

---

|            |                                        |                                    |                                |                                            |                           |
|------------|----------------------------------------|------------------------------------|--------------------------------|--------------------------------------------|---------------------------|
| Stream:    | Internet Engineering Task Force (IETF) |                                    |                                |                                            |                           |
| RFC:       | <a href="#">9909</a>                   |                                    |                                |                                            |                           |
| Category:  | Standards Track                        |                                    |                                |                                            |                           |
| Published: | December 2025                          |                                    |                                |                                            |                           |
| ISSN:      | 2070-1721                              |                                    |                                |                                            |                           |
| Authors:   | K. Bashiri<br><i>BSI</i>               | S. Fluhrer<br><i>Cisco Systems</i> | S. Gazdag<br><i>genua GmbH</i> | D. Van Geest<br><i>CryptoNext Security</i> | S. Kousidis<br><i>BSI</i> |

## RFC 9909

# Internet X.509 Public Key Infrastructure: Algorithm Identifiers for SLH-DSA

---

## Abstract

Digital signatures are used within the X.509 Public Key Infrastructure such as X.509 certificates, Certificate Revocation Lists (CRLs), and to sign messages. This document specifies the conventions for using the Stateless Hash-Based Digital Signature Algorithm (SLH-DSA) in the X.509 Public Key Infrastructure. The conventions for the associated signatures, subject public keys, and private keys are also specified.

## 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/rfc9909>.

## Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

|                                |    |
|--------------------------------|----|
| 1. Introduction                | 2  |
| 1.1. Notation                  | 3  |
| 2. Conventions                 | 3  |
| 3. Algorithm Identifiers       | 4  |
| 4. SLH-DSA Signatures          | 6  |
| 5. Subject Public Key Fields   | 8  |
| 6. Key Usage Bits              | 12 |
| 7. Private Key Format          | 12 |
| 8. Operational Considerations  | 13 |
| 9. Security Considerations     | 14 |
| 10. IANA Considerations        | 15 |
| 11. References                 | 15 |
| 11.1. Normative References     | 15 |
| 11.2. Informative References   | 16 |
| Appendix A. ASN.1 Module       | 17 |
| Appendix B. Security Strengths | 23 |
| Appendix C. Examples           | 24 |
| C.1. Example Public Key        | 24 |
| C.2. Example Private Key       | 25 |
| C.3. Example Certificate       | 25 |
| Acknowledgments                | 38 |
| Authors' Addresses             | 38 |

## 1. Introduction

The Stateless Hash-Based Digital Signature Algorithm (SLH-DSA) is a quantum-resistant digital signature scheme standardized in [FIPS205] by the US National Institute of Standards and Technology (NIST) Post-Quantum Cryptography (PQC) project [NIST-PQC]. Prior to

standardization, the algorithm was known as SPHINCS+. SLH-DSA and SPHINCS+ are not compatible. This document defines the ASN.1 Object Identifiers (OIDs) and conventions for the encoding of SLH-DSA digital signatures, public keys, and private keys in the X.509 Public Key Infrastructure.

SLH-DSA offers three security levels. The parameters for each of the security levels were chosen to be at least as secure as a generic block cipher of 128, 192, or 256 bits. There are small (s) and fast (f) versions of the algorithm, and there is also the option to use the SHA-2 algorithm family [FIPS180] or SHAKE256 [FIPS202] as internal functions. While the fast versions are optimized for key generation and signing speed, they are actually slower at verification than the SLH-DSA small parameter sets. The small versions are optimized for signature size; see Table 1. As an example, id-slh-dsa-shake-256s represents the 256-bit security level, the small version of the algorithm, and the use of SHAKE256.

NIST [CSOR] has assigned separate algorithm identifiers for SLH-DSA for each combination of these security levels: fast vs. small, SHA-2 vs. SHAKE256, and pure mode vs. pre-hash mode.

SLH-DSA signature operations include an optional context string (ctx) as input, defined in Section 10.2 of [FIPS205]. The context string has a maximum length of 255 bytes. By default, the context string is the empty string. This document only specifies the use of the empty context string for use in the X.509 Public Key Infrastructure.

SLH-DSA offers two signature modes: pure mode, where the entire content is signed directly, and pre-hash mode, where a digest of the content is signed. This document uses the term SLH-DSA to refer to the algorithm in general. When a pure or pre-hash mode needs to be differentiated, the terms Pure SLH-DSA and HashSLH-DSA are used. This document specifies the use of both Pure SLH-DSA and HashSLH-DSA in Public Key Infrastructure X.509 (PKIX) certificates and Certificate Revocation Lists (CRLs).

## 1.1. Notation

The following notation is used in this document:

a || b: Concatenation of a and b.

id-slh-dsa-\*: A shorthand to refer to all 12 OIDs used to specify the different parameter combinations for Pure SLH-DSA.

id-hash-slh-dsa-\*: A shorthand to refer to all 12 OIDs used to specify the different parameter combinations for HashSLH-DSA.

## 2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 3. Algorithm Identifiers

The AlgorithmIdentifier type is defined in [\[RFC5912\]](#) as follows:

```
AlgorithmIdentifier{ALGORITHM-TYPE, ALGORITHM-TYPE:AlgorithmSet} ::=
    SEQUENCE {
        algorithm    ALGORITHM-TYPE.&id({AlgorithmSet}),
        parameters   ALGORITHM-TYPE.
                     &Params({AlgorithmSet}{@algorithm}) OPTIONAL
    }
```

NOTE: The above syntax is from [\[RFC5912\]](#) and is compatible with the 2021 ASN.1 syntax [\[X680\]](#). See [\[RFC5280\]](#) for the 1988 ASN.1 syntax.

The fields in AlgorithmIdentifier have the following meanings:

- algorithm identifies the cryptographic algorithm with an object identifier.
- parameters, which are optional, are the associated parameters for the algorithm identifier in the algorithm field.

The object identifiers for SLH-DSA are defined in the NIST Computer Security Objects Register [\[CSOR\]](#) and are reproduced here for convenience. The same algorithm identifiers are used for identifying a public key, a private key, and a signature.

The Pure SLH-DSA OIDs are defined in the ASN.1 module in [\[RFC9814\]](#) and reproduced here for convenience:

```
nistAlgorithms OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
  country(16) us(840) organization(1) gov(101) csor(3) 4 }

sigAlgs OBJECT IDENTIFIER ::= { nistAlgorithms 3 }

id-slh-dsa-sha2-128s OBJECT IDENTIFIER ::= { sigAlgs 20 }
id-slh-dsa-sha2-128f OBJECT IDENTIFIER ::= { sigAlgs 21 }
id-slh-dsa-sha2-192s OBJECT IDENTIFIER ::= { sigAlgs 22 }
id-slh-dsa-sha2-192f OBJECT IDENTIFIER ::= { sigAlgs 23 }
id-slh-dsa-sha2-256s OBJECT IDENTIFIER ::= { sigAlgs 24 }
id-slh-dsa-sha2-256f OBJECT IDENTIFIER ::= { sigAlgs 25 }
id-slh-dsa-shake-128s OBJECT IDENTIFIER ::= { sigAlgs 26 }
id-slh-dsa-shake-128f OBJECT IDENTIFIER ::= { sigAlgs 27 }
id-slh-dsa-shake-192s OBJECT IDENTIFIER ::= { sigAlgs 28 }
id-slh-dsa-shake-192f OBJECT IDENTIFIER ::= { sigAlgs 29 }
id-slh-dsa-shake-256s OBJECT IDENTIFIER ::= { sigAlgs 30 }
id-slh-dsa-shake-256f OBJECT IDENTIFIER ::= { sigAlgs 31 }
```

The HashSLH-DSA OIDs are:

```
nistAlgorithms OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
  country(16) us(840) organization(1) gov(101) csor(3) 4 }

sigAlgs OBJECT IDENTIFIER ::= { nistAlgorithms 3 }

id-hash-slh-dsa-sha2-128s-with-sha256 OBJECT IDENTIFIER ::= {
  sigAlgs 35 }

id-hash-slh-dsa-sha2-128f-with-sha256 OBJECT IDENTIFIER ::= {
  sigAlgs 36 }

id-hash-slh-dsa-sha2-192s-with-sha512 OBJECT IDENTIFIER ::= {
  sigAlgs 37 }

id-hash-slh-dsa-sha2-192f-with-sha512 OBJECT IDENTIFIER ::= {
  sigAlgs 38 }

id-hash-slh-dsa-sha2-256s-with-sha512 OBJECT IDENTIFIER ::= {
  sigAlgs 39 }

id-hash-slh-dsa-sha2-256f-with-sha512 OBJECT IDENTIFIER ::= {
  sigAlgs 40 }

id-hash-slh-dsa-shake-128s-with-shake128 OBJECT IDENTIFIER ::= {
  sigAlgs 41 }

id-hash-slh-dsa-shake-128f-with-shake128 OBJECT IDENTIFIER ::= {
  sigAlgs 42 }

id-hash-slh-dsa-shake-192s-with-shake256 OBJECT IDENTIFIER ::= {
  sigAlgs 43 }

id-hash-slh-dsa-shake-192f-with-shake256 OBJECT IDENTIFIER ::= {
  sigAlgs 44 }

id-hash-slh-dsa-shake-256s-with-shake256 OBJECT IDENTIFIER ::= {
  sigAlgs 45 }

id-hash-slh-dsa-shake-256f-with-shake256 OBJECT IDENTIFIER ::= {
  sigAlgs 46 }
```

The contents of the parameters component for each algorithm **MUST** be absent.

## 4. SLH-DSA Signatures

SLH-DSA is a digital signature scheme built upon hash functions. The security of SLH-DSA relies on the security properties of the underlying hash functions, such as the presumed difficulty of finding preimages.

Signatures can be placed in a number of different ASN.1 structures. The top-level structure for a certificate is given below as being illustrative of how signatures are frequently encoded with an algorithm identifier and a location for the signature.

```
Certificate ::= SIGNED{ TBSCertificate }

SIGNED{ToBeSigned} ::= SEQUENCE {
    toBeSigned          ToBeSigned,
    algorithmIdentifier SEQUENCE {
        algorithm      SIGNATURE-ALGORITHM.
                        &id({SignatureAlgorithms}),
        parameters     SIGNATURE-ALGORITHM.
                        &Params({SignatureAlgorithms}
                                {@algorithmIdentifier.algorithm})
                        OPTIONAL
    },
    signature BIT STRING (CONTAINING SIGNATURE-ALGORITHM.&Value(
                                {SignatureAlgorithms}
                                {@algorithmIdentifier.algorithm}))
}
```

NOTE: The above syntax is from [\[RFC5912\]](#) and is compatible with the 2021 ASN.1 syntax [\[X680\]](#). See [\[RFC5280\]](#) for the 1988 ASN.1 syntax.

The same algorithm identifiers are used for signatures as are used for public keys. When used to identify signature algorithms, the parameters **MUST** be absent.

The data to be signed is prepared for SLH-DSA. Then, a private key operation is performed to generate the raw signature value.

When signing data using the Pure SLH-DSA signature algorithm, Algorithm 22 (slh\_sign) from Section 10.2.1 of [\[FIPS205\]](#) is used. When verifying Pure SLH-DSA signed data, Algorithm 24 (slh\_verify) from Section 10.3 of [\[FIPS205\]](#) is used. When signing data using the HashSLH-DSA signature algorithm, Algorithm 23 (hash\_slh\_sign) from Section 10.2.2 of [\[FIPS205\]](#) is used. When verifying HashSLH-DSA signed data, Algorithm 25 (hash\_slh\_verify) from Section 10.3 of [\[FIPS205\]](#) is used. All four of these algorithms create a message,  $M'$ , from the message to be signed along with other data, and  $M'$  is operated on by internal SLH-DSA algorithms.  $M'$  may be constructed outside the module that performs the internal SLH-DSA algorithms.

