+535SYQ6pnPo/tAa0Dh7Z77A9N+mkz1Wc5U3d1s48tyUImJnAgT3tRhZf2gfojJ7IdcYMDp011+Ud5L9GhXbNd3h1F3rDyn6Y5B1Lx4Z0mLejI+3P1rB+9Ykgf+o+hxfttfx1poXChhteA4bR6gBf5wGnsFw6Sg6sg89UDTAYOpLaal1l7ZmeyZyJ9jrssv36kr1TvArV9qCbxC6bsuybuxmFp/rhVdxhIydmVtdozY0WFDtvjgJUNEaRNc3mVMDA1lWDW0mbQc0ILJs8vW8fJk8u2YAXBhRYow9li3Qzq3dzmFkJkwauMrnMZvHophmEN6rwo3287gnbyyu1/c/di+s6Vyx/bkrVx/lyJwBHnhnkgkhigovohGtt+yAllcVUhrNqE0Q5u39F0JU98d6Wpvm01tq6NpNYUhbcOFGdHFr4swvrgR3Bkgkhkl9w0BBwagaRMOIEIA1BADCCBF0GScqGSIb3DEQHAEcTBoqkhk1L9w0BDBAEDMA4ECpOEFHEQG9dAgOUsCOBAocRhyNf47ZkttJ1yVWQZNgPlzn6vz2poKg76FeMgdS2w1kWTKxunfo/jpG6r6bXTA7uKhJDSxHrzrFxFkJY14H2oaaNihwqtcphcLoNBXc0ufuHv+loP99PlmTrwu3M01sw7Hf65xM02w5fbrp2oLohqJ1RZ89FBupAahFqg91polXPez7GUeLYOnYH98R9iu9bSe1bP14s+AoxzeD4qYiU6Yi67/47aRt3d34u3E0RDNUKosSvts1RSKKh/WgRYUowy7KnDwyDBItmosMY0rPwEEt5rXtwBjkdwo13LpbDnDvzrzy+eeYbOJ9F1ML1xPvfoij9ug9bD2gy08K6f6C3zFD5fjQinhNM3CTc3rcEisj11jn9gLe0LB9c1qyE4PFwFMnH8wqgDFeX96umlUpymYM36H6hjcbnSc657eIzTrCqrgjsHqswmg94waXwehdflFxLzsdxzc+Jc8cftTuUEOUX3tX4110u7K8uAuQTSK/AxwUj+MbVh8zte4FvVw4u1Z8iYglh0dV3IdO0xAzkFSDKxRh2Z/8acxarfYFIkkrcr2SiHMhmoxJH7WENTNislw5R5runj6158v5tvHCQFlNFQfGBB7Tv9AdPY7ka9AQDw0PCOfxg6b01F1Q/BigzV0J5PScrWv71a88BQ8zrH7257113783zH+5z1tgbrzrSfpmh61+rj7Q7LQFVvXnFg6L6Q/GQOQCTaMYZ1D9S26j73qj8uk1LncL9FsaApQ5sWspBxz039F6GCMgZeZKfCqroQvWyhXbtzn90x/pimmGQl5Cyx+wythbBM+11xJcbof1j95wmsCWNPN1MAVSholpZhs9M69rxgBx11HjggD90BDPCLE80h/GNOoX03LxRFcIDEHT8TLYz2Z59hy3t63pm688ozaFahgdjbj1Fkc79nBbc14NLKgSmJ+7gk7t0Ezn01yf5f9J5c028B8nRrOfFDRUWZ2hXbbOx0C4E3E0Zt13aHJ0XRB0ZLC9L2J6eSn1BZyGyk+P14TisXZL2UoP+yd4DDeFbyamY8bd1hFshndqr94Y9E3oFy08ubU2bGeQgT9np9uYmeridsCk hegmmqmgVr/52Xv/Xk9X44YBjX4/3ID0/yVJUoDH8b04ogF4frlkz0alr9zJFugPoU8M0Nysal9y7r7i11juV0MIIDZ0WkJcNuACl0WDBCDAC1QACAQwggMBgkq1xNl9bwBBwEHAYKOkZhcNAwQBAzAO8idIbqXvFwagIFCfKAcMaMGsTrzu4/s12Jgqnu3AL+P6s99uxy1zBmNc1w+HVRHJe0H8afUaybzdSRBAaCh1+8GueU8zy7iWmhn1PHKh1ZbInyFJm1fkkPa34Rks/RxKJeU1MLy1jk/Fmq/ripgcQ56S6S5xv6j9Hb9gtaoGWP8b1b9dfvbNhA8b08r+5iwH4u5b6pcWdRwJleGicsY0Hhx4KMMVF6JrFVCTZevORVomxsZzZqgrbjuv4CZQoXWHeRH13aET5e53X8RiR64EI4Eia6y8U10vO2ZTBQkWnEKM0A5GwSX3mH/kLiya3gwGvdq1nxcC17v1STN1HFPye4bBGk9g4CsZ6MKwjcqowC2pwPm/Tqaz