In the case of HashSLH-DSA, there is a pre-hash component (PH\_M) of  $M'$ . PH\_M may be computed in the signing/verifying module; in which case, the entire message to be signed is sent to the module. Alternatively, PH\_M may be computed in a different module. In this case, either PH\_M is sent to the signing/verifying module, which creates  $M'$ , or  $M'$  is created outside the signing/verifying module and is sent to the module. HashSLH-DSA allows this implementation flexibility in order to reduce, and make consistent, the amount of data transferred to signing/verifying modules. The hash algorithm or extendable-output function (XOF) used to generate the pre-hash when signing and verifying with HashSLH-DSA is specified after the "-with-" component of the signature algorithm name. For example, when signing with id-hash-slh-dsa-sha2-128s-with-sha256, SHA-256 is used as the pre-hash algorithm. When pre-hashing is performed using SHAKE128, the output length is 256 bits. When pre-hashing is performed using SHAKE256, the output length is 512 bits.

Section 9.2 of [FIPS205] defines an SLH-DSA signature as three elements: R, SIG\_FOR, and SIG\_HT. The raw octet string encoding of an SLH-DSA signature is the concatenation of these three elements, i.e., R || SIG\_FOR || SIG\_HT. The raw octet string representing the signature is encoded directly in the BIT STRING without adding any additional ASN.1 wrapping. For example, in the Certificate structure, the raw signature value is encoded in the "signature" BIT STRING field.

## 5. Subject Public Key Fields

In the X.509 certificate, the subjectPublicKeyInfo field has the SubjectPublicKeyInfo type, which has the following ASN.1 syntax:

```
SubjectPublicKeyInfo {PUBLIC-KEY: IOSet} ::= SEQUENCE {  
    algorithm      AlgorithmIdentifier {PUBLIC-KEY, {IOSet}},  
    subjectPublicKey BIT STRING }
```

NOTE: The above syntax is from [RFC5912] and is compatible with the 2021 ASN.1 syntax [X680]. See [RFC5280] for the 1988 ASN.1 syntax.

The fields in SubjectPublicKeyInfo have the following meanings:

- algorithm is the algorithm identifier and parameters for the public key (see above).
- subjectPublicKey contains the byte stream of the public key.

[RFC9814] defines the following public key identifiers for Pure SLH-DSA:

```
pk-slh-dsa-sha2-128s PUBLIC-KEY ::= {  
    IDENTIFIER id-slh-dsa-sha2-128s  
    -- KEY no ASN.1 wrapping --  
    CERT-KEY-USAGE  
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }  
    -- PRIVATE-KEY no ASN.1 wrapping -- }  
  
pk-slh-dsa-sha2-128f PUBLIC-KEY ::= {  
    IDENTIFIER id-slh-dsa-sha2-128f  
    -- KEY no ASN.1 wrapping --  
    CERT-KEY-USAGE  
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }  
    -- PRIVATE-KEY no ASN.1 wrapping -- }  
  
pk-slh-dsa-sha2-192s PUBLIC-KEY ::= {  
    IDENTIFIER id-slh-dsa-sha2-192s  
    -- KEY no ASN.1 wrapping --  
    CERT-KEY-USAGE  
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }  
    -- PRIVATE-KEY no ASN.1 wrapping -- }  
  
pk-slh-dsa-sha2-192f PUBLIC-KEY ::= {  
    IDENTIFIER id-slh-dsa-sha2-192f  
    -- KEY no ASN.1 wrapping -- }
```

```
CERT-KEY-USAGE
  { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
-- PRIVATE-KEY no ASN.1 wrapping -- }

pk-slh-dsa-sha2-256s PUBLIC-KEY ::= {
  IDENTIFIER id-slh-dsa-sha2-256s
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-slh-dsa-sha2-256f PUBLIC-KEY ::= {
  IDENTIFIER id-slh-dsa-sha2-256f
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-slh-dsa-shake-128s PUBLIC-KEY ::= {
  IDENTIFIER id-slh-dsa-shake-128s
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-slh-dsa-shake-128f PUBLIC-KEY ::= {
  IDENTIFIER id-slh-dsa-shake-128f
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-slh-dsa-shake-192s PUBLIC-KEY ::= {
  IDENTIFIER id-slh-dsa-shake-192s
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-slh-dsa-shake-192f PUBLIC-KEY ::= {
  IDENTIFIER id-slh-dsa-shake-192f
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-slh-dsa-shake-256s PUBLIC-KEY ::= {
  IDENTIFIER id-slh-dsa-shake-256s
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-slh-dsa-shake-256f PUBLIC-KEY ::= {
  IDENTIFIER id-slh-dsa-shake-256f
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
```

```
-- PRIVATE-KEY no ASN.1 wrapping -- }  
  
SLH-DSA-PublicKey ::= OCTET STRING  
  
SLH-DSA-PrivateKey ::= OCTET STRING
```

The public key identifiers for HashSLH-DSA are defined here:

```
pk-hash-slh-dsa-sha2-128s-with-sha256 PUBLIC-KEY ::= {  
  IDENTIFIER id-hash-slh-dsa-sha2-128s-with-sha256  
  -- KEY no ASN.1 wrapping --  
  CERT-KEY-USAGE  
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }  
  -- PRIVATE-KEY no ASN.1 wrapping -- }  
  
pk-hash-slh-dsa-sha2-128f-with-sha256 PUBLIC-KEY ::= {  
  IDENTIFIER id-hash-slh-dsa-sha2-128f-with-sha256  
  -- KEY no ASN.1 wrapping --  
  CERT-KEY-USAGE  
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }  
  -- PRIVATE-KEY no ASN.1 wrapping -- }  
  
pk-hash-slh-dsa-sha2-192s-with-sha512 PUBLIC-KEY ::= {  
  IDENTIFIER id-hash-slh-dsa-sha2-192s-with-sha512  
  -- KEY no ASN.1 wrapping --  
  CERT-KEY-USAGE  
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }  
  -- PRIVATE-KEY no ASN.1 wrapping -- }  
  
pk-hash-slh-dsa-sha2-192f-with-sha512 PUBLIC-KEY ::= {  
  IDENTIFIER id-hash-slh-dsa-sha2-192f-with-sha512  
  -- KEY no ASN.1 wrapping --  
  CERT-KEY-USAGE  
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }  
  -- PRIVATE-KEY no ASN.1 wrapping -- }  
  
pk-hash-slh-dsa-sha2-256s-with-sha512 PUBLIC-KEY ::= {  
  IDENTIFIER id-hash-slh-dsa-sha2-256s-with-sha512  
  -- KEY no ASN.1 wrapping --  
  CERT-KEY-USAGE  
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }  
  -- PRIVATE-KEY no ASN.1 wrapping -- }  
  
pk-hash-slh-dsa-sha2-256f-with-sha512 PUBLIC-KEY ::= {  
  IDENTIFIER id-hash-slh-dsa-sha2-256f-with-sha512  
  -- KEY no ASN.1 wrapping --  
  CERT-KEY-USAGE  
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }  
  -- PRIVATE-KEY no ASN.1 wrapping -- }  
  
pk-hash-slh-dsa-shake-128s-with-shake128 PUBLIC-KEY ::= {  
  IDENTIFIER id-hash-slh-dsa-shake-128s-with-shake128  
  -- KEY no ASN.1 wrapping --  
  CERT-KEY-USAGE  
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }  
  -- PRIVATE-KEY no ASN.1 wrapping -- }
```

```

pk-hash-slh-dsa-shake-128f-with-shake128 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-128f-with-shake128
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-shake-192s-with-shake256 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-192s-with-shake256
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-shake-192f-with-shake256 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-192f-with-shake256
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-shake-256s-with-shake256 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-256s-with-shake256
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-shake-256f-with-shake256 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-256f-with-shake256
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

```

Section 9.1 of [FIPS205] defines an SLH-DSA public key as two  $n$ -byte elements: PK.seed and PK.root. The raw octet string encoding of an SLH-DSA public key is the concatenation of these two elements, i.e., PK.seed || PK.root. The octet string length is  $2 \cdot n$  bytes, where  $n$  is 16, 24, or 32, depending on the SLH-DSA parameter set. When used in a SubjectPublicKeyInfo type, the subjectPublicKey BIT STRING contains the raw octet string encoding of the public key.

[RFC9814] defines the SLH-DSA-PublicKey and SLH-DSA-PrivateKey ASN.1 OCTET STRING types to provide an option for encoding a Pure SLH-DSA public or private key in an environment that uses ASN.1 encoding but doesn't define its own mapping of an SLH-DSA raw octet string to ASN.1. HashSLH-DSA public and private keys can use SLH-DSA-PublicKey and SLH-DSA-PrivateKey in the same way. To map an SLH-DSA-PublicKey OCTET STRING to a SubjectPublicKeyInfo, the OCTET STRING is mapped to the subjectPublicKey field (a value of type BIT STRING) as follows: The most significant bit of the OCTET STRING value becomes the most significant bit of the BIT STRING value, and so on; the least significant bit of the OCTET STRING becomes the least significant bit of the BIT STRING.

The AlgorithmIdentifier for an SLH-DSA public key **MUST** use one of the id-slh-dsa-\* or id-hash-slh-dsa-\* object identifiers from [Section 3](#). The parameters field of the AlgorithmIdentifier for the SLH-DSA public key **MUST** be absent.

[Appendix C.1](#) contains an example of an id-slh-dsa-sha2-128s public key encoded using the textual encoding defined in [\[RFC7468\]](#).

## 6. Key Usage Bits

The intended application for the key is indicated in the keyUsage certificate extension; see [Section 4.2.1.3](#) of [\[RFC5280\]](#). If the keyUsage extension is present in a certificate that indicates an id-slh-dsa-\* (Pure SLH-DSA) or id-hash-slh-dsa-\* (HashSLH-DSA) identifier in the SubjectPublicKeyInfo, then at least one of the following **MUST** be present:

- digitalSignature
- nonRepudiation
- keyCertSign
- cRLSign

If the keyUsage extension is present in a certificate that indicates an id-slh-dsa-\* (Pure SLH-DSA) or id-hash-slh-dsa-\* (HashSLH-DSA) identifier in the SubjectPublicKeyInfo, then the following **MUST NOT** be present:

- keyEncipherment
- dataEncipherment
- keyAgreement
- encipherOnly
- decipherOnly

Requirements about the keyUsage extension bits defined in [\[RFC5280\]](#) still apply.

## 7. Private Key Format

"Asymmetric Key Packages" [\[RFC5958\]](#) describes how to encode a private key in a structure that both identifies what algorithm the private key is for and optionally allows for the public key and additional attributes about the key to be included as well. For illustration, the ASN.1 structure OneAsymmetricKey is replicated below.

```
OneAsymmetricKey ::= SEQUENCE {  
    version Version,  
    privateKeyAlgorithm PrivateKeyAlgorithmIdentifier,  
    privateKey PrivateKey,  
    attributes [0] IMPLICIT Attributes OPTIONAL,  
    ...  
    [[2: publicKey [1] IMPLICIT PublicKey OPTIONAL ]],  
    ...  
}  
  
PrivateKey ::= OCTET STRING  
  
PublicKey ::= BIT STRING
```

NOTE: The above syntax is from [\[RFC5958\]](#) and is compatible with the 2021 ASN.1 syntax [\[X680\]](#).

Section 9.1 of [\[FIPS205\]](#) defines an SLH-DSA private key as four n-byte elements: SK.seed, SK.prf, PK.seed, and PK.root. The raw octet string encoding of an SLH-DSA private key is the concatenation of these four elements, i.e., SK.seed || SK.prf || PK.seed || PK.root. The octet string length is 4\*n bytes, where n is 16, 24, or 32, depending on the SLH-DSA parameter set. When used in a OneAsymmetricKey type, the privateKey OCTET STRING contains the raw octet string encoding of the private key.

When an SLH-DSA public key is included in a OneAsymmetricKey type, it is encoded in the same manner as in a SubjectPublicKeyInfo type. That is, the publicKey BIT STRING contains the raw octet string encoding of the public key.

[Appendix C.2](#) contains an example of an id-slh-dsa-sha2-128s private key encoded using the textual encoding defined in [\[RFC7468\]](#).

NOTE: There exist some private key import functions that have not picked up the new ASN.1 structure OneAsymmetricKey, which is defined in [\[RFC5958\]](#). This means that they will not accept a private key structure that contains the public key field. This means a balancing act needs to be done between being able to do a consistency check on the key pair and widest ability to import the key.

## 8. Operational Considerations

SLH-DSA uses the same OID to identify a public key and a signature algorithm. The implication of this is that, despite being mathematically possible, an SLH-DSA key identified by a Pure SLH-DSA OID is not permitted to be used to generate or verify a signature identified by a HashSLH-DSA OID, and vice versa.

Certification authority (CA) operators will need to decide in advance whether their CA certificates will use Pure SLH-DSA or HashSLH-DSA and assign the appropriate OID to the public and private keys when generating their certificate. Some of the following considerations may affect this decision.

- When using an external signing module, such as a Hardware Security Module (HSM), the size of data that can be transferred to and processed by the signature module may be limited. SLH-DSA performs two passes on the internal M' message, so it must be held in memory. Using HashSLH-DSA reduces the size of M'.
- Large CRLs might also exceed the size limits of HSM signing operations when using Pure SLH-DSA. One way to limit the size of CRLs is to make use of CRL Distribution Points and Issuing Distribution Points to create partitioned CRLs in accordance with [Section 5.2.5 of \[RFC5280\]](#).
- End Entity (EE) certificates with many subject alternative names (SANs) might also exceed the size limits of HSM signing operations.
- Potential verifiers' environments might need to be considered. The entire certificate or CRL needs to be held in memory during SLH-DSA signature verification; it cannot be streamed. In particular, there is a randomizer (R) that is extracted from the SLH-DSA signature and fed to a digest function before M' is. Thus, to stream a message for SLH-DSA verification, the signature must come before the message. This is not the case for certificates and CRLs. Using HashSLH-DSA reduces the size of the M' being held in memory.

An SLH-DSA private key has a very large ( $2^{64}$ ) number of signatures it can safely generate (see [Section 9](#)). If an operator might conceivably generate a number of signatures approaching this limit, they should mitigate potential harm by tracking the number of signatures generated and destroying the private key once an appropriate limit is reached or by setting the "Not After" (expiration) date of the certificate such that the limit couldn't possibly be surpassed given the rate of signing.

## 9. Security Considerations

The security considerations of [\[RFC5280\]](#) apply accordingly. Moreover, the security aspects mentioned throughout [\[FIPS205\]](#) should be taken into account; for instance, see Sections 3.1 and 3.2 or the beginning of Section 11.

The security of SLH-DSA relies on the security properties of the internal hash and XOF functions. In particular, it relies on these functions being preimage resistant, but it does not rely on them being collision resistant. Since HashSLH-DSA performs a pre-hash before signing, it relies on both preimage resistance and collision resistance of the pre-hash function. In order to achieve an appropriate level of collision resistance, the output length of the pre-hash functions used for HashSLH-DSA is twice the length of the internal hash and XOF functions.

Implementations **MUST** protect the private keys. Compromise of the private keys may result in the ability to forge signatures.

When generating an SLH-DSA key pair, an implementation **MUST** generate each key pair independently of all other key pairs in the SLH-DSA hypertree.

An SLH-DSA tree **MUST NOT** be used for more than  $2^{64}$  signing operations.

The generation of private keys relies on random numbers. The use of inadequate pseudorandom number generators (PRNGs) to generate these values can result in little or no security. An attacker may find it much easier to reproduce the PRNG environment that produced the keys, searching the resulting small set of possibilities, rather than brute force searching the whole key space. The generation of quality random numbers is difficult; see Section 3.1 of [FIPS205] for some additional information.

Fault attacks can lead to forgeries of message signatures; see [CMP2018] and [Ge2023]. Verifying a signature before releasing the signature value is a typical fault attack countermeasure; however, this countermeasure is not effective for SLH-DSA [Ge2023]. Redundancy by replicating the signature generation process can be used as an effective fault attack countermeasure for SLH-DSA [Ge2023]; however, the SLH-DSA signature generation is already considered slow.

Likewise, passive power and emissions side-channel attacks can leak the SLH-DSA private signing key, and countermeasures can be taken against these attacks [SLotH].

## 10. IANA Considerations

For the ASN.1 module in [Appendix A](#) of this document, IANA has assigned an object identifier (OID) for the module identifier (120) with a Description of "id-mod-x509-slh-dsa-2024". The OID for the module has been allocated in the "SMI Security for PKIX Module Identifier" registry (1.3.6.1.5.5.7.0).

## 11. References

### 11.1. Normative References

- [CSOR] NIST, "Computer Security Objects Register (CSOR)", 13 June 2025, <<https://csrc.nist.gov/projects/computer-security-objects-register/algorithm-registration>>.
- [FIPS205] NIST, "Stateless Hash-Based Digital Signature Standard", NIST FIPS 205, DOI 10.6028/NIST.FIPS.205, 13 August 2024, <<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf>>.
- [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>>.
- [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>>.
- [RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958, DOI 10.17487/RFC5958, August 2010, <<https://www.rfc-editor.org/info/rfc5958>>.
- [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>>.
- [RFC9814] Housley, R., Fluhner, S., Kampanakis, P., and B. Westerbaan, "Use of the SLH-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS)", RFC 9814, DOI 10.17487/RFC9814, July 2025, <<https://www.rfc-editor.org/info/rfc9814>>.
- [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>>.

## 11.2. Informative References

- [CMP2018] Castelnovi, L., Martinelli, A., and T. Prest, "Grafting Trees: A Fault Attack Against the SPHINCS Framework", Post-Quantum Cryptography (PQCrypto 2018), Lecture Notes in Computer Science, vol. 10786, pp. 165-184, 2018, <[https://link.springer.com/chapter/10.1007/978-3-319-79063-3\\_8](https://link.springer.com/chapter/10.1007/978-3-319-79063-3_8)>.
- [FIPS180] NIST, "Secure Hash Standard (SHS)", NIST FIPS 180-4, DOI 10.6028/NIST.FIPS.180-4, August 2015, <<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>>.
- [FIPS202] NIST, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", NIST FIPS 202, DOI 10.6028/NIST.FIPS.202, August 2015, <<http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf>>.
- [Ge2023] Genêt, A., "On Protecting SPHINCS+ Against Fault Attacks", TCHES, vol. 2023, no. 2, pp. 80-114, DOI 10.46586/tches.v2023.i2.80-114, March 2023, <<https://doi.org/10.46586/tches.v2023.i2.80-114>>.
- [NIST-PQC] NIST, "Post-Quantum Cryptography (PQC)", 28 July 2025, <<https://csrc.nist.gov/projects/post-quantum-cryptography>>.
- [RFC7468] Josefsson, S. and S. Leonard, "Textual Encodings of PKIX, PKCS, and CMS Structures", RFC 7468, DOI 10.17487/RFC7468, April 2015, <<https://www.rfc-editor.org/info/rfc7468>>.

- [RFC8410] Josefsson, S. and J. Schaad, "Algorithm Identifiers for Ed25519, Ed448, X25519, and X448 for Use in the Internet X.509 Public Key Infrastructure", RFC 8410, DOI 10.17487/RFC8410, August 2018, <<https://www.rfc-editor.org/info/rfc8410>>.
- [RFC8411] Schaad, J. and R. Andrews, "IANA Registration for the Cryptographic Algorithm Object Identifier Range", RFC 8411, DOI 10.17487/RFC8411, August 2018, <<https://www.rfc-editor.org/info/rfc8411>>.
- [RFC9881] Massimo, J., Kampanakis, P., Turner, S., and B. E. Westerbaan, "Internet X.509 Public Key Infrastructure -- Algorithm Identifiers for the Module-Lattice-Based Digital Signature Algorithm (ML-DSA)", RFC 9881, DOI 10.17487/RFC9881, October 2025, <<https://www.rfc-editor.org/info/rfc9881>>.
- [SLotH] Saarinen, M-J., "Accelerating SLH-DSA by Two Orders of Magnitude with a Single Hash Unit", Cryptology ePrint Archive, Paper 2024/367, DOI 10.1007/978-3-031-68376-3\_9, 2024, <<https://eprint.iacr.org/2024/367.pdf>>.

## Appendix A. ASN.1 Module

This appendix includes the ASN.1 module [X680] for SLH-DSA. Note that as per [RFC5280], certificates use the Distinguished Encoding Rules; see [X690]. This module imports objects from [RFC5912] and [RFC9814].

```
<CODE BEGINS>
X509-SLH-DSA-Module-2024
{ iso(1) identified-organization(3) dod(6) internet(1) security(5)
  mechanisms(5) pkix(7) id-mod(0) id-mod-x509-slh-dsa-2024(120) }

DEFINITIONS IMPLICIT TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS
  PUBLIC-KEY, SIGNATURE-ALGORITHM, SMIME-CAPS
  FROM AlgorithmInformation-2009 -- in [RFC5912]
  { iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-algorithmInformation-02(58) }

pk-slh-dsa-sha2-128s, pk-slh-dsa-sha2-128f,
pk-slh-dsa-sha2-192s, pk-slh-dsa-sha2-192f,
pk-slh-dsa-sha2-256s, pk-slh-dsa-sha2-256f,
pk-slh-dsa-shake-128s, pk-slh-dsa-shake-128f,
pk-slh-dsa-shake-192s, pk-slh-dsa-shake-192f,
pk-slh-dsa-shake-256s, pk-slh-dsa-shake-256f,
sa-slh-dsa-sha2-128s, sa-slh-dsa-sha2-128f,
sa-slh-dsa-sha2-192s, sa-slh-dsa-sha2-192f,
sa-slh-dsa-sha2-256s, sa-slh-dsa-sha2-256f,
sa-slh-dsa-shake-128s, sa-slh-dsa-shake-128f,
sa-slh-dsa-shake-192s, sa-slh-dsa-shake-192f,
sa-slh-dsa-shake-256s, sa-slh-dsa-shake-256f
  FROM SLH-DSA-Module-2024 -- in [RFC9814]
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9)
```

```
    id-smime(16) id-mod(0) id-mod-slh-dsa-2024(81) } ;

--
-- HashSLH-DSA object identifiers from [CSOR]
--

nistAlgorithms OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3) 4 }

sigAlgs OBJECT IDENTIFIER ::= { nistAlgorithms 3 }

id-hash-slh-dsa-sha2-128s-with-sha256 OBJECT IDENTIFIER ::= {
    sigAlgs 35 }

id-hash-slh-dsa-sha2-128f-with-sha256 OBJECT IDENTIFIER ::= {
    sigAlgs 36 }

id-hash-slh-dsa-sha2-192s-with-sha512 OBJECT IDENTIFIER ::= {
    sigAlgs 37 }

id-hash-slh-dsa-sha2-192f-with-sha512 OBJECT IDENTIFIER ::= {
    sigAlgs 38 }

id-hash-slh-dsa-sha2-256s-with-sha512 OBJECT IDENTIFIER ::= {
    sigAlgs 39 }

id-hash-slh-dsa-sha2-256f-with-sha512 OBJECT IDENTIFIER ::= {
    sigAlgs 40 }

id-hash-slh-dsa-shake-128s-with-shake128 OBJECT IDENTIFIER ::= {
    sigAlgs 41 }

id-hash-slh-dsa-shake-128f-with-shake128 OBJECT IDENTIFIER ::= {
    sigAlgs 42 }

id-hash-slh-dsa-shake-192s-with-shake256 OBJECT IDENTIFIER ::= {
    sigAlgs 43 }

id-hash-slh-dsa-shake-192f-with-shake256 OBJECT IDENTIFIER ::= {
    sigAlgs 44 }

id-hash-slh-dsa-shake-256s-with-shake256 OBJECT IDENTIFIER ::= {
    sigAlgs 45 }

id-hash-slh-dsa-shake-256f-with-shake256 OBJECT IDENTIFIER ::= {
    sigAlgs 46 }

--
-- HashSLH-DSA public key identifiers
--

pk-hash-slh-dsa-sha2-128s-with-sha256 PUBLIC-KEY ::= {
    IDENTIFIER id-hash-slh-dsa-sha2-128s-with-sha256
    -- KEY no ASN.1 wrapping --
    CERT-KEY-USAGE
        { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
    -- PRIVATE-KEY no ASN.1 wrapping -- }
```

```
pk-hash-slh-dsa-sha2-128f-with-sha256 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-128f-with-sha256
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-sha2-192s-with-sha512 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-192s-with-sha512
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-sha2-192f-with-sha512 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-192f-with-sha512
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-sha2-256s-with-sha512 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-256s-with-sha512
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-sha2-256f-with-sha512 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-256f-with-sha512
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-shake-128s-with-shake128 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-128s-with-shake128
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-shake-128f-with-shake128 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-128f-with-shake128
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-shake-192s-with-shake256 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-192s-with-shake256
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-shake-192f-with-shake256 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-192f-with-shake256
```

```
-- KEY no ASN.1 wrapping --
CERT-KEY-USAGE
  { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
-- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-shake-256s-with-shake256 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-256s-with-shake256
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

pk-hash-slh-dsa-shake-256f-with-shake256 PUBLIC-KEY ::= {
  IDENTIFIER id-hash-slh-dsa-shake-256f-with-shake256
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  -- PRIVATE-KEY no ASN.1 wrapping -- }

--
-- HashSLH-DSA signature algorithm identifiers
--

sa-hash-slh-dsa-sha2-128s-with-sha256 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-128s-with-sha256
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-sha2-128s-with-sha256 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-sha2-128s-with-sha256 } }

sa-hash-slh-dsa-sha2-128f-with-sha256 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-128f-with-sha256
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-sha2-128f-with-sha256 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-sha2-128f-with-sha256 } }

sa-hash-slh-dsa-sha2-192s-with-sha512 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-192s-with-sha512
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-sha2-192s-with-sha512 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-sha2-192s-with-sha512 } }

sa-hash-slh-dsa-sha2-192f-with-sha512 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-192f-with-sha512
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-sha2-192f-with-sha512 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-sha2-192f-with-sha512 } }

sa-hash-slh-dsa-sha2-256s-with-sha512 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-256s-with-sha512
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-sha2-256s-with-sha512 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-sha2-256s-with-sha512 } }
```

```

sa-hash-slh-dsa-sha2-256f-with-sha512 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-sha2-256f-with-sha512
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-sha2-256f-with-sha512 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-sha2-256f-with-sha512 } }

sa-hash-slh-dsa-shake-128s-with-shake128 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-shake-128s-with-shake128
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-shake-128s-with-shake128 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-shake-128s-with-shake128 } }

sa-hash-slh-dsa-shake-128f-with-shake128 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-shake-128f-with-shake128
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-shake-128f-with-shake128 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-shake-128f-with-shake128 } }

sa-hash-slh-dsa-shake-192s-with-shake256 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-shake-192s-with-shake256
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-shake-192s-with-shake256 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-shake-192s-with-shake256 } }

sa-hash-slh-dsa-shake-192f-with-shake256 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-shake-192f-with-shake256
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-shake-192f-with-shake256 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-shake-192f-with-shake256 } }

sa-hash-slh-dsa-shake-256s-with-shake256 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-shake-256s-with-shake256
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-shake-256s-with-shake256 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-shake-256s-with-shake256 } }

sa-hash-slh-dsa-shake-256f-with-shake256 SIGNATURE-ALGORITHM ::= {
  IDENTIFIER id-hash-slh-dsa-shake-256f-with-shake256
  PARAMS ARE absent
  PUBLIC-KEYS { pk-hash-slh-dsa-shake-256f-with-shake256 }
  SMIME-CAPS {
    IDENTIFIED BY id-hash-slh-dsa-shake-256f-with-shake256 } }

--
-- Expand SignatureAlgorithms from RFC 5912
--
SignatureAlgorithms SIGNATURE-ALGORITHM ::= {
  sa-slh-dsa-sha2-128s |
  sa-slh-dsa-sha2-128f |
  sa-slh-dsa-sha2-192s |
  sa-slh-dsa-sha2-192f |
  sa-slh-dsa-sha2-256s |