Gillmor Informational Page 14

RFC 9216 April 2022
5. Bob's Sample

Bob has the following information:

Name: Bob Babbage
Email Address: bob@smime.example

5.1. Bob's Signature Verification End-Entity Certificate

This certificate is used for verification of signatures made by Bob.

-----BEGIN CERTIFICATE-----
MIIDyjCCArKgAwIBAgITaqOkD33fBy/kGaVsmPv8LghbwzANBkgqtkhi9gw09BA0F
ADBVMQowcwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzExMC8GA1UEAxMo
U2FtcgxlIExBTQBTIFJTQSBXZ3ZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAgFw0xOTEx
MjAwNjU0MThaGA8yMDUyMDkyNzA2NTQxOFowODENMAsGA1UEChMESUVURjERMA8G
A1UECxMITEFNUFMgV0cxFDASBgNVBAMTC0JvYiBCYWJiYWdlMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5nAF0glRof9NjBKke6g+7RLrOgRfwQjcH+2z
m0Af67FJRNrEwTuOutlWamUA3p9+wb7XqizVHOQhVesjwp8PJo8AdmBar84d2t
tey0v9txacJuNe7SjfrwShB6vAm7S8CDG3+Eap09fzon2pWwaERE6twWtHi1QT
51pu0tiQqsuuk8LBdguUM2kiUsaXffK8kzJjQgUa1R5/3Kf9r+b6vKcdxTZYL
Zxt6+a/3kAca319my9pubtHFB5P5+sboROSkm1OB1g5Low8eF970tGcozz
Ji1QUCR14NAu51byfKEV2YStxwzdtoEJJ2frUIK+6Ynw1B3QDAABo4GtMIgQ
MawGAI1deWeB/wQCMAMAwfYDVR0GRbAwDjAMBpgphkgqBZOMCATAMBwGA1UDQEQ
BOBEWJyVbzbWlzTSZ1eGftc6x1MBMGA1UdJQQMA0GCcsGAUFBwEME4A1AgAD
Dw/e/bw/ew/igw/iGk3QGoH4EfgQOFU8EWe9Cn73aQLizbw18krWk5QwHwDVR0j
BBqcoFauKTC0fAcXDKfcsH1NnhpHGh29fKwDQYOJkoIzhcNAQENBQADgEBAG7e
QY6P/7wXc5vChJlOtto2zOlmy+VrCSTGWOYDxm/1wUsy31pE3dtAdvervRPs8
uN7xxyfK6XZBzShA/BtKq5YifVxdpu0XWmQ0C0Mpc1Pnk2mHil+gPmFmUnVxwd
6gKcHEd5p+bhuDfIHYf9yGse08nvi+7/HwBipN+nA/PfPs+n+u4ILK6qoDo/ir
kuywlocFflc5yE5kaaeJ2/a+8/HTzNnk4jJ/B+Gby6x1UszPlQKE+Es10Xut/Y
UWY6vpkKQap/R8D7Qp371Mqs2+Zx4f4GheKzU3XRlPcApYjiWyiU1DzpqgSJM
OIP/HtxDfScHb9+Qic==
-----END CERTIFICATE-----
5.2. Bob's Signing Private Key Material

This private key material is used by Bob to create signatures.