```

```

sa-slh-dsa-sha2-256f |
sa-slh-dsa-shake-128s |
sa-slh-dsa-shake-128f |
sa-slh-dsa-shake-192s |
sa-slh-dsa-shake-192f |
sa-slh-dsa-shake-256s |
sa-slh-dsa-shake-256f |
sa-hash-slh-dsa-sha2-128s-with-sha256 |
sa-hash-slh-dsa-sha2-128f-with-sha256 |
sa-hash-slh-dsa-sha2-192s-with-sha512 |
sa-hash-slh-dsa-sha2-192f-with-sha512 |
sa-hash-slh-dsa-sha2-256s-with-sha512 |
sa-hash-slh-dsa-sha2-256f-with-sha512 |
sa-hash-slh-dsa-shake-128s-with-shake128 |
sa-hash-slh-dsa-shake-128f-with-shake128 |
sa-hash-slh-dsa-shake-192s-with-shake256 |
sa-hash-slh-dsa-shake-192f-with-shake256 |
sa-hash-slh-dsa-shake-256s-with-shake256 |
sa-hash-slh-dsa-shake-256f-with-shake256,
... }

SMimeCaps SMIME-CAPS ::= {
  sa-slh-dsa-sha2-128s.&smimeCaps |
  sa-slh-dsa-sha2-128f.&smimeCaps |
  sa-slh-dsa-sha2-192s.&smimeCaps |
  sa-slh-dsa-sha2-192f.&smimeCaps |
  sa-slh-dsa-sha2-256s.&smimeCaps |
  sa-slh-dsa-sha2-256f.&smimeCaps |
  sa-slh-dsa-shake-128s.&smimeCaps |
  sa-slh-dsa-shake-128f.&smimeCaps |
  sa-slh-dsa-shake-192s.&smimeCaps |
  sa-slh-dsa-shake-192f.&smimeCaps |
  sa-slh-dsa-shake-256s.&smimeCaps |
  sa-slh-dsa-shake-256f.&smimeCaps |
  sa-hash-slh-dsa-sha2-128s-with-sha256.&smimeCaps |
  sa-hash-slh-dsa-sha2-128f-with-sha256.&smimeCaps |
  sa-hash-slh-dsa-sha2-192s-with-sha512.&smimeCaps |
  sa-hash-slh-dsa-sha2-192f-with-sha512.&smimeCaps |
  sa-hash-slh-dsa-sha2-256s-with-sha512.&smimeCaps |
  sa-hash-slh-dsa-sha2-256f-with-sha512.&smimeCaps |
  sa-hash-slh-dsa-shake-128s-with-shake128.&smimeCaps |
  sa-hash-slh-dsa-shake-128f-with-shake128.&smimeCaps |
  sa-hash-slh-dsa-shake-192s-with-shake256.&smimeCaps |
  sa-hash-slh-dsa-shake-192f-with-shake256.&smimeCaps |
  sa-hash-slh-dsa-shake-256s-with-shake256.&smimeCaps |
  sa-hash-slh-dsa-shake-256f-with-shake256.&smimeCaps,
  ... }

--
-- Expand PublicKeyAlgorithms from RFC 5912
--
PublicKeyAlgorithms PUBLIC-KEY ::= {
  pk-slh-dsa-sha2-128s |
  pk-slh-dsa-sha2-128f |
  pk-slh-dsa-sha2-192s |
  pk-slh-dsa-sha2-192f |
  pk-slh-dsa-sha2-256s |
  pk-slh-dsa-sha2-256f |

```

```

pk-slh-dsa-shake-128s |
pk-slh-dsa-shake-128f |
pk-slh-dsa-shake-192s |
pk-slh-dsa-shake-192f |
pk-slh-dsa-shake-256s |
pk-slh-dsa-shake-256f |
pk-hash-slh-dsa-sha2-128s-with-sha256 |
pk-hash-slh-dsa-sha2-128f-with-sha256 |
pk-hash-slh-dsa-sha2-192s-with-sha512 |
pk-hash-slh-dsa-sha2-192f-with-sha512 |
pk-hash-slh-dsa-sha2-256s-with-sha512 |
pk-hash-slh-dsa-sha2-256f-with-sha512 |
pk-hash-slh-dsa-shake-128s-with-shake128 |
pk-hash-slh-dsa-shake-128f-with-shake128 |
pk-hash-slh-dsa-shake-192s-with-shake256 |
pk-hash-slh-dsa-shake-192f-with-shake256 |
pk-hash-slh-dsa-shake-256s-with-shake256 |
pk-hash-slh-dsa-shake-256f-with-shake256,
... }

END

<CODE ENDS>

```

## Appendix B. Security Strengths

Instead of defining the strength of a quantum algorithm in a traditional manner using precise estimates of the number of bits of security, NIST defined a collection of broad security strength categories. Each category is defined by a comparatively easy-to-analyze reference primitive that covers a range of security strengths offered by existing NIST standards in symmetric cryptography, which NIST expects to offer significant resistance to quantum cryptanalysis. These categories describe any attack that breaks the relevant security definition that must require computational resources comparable to or greater than those required for:

- Level 1 - key search on a block cipher with a 128-bit key (e.g., AES128),
- Level 2 - collision search on a 256-bit hash function (e.g., SHA256/ SHA3-256),
- Level 3 - key search on a block cipher with a 192-bit key (e.g., AES192),
- Level 4 - collision search on a 384-bit hash function (e.g., SHA384/SHA3-384), and
- Level 5 - key search on a block cipher with a 256-bit key (e.g., AES 256).

The SLH-DSA parameter sets defined for NIST security levels 1, 3, and 5 are listed in [Table 1](#), along with the resulting signature, public key, and private key sizes in bytes. The HashSLH-DSA parameter sets have the same values as the Pure SLH-DSA equivalents.

| OID                         | NIST Level | Size (in bytes) |          |           |
|-----------------------------|------------|-----------------|----------|-----------|
|                             |            | Sig.            | Pub. Key | Priv. Key |
| id-(hash-)slh-dsa-sha2-128s | 1          | 7856            | 32       | 64        |

| OID                          | NIST Level | Size (in bytes) |          |           |
|------------------------------|------------|-----------------|----------|-----------|
|                              |            | Sig.            | Pub. Key | Priv. Key |
| id-(hash-)slh-dsa-sha2-128f  | 1          | 17088           | 32       | 64        |
| id-(hash-)slh-dsa-sha2-192s  | 3          | 16224           | 48       | 96        |
| id-(hash-)slh-dsa-sha2-192f  | 3          | 35664           | 48       | 96        |
| id-(hash-)slh-dsa-sha2-256s  | 5          | 29792           | 64       | 128       |
| id-(hash-)slh-dsa-sha2-256f  | 5          | 49856           | 64       | 128       |
| id-(hash-)slh-dsa-shake-128s | 1          | 7856            | 32       | 64        |
| id-(hash-)slh-dsa-shake-128f | 1          | 17088           | 32       | 64        |
| id-(hash-)slh-dsa-shake-192s | 3          | 16224           | 48       | 96        |
| id-(hash-)slh-dsa-shake-192f | 3          | 35664           | 48       | 96        |
| id-(hash-)slh-dsa-shake-256s | 5          | 29792           | 64       | 128       |
| id-(hash-)slh-dsa-shake-256f | 5          | 49856           | 64       | 128       |

Table 1: SLH-DSA Security Strengths

## Appendix C. Examples

This appendix contains examples of SLH-DSA public keys, private keys, and certificates.