-----BEGIN PRIVATE KEY-----
MIIE+wIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDmcAXSCVGh/02M
Eqr7qD7tEus6BF/CBNwft7bObQ8/rsUEI2sTB04662VZqZQen37ByteQL/wc5CFV
6yPCCnw8mmjw2bixvqzh3a217LUS45Fm4m17IInM+vBKEHo28CbtLwMbf4RmqQ
71/oFa1bBpERDq38a8eLVBPnU925Z2JDwqy4mTwsEOBQxmUpSxsdB8XwYMrMuA5o
tXn/cp+v35wpWIO7f1qnG3r5rf9CRoLCj2bbKAsu8C0Khk/n6zuxhE51gUB
H4WCUwjDx4X1Ls5wY6ihkmIIRAHJH1g1P71sj80R1XZkh1F3B032g0kzn29FREgr
7xifCUHdAgMBAAECggEAcBQf1TtieZ+0/aNdU149NK0qx97GLTBJ1guQEDDBBFK
2lu4PbHg9AqAdAULpH1PE+eq6JAnGzwFH8X1Ms2AK1RlZZYsPOQIOJ4n1h6c9u1EiN
Ykvck4QH0vqCtWjYjy5b5y9WPeLH6QynJ6FLBosxhURSWyFTuwqt10HeSUHh
d3N5BmbfIRBNj4IAI9zz+i5xL0m33kMKai/Aj3s10AJsZ5VAVhYbc8cSt1Xevbs
i14p9s6GSwGc1bvy+1y9WCQgtb5GDotvChMvmZS/03NeDcxcL/LzOqCHvNg1Zd7
tg61eIKjICYK+D7xsd7Y6s0w75Hajvnh1jJ0sSA+wKBQGQdxv8j2P261VRGyFYa
cnU3M7wqjg£fPH09SK6e9c8CGqORh2uwWiTawu88x8BGFyZ+xnWq75CNs1tas
3m9r14a4R94+5UL8+oOLC26gMDfzAtD1Qk/h919YLy89tonQEBuCFZjdPthetb
vg2W+nNesEvQGClzhX0AyGMSwBKBQ0DxB0kYsdQGBa/1HY1EY5zfYeBu1Yv21T
VGRTotKfCRAWt0GPY9RbKyEvKblhjgXzS9xGqKwP71z9lny+zDGbzk8e1B/g
1S7GFGX50T0GI5faFWTyxt4m9npu2e2B1t/26uuyU8DXXEChF/BQ/hwqJHjKTTY
R13Ara5flrWkBQgQDQQvYtjIyD2q8naY2D8cm03vHtzc21tQzcUD8Z4vYSps1hbo
KN/48qJMrV3tjq+o+XsasYKFsE/4PirolxtVNNKbQM6ekfettw0p1yGP8340Wk
97Hv0Ji/g/x6m0Qv1yBsm=+q9Tkt+rvmp1RGlMe6BQgSYy4rS0UJvCnWkBQGC1
B4FvW0ytVQhWAwHFuQj3av/k+T+u/5kg6vGKFJ1Nw1x8ZW5knvbJCspAlqTnFyZK
ss51mo1IzEtbKtqK6tq8tqAv9v9AYWQKrgzXUs0sUwCZ+x3aWEF87lPNE
1QKxifZaquZ3T2kVysOZz8mngq9x7u8H3gULY26HOKBpGOC/COj1Y25Nw5Z5Fdii
PsQMvH7+yJycJaLhHSc/c7P0qOQRMDnevAk/eYJbKbU7qsFJ4IS1/IG0CfmVU/m
ax5bfbfYztoB/0xwaLkIEStVwAWrkSKRdTrnZtAOreeJKeY4RNP6vmpgojBiA1
Tg8Mup80x8QF4d28rtUenyHxosDswOQYKHkwYBBAGCSBIATErMCKGWCXCSAFwAoQc
AgC9K+qvYVHPZoBqw4A4GI/fKzhRJXm88E00Pbg==
-----END PRIVATE KEY-----

This secret key was generated using provable prime generation found in [FIPS186-4] using the seed f4afaacb5b5473f360e632e00188fe4173ae15c99bcf843ab8b8fe6. This seed is the first 224 bytes of the SHA-256 digest of the string draft-lamps-sample-certs-keygen.bob.sign.seed.

5.3. Bob's Encryption End-Entity Certificate

This certificate is used to encrypt messages to Bob.
5.4. Bob's Decryption Private Key Material

This private key material is used by Bob to decrypt messages.
This secret key was generated using provable prime generation found in [FIPS186-4] using the seed 98c8998652958929e889e3419f3bf6edefe0aca15da3060d3ef8ae8. This seed is the first 224 bits of the SHA-256 digest of the string draft-lamps-sample-certs-keygen.bob.encrypt.seed.

5.5. PKCS #12 Object for Bob

This PKCS #12 ([RFC7292]) object contains the same information as presented in Sections 3.3, 5.1, 5.2, 5.3, and 5.4.

It is locked with the simple three-letter password bob.
6. Example Ed25519 Certification Authority

The example Ed25519 Certification Authority has the following information:

Name: Sample LAMPS Ed25519 Certification Authority

6.1. Ed25519 Certification Authority Root Certificate

This certificate is used to verify certificates issued by the example Ed25519 Certification Authority.

-----BEGIN CERTIFICATE-----
MIIBtzCCAWmgAwIBAgITH59R65FuWGNFHoyc0N3iWesrXzAFBgMrZXAwWTENMAsg
A1UEChMESUVURjERMA8GA1UECxMITEFNUFMgV0cxNTAzBgNVBAMTLFNhbXBsZSBM
QU1QUyBFZDI1NTE5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTIwMTIxNTIx
MzUyMTMwODAwNzczMFoYDzIwMTIxMjE1MjEzNTQ0WjBZMQ0wCwYDVQQKEwRJRVRGMREw
DQYDVQQ_KMAwYzQxHwYDVQQDIEVITTVBMGCCsGAQUFBzAChGZGMiYwczAICHegMA0GCSqG
SIb3DQJEIEhNgYwBAGgUHAwIBAgIDAQAB
-----END CERTIFICATE-----

6.2. Ed25519 Certification Authority Secret Key

This secret key material is used by the example Ed25519 Certification Authority to issue new certificates.

-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VwbCBDIEIAAt889xRDvxNT8ak53T7tzKU6nCAwCj4v9P0y7o9sL
-----END PRIVATE KEY-----
This secret key is the SHA-256 ([SHA]) digest of the ASCII string `draft-lamps-sample-certs-keygen.ca.25519.seed`.

### 6.3. Ed25519 Certification Authority Cross-Signed Certificate

If an email client only trusts the RSA Certification Authority Root Certificate found in Section 3.1, they can use this intermediate CA certificate to verify any end-entity certificate issued by the example Ed25519 Certification Authority.

```
-----BEGIN CERTIFICATE-----
MIICvzCCAaegAwIBAgITR49T5oAgYhF5+eBYQ3ZBZIMuujANBgkqhkiG9w0BAQsFADBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzExMC8GA1UEAxMoU2FtcGxlIExbBTBTIEh0aT1BTI1NXIjBkZWIzNzBCMA0GCS0GCS0GA1UdDwEB/wQQEAwIBBjAdBgNVHQ4EFgQUa6KVfboUHm+QtBNEHpNCg55grJLUwhYDVR0jBBgwFoAuKTC0AcKaDXKf0XSh1NhpnHGu29FkwDQYJKoZIhvcNAQELBQAADgEBGAV0x00F3gY1KixMeztiiixxJDbmRat1pcipD151n81B0GhsT4fNZVJviL0Q9Qa/WTMnl+qcAk2itqZCNIEZeGk1UjXBaz5tK0AFv99LEcstcuIbnJq35danQK4/pqCq4hPffx+nbc1bsVylrITwIHOWpNH7Z7eC
VCK03DFE3Q4t4w9mvQ9yuMse333nisBGXog/XZvm2JRY0ikt8xksQqD9uYm7MoMeHqOs30t7EaoPj54xYhv42run6TLUye64D94SNjB/q/wjL96bsVIKGrRn10T1ybC4F5HDB0hQzgP15Dlb1rg+vskNMSk5nuD+6z1VsugioW0+k=
-----END CERTIFICATE-----
```

### 7. Carlos's Sample Certificates

Carlos has the following information:

Name: Carlos Turing

Email Address: carlos@smime.example

#### 7.1. Carlos's Signature Verification End-Entity Certificate

This certificate is used for verification of signatures made by Carlos.
7.2. Carlos's Signing Private Key Material

This private key material is used by Carlos to create signatures.

-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VWBIEIEIlhvcNLJ4fLx+EpmI3yee3Cjrr4JmNOIVYhZPM4M9N1IHY
-----END PRIVATE KEY-----

This secret key is the SHA-256 ([SHA]) digest of the ASCII string draft-lamps-sample-certs-keygen.carlos.sign.25519.seed.

7.3. Carlos's Encryption End-Entity Certificate

This certificate is used to encrypt messages to Carlos. It contains an SMIMECapabilities extension to indicate that Carlos's MUA expects Elliptic Curve Diffie-Hellman (ECDH) with the HMAC-based Key Derivation Function (HKDF) using SHA-256, and that it uses the AES-128 key wrap algorithm, as indicated in [RFC8418].

-----BEGIN CERTIFICATE-----
MIICDCCAcagAwIBAgITfz0B0vb+bIOMAT79aCh3arViNvhDAFBgMrZXAwWTENMAsG
A1UEchMESUVurmjrERMA8GA1UEcMITEFNUFNgV8cxNTAzBgNVBAMTLEFQMDjFpMT
QUlQfBFZDI1NTE5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTIwMTIxNTIx
Mz0NzF4OjTzIwETM1I3NhIveT3mMzI6M0VwbDAwF0QYhZPM4M9N1IHY
-----END CERTIFICATE-----
7.4. Carlos's Decryption Private Key Material

This private key material is used by Carlos to decrypt messages.

```
-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VuBCIEIIH5782H/otrhLy9Dtvzt79ffsvpcVXgdUczTdUvSQsK
-----END PRIVATE KEY-----
```

This secret key is the SHA-256 ([SHA]) digest of the ASCII string `draft-lamps-sample-certs-keygen.carlos.encrypt.25519.seed`.

7.5. PKCS #12 Object for Carlos

This PKCS #12 ([RFC7292]) object contains the same information as presented in Sections 6.3, 7.1, 7.2, 7.3, and 7.4.

It is locked with the simple five-letter password `carlos`. 
8. Dana's Sample Certificates

Dana has the following information:

**Name:** Dana Hopper

**Email Address:** dna@smime.example

8.1. Dana's Signature Verification End-Entity Certificate

This certificate is used for verification of signatures made by Dana.

```
-----BEGIN CERTIFICATE-----
MIICAzCCAbWgAwIBAgITaWZI+hVtn8pQZviAmPmBXzWfnjAFBgMrZXAwWTENMAsg
A1UEchMESUVRjERMA8GA1UECzMEbTQwDQYJKoZIhvcNAQcDBgkMEQyBQRkJR
-----END CERTIFICATE-----
```

8.2. Dana's Signing Private Key Material

This private key material is used by Dana to create signatures.

```
-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VwCBEIEINZBGfQmh2A+uNIzZMbvzyT0ltwVEt13usjnuA4N
-----END PRIVATE KEY-----
```

This secret key is the SHA-256 ([SHA]) digest of the ASCII string `draft-lamps-sample-certs-keygen.dana.sign.25519.seed`.

8.3. Dana's Encryption End-Entity Certificate

This certificate is used to encrypt messages to Dana. It contains an SMIMECapabilities extension to indicate that Dana's MUA expects ECDH with HKDF using SHA-256, and that it uses the AES-128 key wrap algorithm, as indicated in [RFC8418].
8.4. Dana's Decryption Private Key Material

This private key material is used by Dana to decrypt messages.

-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VUBIEIgIzYt8L71Y480E4gs/smQ4weDhRNMLYHG21StivFpz3
-----END PRIVATE KEY-----

This seed is the SHA-256 ([SHA]) digest of the ASCII string draft-lamps-sample-certs-keygen.dana.encrypt.25519.seed.

8.5. PKCS #12 Object for Dana

This PKCS #12 ([RFC7292]) object contains the same information as presented in Sections 6.3, 8.1, 8.2, 8.3, and 8.4.

It is locked with the simple four-letter password dana.
9. Security Considerations

The keys presented in this document should be considered compromised and insecure, because the secret key material is published and therefore not secret.

Any application that maintains a deny list of invalid key material should include these keys in its list.

10. IANA Considerations

This document has no IANA actions.

11. References

11.1. Normative References


---

S/MIME
April 2022

Gillmor
Informational
Page 30
11.2. Informative References


Acknowledgements

This document was inspired by similar work in the OpenPGP space by Bjarni Rúnar Einarsson and juga; see [OPENPGP-SAMPLES].

Eric Rescorla helped spot issues with certificate formats.

Sean Turner pointed to [RFC4134] as prior work.

Deb Cooley suggested that Alice and Bob should have separate certificates for signing and encryption.
Wolfgang Hommel helped to build reproducible encrypted PKCS #12 objects.

Carsten Bormann got the XML sourcecode markup working for this document.

David A. Cooper identified problems with the certificates and suggested corrections.

Lijun Liao helped get the terminology right.

Stewart Bryant and Roman Danyliw provided editorial suggestions.

**Author's Address**

**Daniel Kahn Gillmor (EDITOR)**
American Civil Liberties Union
125 Broad St.
New York, NY 10004
United States of America
Email: dkg@fifthhorseman.net