### C.1. Example Public Key

An example of an SLH-DSA public key using id-slh-dsa-sha2-128s:

```
-----BEGIN PUBLIC KEY-----
MDAwCwYJYIZIAWUDBAMUAYEAK4EJ7Hd8qk4fAkzPz5SX2ZGAUJKA9CVq8rB6+AKJ
tJQ=
-----END PUBLIC KEY-----
```

```

0  48: SEQUENCE {
2  11: SEQUENCE {
4   9: OBJECT IDENTIFIER '2 16 840 1 101 3 4 3 20'
      : }
15 33: BIT STRING
      : 2B 81 09 EC 77 7C AA 4E 1F 02 4C CF CF 94 97 D9
      : 91 80 50 92 80 F4 25 6A F2 B0 7A F8 02 89 B4 94
      : }

```

## C.2. Example Private Key

An example of an SLH-DSA private key without the public key using id-slh-dsa-sha2-128s:

```

-----BEGIN PRIVATE KEY-----
MFICAQAwCwYJYIZIAWUDBAMUBECiJjvKRYYNlIxYASVI9YhZ3+tkNUetgZ6Mn4N
HmSlASuBCex3fKp0HwJMz8+U19mRgFCSgPQlavKwevgCibSU
-----END PRIVATE KEY-----

```

```

0  82: SEQUENCE {
2   1: INTEGER 0
5  11: SEQUENCE {
7   9: OBJECT IDENTIFIER '2 16 840 1 101 3 4 3 20'
      : }
18 64: OCTET STRING
      : A2 26 3B CA 45 86 08 36 52 31 60 04 95 23 D6 21
      : 67 7F AD 90 D5 1E B6 06 7A 32 7E 0D 1E 64 A5 01
      : 2B 81 09 EC 77 7C AA 4E 1F 02 4C CF CF 94 97 D9
      : 91 80 50 92 80 F4 25 6A F2 B0 7A F8 02 89 B4 94
      : }

```

## C.3. Example Certificate

An example of a self-signed SLH-DSA certificate using id-slh-dsa-sha2-128s:

## Certificate:

## Data:

Version: 3 (0x2)

Serial Number:

43:85:63:a2:69:01:99:2c:39:cf:bc:40:57:1b:5f:a3:  
cc:c7:88:45

Signature Algorithm: slhdsa\_sha2\_128s

Issuer: C=FR, L=Paris, O=Bogus SLH-DSA-SHA2-128s CA

Validity

Not Before: Oct 16 13:42:12 2024 GMT

Not After : Oct 14 13:42:12 2034 GMT

Subject: C=FR, L=Paris, O=Bogus SLH-DSA-SHA2-128s CA

Subject Public Key Info:

Public Key Algorithm: slhdsa\_sha2\_128s

slhdsa\_sha2\_128s public key:

PQ key material:

2b:81:09:ec:77:7c:aa:4e:1f:02:4c:cf:cf:94:97:  
d9:91:80:50:92:80:f4:25:6a:f2:b0:7a:f8:02:89:  
b4:94

X509v3 extensions:

X509v3 Subject Key Identifier:

CD:59:36:AA:FE:C4:11:C7:A4:72:69:3F:0B:E8:B3:8B:  
21:7B:19:ED

X509v3 Authority Key Identifier:

CD:59:36:AA:FE:C4:11:C7:A4:72:69:3F:0B:E8:B3:8B:  
21:7B:19:ED

X509v3 Basic Constraints: critical

CA:TRUE

X509v3 Key Usage: critical

Certificate Sign, CRL Sign

Signature Algorithm: slhdsa\_sha2\_128s

Signature Value:

aa:a0:51:de:b0:c3:14:d0:cd:fb:12:46:a2:31:20:c9:ed:ab:  
3f:dc:57:a5:fb:45:f6:f0:3b:7f:e3:5a:8c:b5:87:1e:1f:0b:  
15:9f:aa:56:68:43:7e:ea:23:05:21:d1:33:cb:84:61:55:7e:  
39:74:18:3c:ea:8e:01:a4:8d:9a:fb:35:74:69:c9:62:35:7f:  
0e:34:01:1c:90:41:97:13:ff:c5:a4:65:ae:0f:bf:9b:32:d2:  
2a:2c:97:86:2d:49:eb:ba:ae:9a:70:e7:35:67:3f:0a:7e:3a:  
dd:0b:66:4e:f8:45:b2:e6:d8:70:ab:fb:72:60:eb:85:ae:62:  
3c:a4:bf:3c:7a:e5:dd:4a:24:e2:4e:d0:b5:3b:c3:ac:e9:26:  
f8:6c:ca:3b:e1:46:15:7f:18:c5:41:40:90:73:b9:19:63:86:  
23:3a:b2:7f:12:3a:5f:bb:c3:10:6c:4e:b2:62:ee:3b:4b:c5:  
e2:69:24:74:3e:6e:81:e2:68:48:c8:27:25:bc:b2:ac:da:a8:  
ae:75:5a:5c:09:22:1c:be:95:0a:0b:5e:0c:08:49:42:3a:0d:  
2d:fb:89:3b:b3:15:de:ee:e7:b2:5e:1f:a6:f0:4a:f6:65:c1:  
5d:5e:05:7a:6d:2a:e7:c2:c3:20:37:ce:ab:0f:6c:ea:c9:39:  
f3:28:d1:75:81:31:7f:01:e2:09:c8:56:81:50:cf:4e:fa:82:  
1a:60:3e:87:bf:61:ca:a0:40:27:95:bf:f8:4f:04:b1:fd:1f:  
7f:ce:29:fa:15:5c:ef:94:9a:f6:f0:0c:7f:09:7f:ec:b6:36:  
26:83:69:aa:2d:69:9e:17:7a:15:aa:9b:51:43:c1:90:7c:c9:  
69:3a:5a:b1:ee:77:c9:28:e7:21:d8:93:0a:80:19:9c:5e:b7:  
61:5f:14:6c:9a:00:22:aa:4d:b8:86:03:b5:83:4a:e9:f3:5a:  
76:cc:a3:3b:e4:13:94:f7:56:96:56:33:dd:19:d9:3d:8d:55:  
ab:99:e5:00:24:f7:ff:f4:ee:08:47:8d:43:b3:f4:e3:3a:d5:  
12:ef:04:00:99:62:a1:5e:cd:5f:9f:90:f3:c2:8e:35:9b:8a:  
46:ec:54:4e:13:20:59:5f:63:d9:61:b1:e2:c4:36:d2:e5:27:  
56:1f:53:59:9c:24:ec:6a:79:2b:1d:6a:f2:93:38:d8:eb:7a:

```
cd:d7:8a:c8:98:d4:87:61:bf:79:3c:2a:64:42:0f:5b:15:b4:
bd:c0:c7:c4:de:20:4c:bb:d8:0f:61:2e:aa:67:e1:a7:ff:0d:
b7:dd:05:cf:5c:cb:0c:46:26:e0:d9:48:cb:45:76:27:88:51:
49:df:4c:16:65:8c:1a:84:82:09:f3:d4:ee:c4:2a:17:a9:7b:
c0:77:24:fd:4f:00:98:12:ed:10:e7:67:c3:7d:54:78:0f:c8:
67:7f:f4:f2:80:2b:1b:34:0c:fa:5f:c4:12:85:1c:5f:e6:84:
8d:ce:12:e7:ae:f5:ef:eb:96:5f:62:6f:87:3a:35:67:ca:d8:
ad:b5:55:0b:0d:06:91:d3:9d:1a:96:2e:67:d8:b1:0e:8f:07:
3f:7b:d6:fe:b5:76:62:19:83:f6:d2:08:35:3b:9f:1d:0a:f7:
14:d2:45:50:70:5c:91:cc:b5:0f:4b:ef:79:ef:d3:c7:bd:02:
7a:fa:8b:83:cd:31:07:b0:f7:8a:79:c4:68:19:de:01:f8:73:
1a:6d:8a:c7:54:c8:4b:9a:40:53:e3:4b:e4:bd:3a:52:50:c6:
de:de:19:d7:9e:a8:88:70:f1:70:a6:11:55:b0:46:5e:40:37:
b2:90:5c:91:76:bd:20:1d:24:db:71:33:81:b8:47:ef:ec:7e:
78:d2:25:2b:4b:e2:6e:01:81:d4:12:ff:40:ff:e0:d7:90:29:
85:80:e6:4a:f5:5b:32:6c:b7:05:1c:20:27:e0:98:57:80:e7:
a2:97:cb:91:ce:d9:c1:a3:5f:dc:24:7f:b8:f5:5c:da:91:83:
e5:ae:8c:65:73:84:6a:5b:c9:3f:97:51:7d:cc:3f:d6:39:e1:
71:f1:54:8d:1f:4f:33:70:cc:07:f8:03:70:be:8c:81:e1:5d:
73:c1:9c:be:7c:3d:69:c0:cc:72:90:cf:65:38:35:71:16:ae:
1d:e2:a6:08:c8:7b:dd:c0:30:f4:b4:2a:45:fc:05:e6:1c:ef:
af:f3:53:03:2f:76:b5:7b:f1:a9:7d:16:33:b1:b5:c2:4f:9b:
55:7b:0d:22:f6:08:4b:38:b2:67:4e:d9:f8:f1:65:03:d6:5a:
1f:1f:8b:cb:da:78:fc:7b:52:a5:d7:1b:35:b2:cd:06:7e:1e:
1d:8b:60:40:91:74:2f:91:c9:c6:c7:c4:01:f5:2f:10:c2:ea:
ab:84:f6:f6:2e:fc:77:c1:85:28:90:a5:11:dc:ed:07:78:c2:
74:9c:60:86:69:40:3c:17:9b:3a:e5:e8:65:22:c2:7f:d9:88:
be:43:6a:31:90:d5:23:37:eb:93:70:e4:bc:34:94:4f:af:a4:
c1:6f:f3:30:1b:c6:e1:f5:f1:d8:7b:a4:4e:6e:69:be:82:d0:
80:a8:ae:99:44:e1:d6:fa:45:e5:05:a5:52:0a:5d:60:17:3a:
1e:2e:dd:2e:b4:86:93:31:93:0f:ca:5f:05:52:8e:31:15:e8:
8b:30:88:33:d7:da:91:52:40:3c:d7:18:bc:72:8d:88:b2:65:
c5:fe:0a:7c:50:44:7e:0f:b6:52:53:8b:28:fc:5b:fa:93:54:
36:ca:e1:c1:6b:7f:46:13:de:05:7d:be:33:8d:67:52:ba:6d:
af:4b:ee:01:0b:c7:56:21:7d:16:bd:19:83:90:c8:14:51:8b:
fb:83:c1:a5:ca:69:5a:ae:d9:f1:a7:dc:f7:53:9f:f6:a3:43:
94:fb:38:86:1f:2a:0f:50:cf:8d:bc:36:51:ce:8e:af:80:fe:
b5:80:f8:43:73:ea:3a:d7:a2:a4:b6:73:3a:5a:6b:48:a7:31:
a3:d3:42:3a:fc:2e:b0:29:d2:67:8a:9a:d1:26:95:08:0b:61:
3f:71:ee:b1:96:f4:49:0c:d7:3b:50:61:6c:15:ca:31:31:dc:
0d:fc:d8:5f:a1:26:d3:e2:43:cd:13:39:4a:50:2d:64:57:bf:
02:a8:5c:54:4a:d4:37:45:f2:09:fd:cf:53:67:19:e9:92:a4:
cd:1b:82:09:2c:4d:29:30:80:c1:23:8b:ca:1c:38:c6:11:8f:
a2:3c:2c:7f:86:25:c9:fe:a3:1a:fc:82:ab:69:e9:b5:37:b1:
0e:9a:99:10:cd:a7:b6:52:9f:c6:e4:6e:08:f1:90:cd:14:b8:
c2:e0:a9:58:2e:8a:4c:52:df:d5:ee:8a:57:ce:82:57:a6:89:
0f:74:20:4c:22:1d:02:c9:04:52:68:78:f3:59:c9:c3:60:85:
92:01:30:75:a0:eb:29:2b:66:55:b7:48:4a:df:8f:ba:df:a8:
bc:d9:45:5c:eb:04:a8:c3:94:b6:bb:1d:05:19:48:9b:ae:8d:
63:2d:ba:d6:d3:5e:e5:7a:40:b6:05:74:a1:b0:7a:b7:d7:b4:
67:d6:d6:ac:f5:05:6f:53:45:a6:ed:e0:0c:b3:0c:32:c6:89:
fb:42:7b:11:74:94:25:dc:01:7c:bb:4e:4f:4f:97:54:28:b0:
fb:48:66:87:3a:d0:da:18:bf:aa:13:0c:6a:d3:c7:3e:11:26:
43:e8:40:b3:57:29:00:70:00:af:58:b0:75:83:9e:b9:4b:5b:
39:f1:7f:3f:89:8d:1d:0b:1a:78:4d:e5:8c:e6:07:86:75:23:
1b:14:1f:cd:04:4d:98:d1:cd:f5:4f:1d:00:55:fb:f8:c7:92:
f5:ee:5e:c5:f3:24:84:22:ee:11:48:91:4b:51:f7:87:a8:9c:
a0:9a:48:bc:93:f5:3c:1c:7e:d9:ac:15:1c:1f:b7:f9:b9:66:
```

```
9f:f4:e5:58:4a:f9:7e:5c:3f:a3:5a:20:54:be:57:74:74:65:
80:0d:f4:30:a9:0d:53:e6:71:52:f9:7e:f4:02:24:e5:b4:21:
0b:bc:13:2e:67:00:bd:64:54:8b:82:b4:64:f8:52:46:b2:f2:
37:5d:32:49:8a:be:19:4e:21:a7:cc:9a:19:29:c9:57:aa:fe:
db:4a:ef:e0:a1:06:1a:5f:58:4c:97:ae:fe:ac:16:a0:e3:a7:
60:ef:b6:bf:80:67:35:c8:6c:fe:11:16:18:bd:04:90:32:b6:
75:64:13:55:b2:2e:c6:df:2f:b7:35:d6:3c:f1:ab:4c:1e:da:
c2:4f:fc:24:f2:92:ce:64:dd:ef:70:7a:ae:26:07:01:61:9f:
e6:2e:fe:e4:35:8c:d5:ee:e2:be:fd:3b:8f:c4:dc:5c:50:4c:
5a:2e:aa:14:c4:0e:b5:81:13:55:d0:85:81:16:3d:ce:03:f0:
2b:25:39:b6:f9:ce:ff:c0:f5:4d:77:60:86:03:25:ff:dd:57:
cb:fd:28:fd:e2:8e:bb:7c:fb:49:46:9c:2c:0e:34:74:cf:d2:
b8:45:be:fd:c1:2a:6b:8e:30:48:c3:a7:41:67:04:78:68:9d:
81:1c:35:f4:93:5a:1f:47:ab:3a:34:5e:4e:2d:43:2b:f4:52:
bc:58:34:52:15:53:36:19:c9:b0:bc:57:7c:95:b3:86:ee:7e:
68:9f:73:b2:09:30:4f:f8:90:ae:0b:8d:f4:f4:d1:47:1b:e8:
d1:03:85:92:2d:8a:60:ab:30:f3:ea:26:5e:37:e9:90:b6:2d:
f6:08:1f:bc:fd:13:5a:fd:a9:29:7c:ab:58:10:d9:6d:3b:27:
75:31:f4:74:a8:e8:70:00:a3:63:f1:8c:b4:97:22:2b:d0:f8:
e0:b2:6e:4f:4a:96:d5:f0:3d:fe:73:e1:c8:ba:fb:a8:96:bf:
01:c2:63:70:fa:dd:97:e5:c9:8f:00:04:5d:fa:c0:39:68:ba:
e5:dc:aa:7b:3d:bd:25:aa:43:e2:02:a1:57:2b:78:74:80:f8:
d6:ea:a2:44:7f:1e:35:46:cb:7d:2f:83:dc:7a:25:87:e0:27:
ce:df:12:15:83:b6:26:2a:f9:4e:22:18:ca:69:7d:e3:68:86:
08:40:fa:45:1b:a5:3d:63:a1:aa:19:ca:83:3d:2e:4b:13:4d:
58:26:62:f2:ef:3c:6b:13:cc:99:95:21:c2:c7:f5:af:08:ef:
a0:21:1a:4b:e9:f4:1c:4d:46:72:88:22:8b:aa:b5:dc:fe:3b:
e6:8d:b9:51:8d:45:f4:70:13:68:a2:2b:0a:9c:82:16:64:fc:
3a:5a:2a:19:a6:fe:92:34:65:e2:6a:9c:a5:93:24:21:b4:b6:
50:b8:04:31:02:1c:df:4f:b8:9c:b6:3b:19:66:26:aa:c0:33:
fd:9b:fb:02:2f:c8:07:8c:1f:66:8a:f6:f3:c5:0b:74:ce:75:
c4:94:34:80:60:53:c1:42:09:2d:21:fb:25:b4:ff:c1:00:30:
f1:c8:ad:ce:62:c6:1d:d7:94:cc:0f:7b:2a:00:be:b3:f3:c8:
3f:e5:88:af:6d:19:90:31:71:96:d6:8c:5b:34:b8:85:b5:42:
f2:fb:17:a0:83:bb:6a:61:86:f0:ef:1f:db:ce:00:2f:90:aa:
ee:07:97:59:56:85:96:1c:97:6b:ca:d4:7d:9a:bd:dc:01:52:
dd:1c:bc:82:5e:81:08:91:36:85:7f:3e:12:63:59:aa:03:10:
b3:03:2d:ad:17:7d:61:91:d6:e1:b9:2e:39:54:27:8a:a4:91:
87:ba:33:54:28:52:0d:46:f0:e7:63:40:6d:15:76:11:51:28:
1b:5f:94:ea:30:6f:00:34:a6:d8:42:c4:32:a0:36:1b:55:04:
90:87:8e:2e:04:47:f1:25:c8:fb:d4:58:79:36:5c:b9:81:18:
c5:ff:16:ab:fe:b8:01:0a:fb:4a:93:3d:9b:c5:82:d5:1f:bf:
95:ea:aa:36:ef:c5:f8:d8:ab:f7:ca:c8:49:dc:30:fb:34:9d:
81:e2:7c:6c:06:78:34:a9:aa:44:74:9f:42:a5:c5:91:9f:41:
c4:f1:79:7e:0d:cd:36:d5:21:32:5d:82:4d:b3:80:0d:72:19:
ab:2a:0e:de:f4:22:ce:48:b7:b2:44:02:f1:99:b1:bf:79:dd:
49:0b:bf:3e:f8:b9:a5:e3:28:8d:8f:89:b3:d8:bc:97:cb:2e:
f8:c0:8f:f0:10:cd:00:2f:df:bc:bb:ab:e0:77:de:d9:44:17:
8e:70:f0:07:e1:9d:c5:a5:fb:91:ee:3d:ee:f4:98:9d:67:10:
04:3a:a6:f2:03:fc:e8:05:53:ee:00:29:3c:84:ff:35:f4:df:
93:74:82:16:ec:58:25:43:81:01:b2:68:d2:a7:51:ed:97:ed:
c2:06:1e:eb:8d:75:cf:11:30:b0:f7:0f:c1:d2:c1:f1:43:5d:
42:70:fa:c1:f9:2a:eb:a2:af:00:07:cb:99:ca:cb:9a:50:85:
c3:63:76:d3:ad:f5:ef:d4:f0:c9:75:a4:4b:88:4b:32:81:c3:
43:97:bf:a8:0b:c0:5a:23:b4:28:46:4c:04:70:36:88:ee:eb:
f5:26:b2:99:05:cc:6b:0a:0e:f9:06:73:fd:c3:be:37:c7:26:
29:11:62:d4:20:e0:06:f2:68:c3:57:db:bf:85:e6:2f:cb:f1:
81:96:88:70:9e:a2:6a:42:02:fc:79:90:f6:c9:b0:fb:b3:6e:
```

```
a5:68:c4:ee:bb:8c:87:6c:81:20:15:a8:7f:1b:ba:f7:2e:b2:
f7:5f:a3:c0:03:44:ce:e2:27:f2:04:d0:c0:b2:7d:be:b3:11:
4e:e9:77:7c:be:83:94:03:13:75:2f:c4:d4:8a:e9:bc:a3:fa:
6d:5c:72:fa:62:86:17:e2:db:97:88:ca:6c:4c:ad:68:2b:57:
cf:f5:b6:92:2e:02:2e:82:d1:5c:9f:3b:8e:e9:e5:8d:76:7c:
65:9d:57:e5:2b:df:c9:ca:b1:8c:ec:86:e7:09:95:de:73:57:
4e:ec:af:62:47:45:79:c6:fd:09:32:d9:5b:73:de:67:44:39:
28:a3:ff:1d:8f:22:61:04:48:84:fb:f0:44:04:0f:01:1b:ad:
bf:9f:ff:34:2c:83:3d:d6:85:3c:9b:82:ef:47:c7:ab:a2:e2:
9e:ac:71:eb:d6:5e:a7:d8:e0:79:53:39:29:15:0e:a6:b9:56:
39:93:16:7f:0a:48:00:6d:36:0a:2a:4a:11:ef:80:d7:43:c4:
f0:06:e2:a2:49:9a:e6:2d:c5:fd:46:96:a8:83:45:22:b5:c7:
55:dc:cf:3f:84:8e:0b:69:7c:dc:e0:30:1a:1f:a6:14:d6:42:
d3:0f:91:4b:6c:3f:2f:f9:64:25:bb:e4:83:b9:44:80:b3:6c:
c7:f2:3e:58:a3:61:7a:1a:04:61:d8:a2:8c:e7:43:d7:eb:f4:
90:48:90:30:dc:c1:55:b3:eb:4b:68:09:af:62:79:d7:f6:09:
61:89:b7:6b:37:3e:09:4e:d5:d7:e3:05:b1:4b:f0:e5:1f:6b:
3e:f0:6b:eb:2a:8d:1d:ae:f6:87:c6:70:f2:74:fa:92:46:1d:
d6:7e:d6:ab:1a:d3:de:11:71:be:f0:a1:e3:05:82:4e:3a:a1:
2e:d2:2b:c4:92:0e:a3:70:10:3f:df:c4:cc:52:97:f7:4c:a6:
5a:7b:cc:e8:74:5a:47:12:42:73:d8:5b:09:7e:31:a9:68:33:
77:f6:d1:72:72:a3:22:e2:d9:6e:c5:fc:f2:30:d5:85:c5:c2:
50:79:10:a6:9f:15:50:31:a4:87:d7:cb:da:b9:5f:37:ab:fe:
7f:09:25:e5:c3:1e:c0:d6:78:20:a0:21:20:10:6f:3c:d0:bd:
46:fe:bc:ad:df:25:27:8d:f4:0d:0c:4d:b2:30:b1:70:8e:aa:
25:9f:80:b9:60:b7:79:b2:25:be:a5:df:ee:ed:8c:ac:87:c9:
69:3f:ea:e5:cf:4d:d1:44:73:7f:a7:4e:9b:69:64:df:da:8a:
57:53:11:0e:54:fd:af:ca:4c:6d:e0:ad:56:1f:7f:c5:07:00:
8b:e4:b3:09:53:af:a4:db:e1:a1:c4:e1:c0:d6:70:d4:2d:e8:
d4:bd:38:94:c7:93:39:64:71:50:6d:a5:30:7d:fe:1e:61:d0:
a1:26:bb:6a:f8:32:63:05:37:65:bb:23:97:06:13:c6:d6:46:
b5:83:fd:d3:9b:a3:94:ec:67:8e:9c:bb:9e:af:0b:df:e8:28:
ed:45:ff:a4:8c:d9:f9:e3:30:dd:20:f2:3d:ad:4f:d0:b9:2b:
17:bf:d0:4a:8e:03:8d:a2:1f:16:fa:fe:87:eb:3c:57:7d:f8:
78:f9:2d:74:d4:82:d8:53:e0:91:b6:83:6f:73:79:ca:d9:ca:
83:ed:84:75:10:e0:5e:fa:a7:0f:a1:9b:67:21:d0:9a:b0:90:
83:68:3c:99:97:69:42:11:2c:51:b9:6f:5c:03:1f:2e:ee:78:
b7:3a:14:db:d8:9d:17:69:9a:ad:9e:80:d5:d7:de:fe:3b:18:
ee:a6:7d:9f:3b:6f:30:67:74:a1:f4:ff:fb:68:ad:e4:ec:8f:
7f:5b:02:46:62:26:10:6a:88:b1:a7:89:d1:87:00:a4:95:84:
96:9e:b4:1f:bf:f1:6f:67:b6:3f:d5:c2:5c:1f:41:10:cd:06:
a5:e8:fe:e2:1e:52:e3:5c:46:b9:c4:e9:18:aa:78:e0:4b:78:
82:78:ac:3d:59:fd:24:40:44:01:d6:ad:6b:87:bd:11:a1:c1:
bd:f2:a9:cc:be:ae:05:52:7b:bd:86:63:d6:9e:bd:52:3c:25:
dc:a4:bb:73:bc:0c:04:04:c1:0c:e9:6e:d1:26:c3:50:ac:98:
fb:4b:49:c5:69:ed:d8:30:bb:7c:d2:6e:d3:76:5a:13:0c:82:
28:cf:40:5c:0e:16:24:e8:82:5d:2a:f0:87:89:23:99:2d:7e:
6a:85:a1:dd:ab:78:1b:e6:cf:76:bc:fe:26:b2:26:a5:a7:e1:
d4:44:a3:ff:20:ad:84:73:5b:26:b2:3a:15:c9:c4:02:9d:fb:
b2:2b:cf:b5:f2:a3:7e:99:de:f9:d9:93:f7:8b:16:e3:04:4f:
c4:bc:4d:67:9b:3f:ba:2d:79:7a:47:f1:ea:d8:36:cf:5d:eb:
f7:b3:ae:0c:e0:62:f8:f6:2c:d0:29:91:8a:fa:68:bf:20:57:
ef:79:0d:71:62:f7:a7:25:c7:77:f2:03:48:2d:95:73:7b:ba:
c0:f5:62:7b:bb:0d:06:b6:88:74:a4:b4:7e:48:b9:a6:6d:92:
78:3d:87:4e:68:44:d6:45:23:c9:7b:04:02:7e:c7:40:7f:a0:
41:fc:24:8e:e5:43:19:f4:65:b2:a5:e7:73:27:03:b4:52:0e:
de:33:12:62:ed:b6:c3:2b:19:cd:a0:69:0b:cb:63:eb:85:83:
a1:16:a9:2b:72:c1:e7:c6:63:7f:a4:41:6e:19:61:3b:78:ba:
```

```
db:6a:18:5c:f4:b1:5d:a5:5d:df:38:fd:5f:80:cf:cf:f0:95:
e1:b1:bc:7a:2e:2c:ff:04:00:5e:c7:79:1c:47:e0:a7:57:de:
1b:e6:69:13:7a:3b:cf:a0:d8:69:16:f2:9e:45:e6:b1:7d:9f:
f7:47:25:d9:1f:50:0a:6e:dd:da:53:e0:4d:52:91:33:87:8a:
3f:37:ef:7a:eb:1a:98:a0:55:e0:f9:e5:f2:03:1f:e2:eb:e5:
30:6c:0c:4b:75:a4:cf:40:87:da:30:49:25:e1:25:fd:38:ce:
44:20:e3:75:7f:25:2b:7b:dd:b2:02:d7:e2:0f:96:a4:bb:cf:
0c:df:16:e7:5b:91:46:31:bc:4d:18:b6:ca:33:a1:5b:e6:70:
95:03:40:79:a9:12:a9:1d:09:e8:38:d7:d4:7d:c3:a8:25:6c:
c2:aa:0b:78:19:5b:16:cb:8a:24:4f:b2:7a:ca:87:68:85:9b:
22:17:50:ea:fd:28:ae:45:f7:b6:ba:76:de:49:ce:9f:a4:48:
b1:bb:f1:ba:f8:88:8e:14:1e:2f:2d:53:79:bf:32:0e:fc:19:
20:b1:ba:12:68:5d:8c:d8:3c:3c:d6:63:8a:2e:8b:e4:7c:75:
05:27:a8:e9:e0:5b:be:87:77:d5:b3:88:74:db:cd:5f:59:10:
5c:9c:44:e1:d4:7d:bf:36:ec:fb:70:95:bf:a7:1b:d9:a8:ee:
fd:d7:91:4d:72:b1:d1:72:87:0b:02:58:22:23:cb:b1:72:36:
04:47:33:a6:39:99:34:fa:73:6a:e1:b9:21:17:7a:04:5b:23:
64:65:9f:bf:14:e6:8d:4e:70:1b:9e:19:af:9b:98:3e:6f:13:
2e:35:a5:90:a7:c6:24:8a:b6:d0:0a:a1:60:eb:40:cf:7b:c5:
03:87:e2:a7:76:8a:10:5b:4e:75:c1:3e:ad:37:1e:ff:46:59:
a8:b1:6e:c4:fe:65:81:61:67:6d:83:51:9f:22:58:1f:a2:e1:
39:dd:d4:33:74:22:90:cb:93:bf:65:a6:5a:8d:92:db:9e:9a:
60:1e:96:5f:5d:66:13:b8:f3:82:fb:13:5a:ea:3c:e9:1f:5d:
d7:b4:7f:18:99:38:d3:1e:49:83:26:a8:ec:c0:13:98:af:a2:
cf:2d:2a:4a:4a:7e:32:fc:20:b5:84:c0:2f:d6:0c:40:5a:ad:
34:db:fc:d5:f3:8c:5e:ce:cd:15:fb:68:d4:60:c4:0e:fa:9c:
f1:7e:0b:c2:95:cf:e1:1f:6b:4b:b4:8b:7d:1b:05:45:8e:65:
62:d8:24:4f:c9:31:f5:9e:1b:3a:d3:cd:47:05:93:e0:91:89:
9f:7e:87:50:a9:0a:4b:28:df:00:55:01:7f:58:f6:d4:8a:17:
c2:60:1a:56:2a:49:9c:8d:11:25:7e:42:e7:60:90:20:f7:3e:
12:25:7b:82:05:49:d5:2f:88:cf:73:db:09:7e:0f:f1:7d:c6:
a4:0f:dc:3d:5f:25:a4:2b:e1:74:7d:70:5a:a5:b4:67:6c:66:
74:c4:86:01:30:af:d5:e9:fa:49:72:38:3b:00:95:de:fb:c6:
ae:ee:c8:d0:af:b2:14:8f:9d:da:32:5f:9e:e7:85:76:a9:1a:
7c:d3:69:8b:02:4b:3c:ff:51:3b:a0:80:69:f0:95:01:10:ae:
ba:94:a9:59:ce:a0:90:af:8d:f5:db:45:63:0b:4f:8a:fb:96:
db:26:66:da:b8:e2:cf:7e:15:47:c8:10:03:46:8c:3b:bf:46:
0c:29:e6:7d:80:42:3a:c2:8d:38:b4:48:2d:2c:96:a1:37:71:
13:9c:72:00:02:ff:a4:79:ff:74:5a:31:ba:a6:3a:24:08:bf:
8e:41:b4:48:6f:bc:43:85:31:7d:b9:ca:06:60:76:fb:a7:d1:
a3:af:ad:d0:a7:cb:07:02:08:ba:b7:ce:ab:06:56:28:5d:31:
79:2c:db:10:52:55:4c:65:53:10:ce:1e:5f:0e:e5:15:25:c4:
e0:78:12:3c:d2:0c:89:f3:60:dd:f1:ef:8b:ec:7e:8a:9b:2c:
58:9b:1f:7b:f0:d3:dd:47:d7:49:5f:11:fa:ed:7a:72:1c:84:
6c:06:0f:76:44:a8:e6:2f:24:1b:3f:66:46:3c:e7:c6:7f:e3:
06:1b:5e:7c:e6:d6:67:08:34:f3:64:2c:fd:30:9d:d8:e2:75:
14:95:91:d0:0f:4c:d9:f0:95:43:42:b2:15:db:4f:3d:15:cb:
60:6c:22:f8:fb:e0:c4:43:1c:d0:71:9d:10:9b:f6:76:c3:d4:
e8:f1:d8:62:b3:b3:8f:f4:e2:69:a5:fd:e3:0a:23:e6:4e:9b:
0f:a5:2c:a1:09:01:ce:27:26:94:a7:90:c0:e8:0e:82:98:43:
44:87:9d:34:57:73:b5:b7:35:fa:a3:af:47:cf:09:48:27:79:
d3:c6:1b:04:7a:08:df:a6:78:0f:6a:2e:5c:e5:c6:a6:16:ac:
4f:4d:6d:06:d6:45:de:68:3a:2c:f2:22:32:61:8c:e6:d0:e5:
62:a9:49:fe:ba:86:ad:cb:c6:be:29:6b:0b:4b:cd:4c:59:4e:
bd:17:6c:9b:c9:d6:d9:cd:9f:aa:01:8c:c9:a3:dd:af:6b:5f:
e9:f5:18:24:6d:90:e1:14:9e:56:86:04:2e:3b:a2:42:21:f8:
0a:ee:05:71:31:55:f7:56:99:5f:72:18:87:22:ff:6d:4f:7c:
c2:c2:32:84:5d:4c:1d:da:59:12:71:48:98:37:68:c8:6c:14:
```

```
8c:b6:8c:d4:49:e5:f6:2b:0f:04:ac:66:1b:f7:c4:d0:18:6d:
e3:5d:12:4d:9d:34:c6:4c:36:cf:96:2b:5d:ae:d7:b1:74:c9:
f0:44:b6:f0:c6:45:32:4e:b7:42:42:d3:f9:b5:c3:51:54:3e:
b8:4a:70:0e:82:2e:39:07:bc:66:a9:91:93:43:f2:7f:ed:a4:
61:f2:35:fa:e0:9f:86:00:c9:87:5b:69:7e:3b:f8:d1:fa:e7:
78:e6:d0:46:27:d5:80:d4:34:0f:8f:bf:1c:27:47:60:3f:a7:
b5:c4:ed:b3:c2:15:37:37:b3:8b:d1:c1:a7:1b:47:24:73:ce:
22:74:da:fb:c8:3f:a1:65:4d:79:67:d1:8a:db:71:79:d4:5d:
7d:a1:ae:05:93:78:31:98:d3:f6:cc:a3:42:93:e1:11:06:51:
2c:3c:4c:b7:6b:5d:07:fa:a8:08:72:4c:9a:26:0b:af:28:1c:
70:55:b1:1d:c8:82:98:3d:a5:b4:62:ff:77:07:13:84:b0:10:
7e:f3:33:37:21:41:2e:cd:3b:da:4e:e6:fa:ad:3f:ee:f3:05:
39:8d:65:20:dc:94:49:98:e4:e9:a1:26:b3:3a:3d:c9:69:1f:
e4:9c:29:7d:1b:91:02:70:27:8b:77:df:18:7e:50:50:58:06:
1b:fc:37:6b:4c:00:71:ea:ee:82:4c:e2:8b:a4:a7:81:f8:87:
57:07:50:d9:d0:bf:f4:85:c7:4f:9b:cf:e4:51:ee:d1:6b:0a:
a3:a7:79:a9:7f:e4:6a:eb:83:59:82:f8:e5:32:c6:6b:93:57:
18:61:e7:89:b1:ff:a7:f7:31:8b:54:31:df:30:c8:0b:2f:7e:
5c:4d:1d:99:e2:cd:61:97:b5:28:14:36:3f:36:0e:b4:27:38:
c8:61:68:e0:95:8d:26:3c:d4:83:5d:96:9f:a6:37:96:59:db:
10:a4:5f:90:b6:44:f1:7e:6c:86:44:25:40:0a:fc:ef:d7:5c:
97:ba:1b:4c:95:9e:e3:9e:90:b9:02:58:30:1d:60:b7:94:30:
f5:78:b5:a4:ea:37:82:7a:f5:73:6c:0d:d3:81:ca:72:cc:8c:
cd:bf:6f:fa:7f:cb:39:27:1a:59:9a:71:51:d8:f3:b3:40:d3:
da:66:83:f4:f2:94:a5:8f:b5:a0:7f:72:c2:c8:e7:1b:41:36:
fe:fb:6d:81:d8:ab:8a:33:41:18:bf:42:c9:1a:8a:22:fa:25:
9e:e0:b7:45:46:ee:ab:3b:57:3a:8f:64:96:51:7a:1f:66:95:
f9:52:95:40:77:51:69:f5:6e:bd:3c:97:95:53:90:09:b0:fc:
5f:8c:ca:d5:2d:40:ab:29:c2:21:31:80:75:b9:0c:c9:57:46:
f9:7e:e1:fc:95:63:c1:91:ad:10:90:af:2d:a2:85:02:55:d1:
a1:10:76:db:24:ac:37:1d:35:bf:8a:09:29:21:b7:da:d5:26:
6d:00:6e:77:3f:64:e0:88:6b:09:37:e9:82:f8:c7:ad:bc:05:
ea:1d:75:a4:ba:c3:d4:fb:43:ae:99:28:3a:19:fd:84:53:4b:
84:8a:b3:76:ae:a6:dd:a9:bb:fe:56:c2:7d:14:05:62:3a:a4:
af:7d:3b:cd:80:c4:dd:87:58:54:21:9e:21:f2:60:a3:42:a6:
de:55:31:8e:c9:7c:01:ae:fd:87:67:52:43:ba:7a:a4:ee:23:
9f:6f:0a:52:db:38:12:41:18:c4:2d:4a:85:84:36:59:a6:23:
9e:38:8e:51:c2:88:23:85:3a:dc:60:52:56:79:99:84:b0:a5:
a9:b3:1b:ac:27:c8:5d:4d:82:8d:3c:ee:e7:84:c7:0d:72:ac:
80:c8:82:55:bb:05:7b:1e:33:f4:a3:0c:39:5b:2b:ed:a4:f6:
cf:a5:15:8f:58:be:a0:bb:9b:35:27:cc:7b:78:aa:ee:ab:0f:
fa:de:aa:bb:95:94:37:b6:44:ff:21:e1:64:41:73:46:22:d9:
b0:89:61:24:b4:53:01:99:17:4b:79:e9:dd:e0:3d:0a:c9:3d:
d5:02:1c:49:4e:bd:26:d9:9b:b0:32:2e:6a:22:b8:70:f5:c6:
ed:51:4f:ee:a0:37:29:75:f3:17:5d:35:d2:a6:3b:71:43:8b:
6f:22:9b:1a:7d:a0:c5:f7:7f:7e:24:7a:93:67:b9:0b:4c:84:
61:f2:dd:6d:6f:60:7b:63:56:47:c6:cd:1c:ae:25:18:a9:cf:
21:aa:bc:d5:70:48:75:38:a7:10:5e:bc:bc:a1:e0:27:4f:6c:
18:b4:40:f8:80:01:74:1f:fc:d2:82:58:b3:c4:f3:1c:f1:e5:
66:61:c0:6c:63:4c:3b:b6:61:7a:15:9d:be:75:4b:c3:04:35:
a3:a7:03:f9:cc:50:62:d0:38:74:c1:e2:c8:ce:46:1b:76:42:
a0:3b:ff:5c:3c:04:c7:73:3d:ab:36:b4:1c:ef:47:7e:99:79:
0c:87:9d:54:c9:45:4a:61:29:43:34:72:4e:a6:d9:24:2c:30:
74:75:3d:16:87:91:03:58:3e:79:3b:f3:d1:8b:6a:10:87:18:
92:c9:0d:e5:aa:63:45:0a:60:83:c2:81:11:38:b6:c3:cd:f8:
b0:71:d8:e0:5b:04:c5:57:2a:55:3c:db:3f:82:26:eb:db:09:
b7:0b:f2:68:90:34:be:79:41:25:97:9d:d1:97:0e:af:4c:ae:
40:21:61:5e:f3:be:99:da:a3:82:31:98:96:5b:1c:86:20:48:
```

```
6b:af:92:df:e7:2d:f5:0d:97:55:04:4b:3d:6f:10:47:98:69:
f3:06:8b:a0:9a:88:7c:0a:a2:84:8d:71:4a:5f:23:74:2e:ed:
bb:28:32:d2:33:34:ab:77:40:e7:f8:d4:16:fe:b0:73:e4:14:
a5:f5:3c:3e:a0:f0:e0:42:1d:cf:c3:c3:f8:bb:07:5a:56:20:
6d:4f:8e:ac:63:f6:3c:fd:f6:11:2b:97:2c:86:66:66:11:16:
eb:51:c2:29:06:30:84:ba:e4:81:98:56:68:70:43:31:5d:c2:
ef:eb:e6:e5:86:cb:9b:e3:37:8e:a3:fa:ad:46:cd:63:9d:d2:
a1:6d:5d:df:65:cf:7c:39:cd:24:ae:86:40:b0:3f:d3:77:1d:
58:54:4a:11:b9:7d:25:c0:88:79:d7:36:c7:aa:2c:d8:3f:db:
86:82:ff:f9:0f:22:d0:5a:71:8c:5b:b2:23:ea:ca:cb:ee:b6:
51:2d:5e:43:da:fd:18:84:47:22:95:31:e0:e5:68:2d:65:6b:
0f:f9:94:40:e8:45:4d:16:d0:6b:ac:57:24:de:e2:c1:eb:99:
65:91:9e:7a:6c:6c:6e:c7:37:ab:2e:4e:80:80:09:60:d5:10:
0b:51:9b:24:7f:20:b2:7d:77:b5:e1:33:a2:2e:c0:7a:62:fb:
aa:bc:a8:ba:07:ef:27:c4:69:c0:4b:da:ff:89:80:13:82:1f:
25:59:3b:40:dc:11:f4:5d:de:c5:a4:a0:d5:47:c0:19:ed:1e:
d3:67:4a:b0:76:db:85:2d:df:4f:eb:6e:17:ac:9e:cc:67:0d:
74:03:10:5b:88:d3:de:c7:e0:05:55:48:01:bc:be:7a:82:2c:
fb:5e:3d:f7:ca:2c:42:20:ed:50:ff:3c:2b:07:c4:8d:d1:13:
57:aa:26:67:83:02:1b:79:88:04:c5:ef:0a:6e:c8:f8:a4:cd:
93:57:bb:4a:39:4b:9e:c1:17:67:54:9f:85:5e:8b:a4:15:f3:
81:ba:2d:85:64:a8:99:ea:11:0c:9b:83:52:80:03:18:c0:1d:
72:9e:d2:0b:d4:8c:e5:59:08:28:a5:cf:8b:46:ef:e9:82:9b:
54:f0:e2:09:70:b4:2d:f4:31:d1:f1:ea:da:57:1c:1b:bb:de:
b3:85:47:f4:19:e4:c4:06:85:87:54:23:76:6c:e1:3d:28:c1:
c0:25:00:b3:34:d3:51:af:d9:df:0f:8b:b8:b5:6d:c8:53:fe:
8d:59:ba:f1:0e:00:05:4e:bf:51:9b:59:10:59:07:0f:5f:27:
99:9f:7c:6b:a3:14:40:32:da:e4:89:8d:b5:c6:d3:3f:ed:e3:
f9:2d:15:ac:d1:a8:11:41:2d:2c:72:ab:a4:d5:f4:9c:ae:d7:
af:7d:39:e2:1c:8f:a8:ff:3e:92:7d:e4:76:38:d4:fe:a2:99:
6e:1d:6b:11:70:e3:de:f2:4d:1f:4d:e5:cc:44:43:f8:42:c8:
99:11:c6:29:22:ee:f9:13:d5:08:15:71:fc:0e:ca:82:97:b1:
11:fb:b9:8c:27:3c:be:a4:d7:d8:4f:3c:0d:3c:82:5d:cf:18:
01:09:28:ca:1d:f0:f7:ba:71:80:eb:76:7a:58:e9:91:b8:86:
71:d0:71:d2:13:3c:b7:65:e7:c4:ff:27:f7:2f:f2:3f:24:d5:
c6:df:6c:d0:dd:0a:ee:de:4b:16:66:6f:68:ce:94:b1:f9:69:
67:0c:c4:19:20:2c:29:74:f8:a7:e2:00:06:13:c9:2d:1d:4f:
76:74:03:28:46:79:b7:80:b2:da:d2:39:0a:56:47:5f:c3:81:
9a:ee:17:91:0d:49:f4:23:3f:36:db:55:48:d8:16:43:ff:6c:
6f:fa:ca:ac:17:ca:a3:62:4d:de:60:5c:ed:f5:a3:96:33:35:
53:24:06:99:8f:30:d6:a4:b8:07:3d:e1:d9:ca:07:9b:54:70:
50:c6:0e:d2:4b:93:9c:07:16:b7:9e:1e:d7:42:8c:c6:fd:41:
cd:aa:4e:fc:2c:11:1a:6e:00:db:5b:25:6e:96:c8:29:43:ac:
68:be:c0:d3:2c:3c:1b:d4:b6:9c:2a:a0:9f:9b:16:a3:2a:dd:
ed:00:2c:b9:9d:93:59:65:81:de:a9:a9:b8:96:ac:c4:43:30:
93:21:4c:3c:42:06:8e:ab:fa:37:96:72:c8:ec:22:19:1b:8b:
ca:22:73:be:08:df:6a:1d:d7:ef:13:0b:43:ae:fd:a0:d6:a1:
10:8a:f7:5e:13:e5:5d:a1:81:c0:81:06:3f:5f:ea:b3:e1:78:
99:f5:2d:1c:56:0b:df:c3:1d:4e:1f:f6:ea:22:9e:d8:33:13:
2b:bb:e9:3f:b1:17:cf:33:0e:80:85:72:72:72:c0:ad:70:b4:
81:9b:d8:57:d6:a4:9f:f7:92:15:e3:72:d0:ee:22:a1:47:b0:
90:e3:f1:14:b6:99:ff:fc:c3:cb:34:03:f8:00:76:dd:7d:c4:
4d:1d:c2:eb:48:73:4d:41:40:9d:e1:80:5c:37:cc:65:a7:6a:
8a:b0:9a:35:d5:2c:cc:f3:a3:cd:43:f7:e7:5c:46:7a:e1:5f:
b2:a0:93:d7:00:ca:9e:3a:15:4c:61:ab:fc:62:e4:39:79:d6:
22:2a:d9:7e:8f:a4:65:1a:e9:1d:89:2b:9c:ef:d7:3f:36:fc:
93:9c:ec:e5:a6:93:ce:ec:32:91:48:46:b0:0a:b2:e3:33:19:
df:a1:fb:78:20:e3:13:54:13:f3:fb:8a:5a:f2:9e:ba:34:e1:
```

```
fe:eb:58:e2:c4:af:b6:63:56:32:42:cf:e3:7d:c5:f0:d5:6f:
f6:64:53:40:17:c0:88:f0:54:8d:9c:05:8d:52:39:63:68:23:
86:86:91:34:f2:9c:a4:dd:17:ba:26:5a:7f:73:77:19:5b:93:
5a:2c:89:07:5f:27:45:2b:aa:86:1a:98:98:59:2a:46:c8:8e:
4f:75:30:dc:3a:e9:f6:1f:c0:33:ef:0a:13:30:5c:32:45:88:
19:67:4e:4d:a8:f1:fa:89:b0:ef:e4:42:3e:26:60:80:93:21:
7b:46:b9:f4:6c:be:9f:c6:7f:c6:49:c9:e1:49:c8:2d:07:36:
93:69:14:18:e3:fb:3b:6b:79:37:00:bd:f2:e1:f6:06:7b:2c:
07:ea:86:e2:1e:62:64:48:43:59:7d:2f:fd:24:c8:a1:4f:94:
ac:8d:1e:7d:15:a1:32:01:25:ba:3f:35:d6:16:57:24:28:f6:
68:35:d3:80:21:cc:91:76:bd:15:7f:a1:42:6b:8e:a5:90:7b:
fa:5d:01:7a:2e:02:21:b4:31:f9:2c:40:88:34:75:01:cb:83:
39:1b:3c:38:a2:c2:5d:33:e3:83:55:7f:fa:f0:d7:cf:c9:64:
9f:06:39:b2:18:f3:41:81:60:ff:50:5d:50:12:37:0e:82:c0:
da:2f:6a:f8:fc:16:5f:bb:22:29:83:14:46:a4:01:ca:f8:d8:
2c:79:ed:cf:40:37:46:a8:48:7f:66:7d:0e:a0:ff:2f:07:c0:
a3:58:ec:2c:3a:27:33:e3:3f:52:ac:94:99:10:2b:15:84:11:
e9:71:c0:35:c3:79:f7:25:bf:f3:5b:42:46:17:44:5d:c1:c4:
ac:fc:01:60:6a:69:5d:cc:65:08:e0:31:c0:db:01:ed:78:70:
18:1b:93:af:f7:b1:2c:0b:1f:b5:68:96:b8:f9:69:9f:e5:e6:
35:cb:bc:06:65:64:11:d5:ab:d4:e6:d3:79:31:a1:b0:e2:d3:
80:78:c2:f6:87:74:e3:34:48:ab:8b:5e:30:52:d6:3b:02:72:
cd:3e:a4:f9:da:ca:6d:da:6c:59:07:39:73:da:08:f0:d0:3c:
9d:f9:52:83:77:60:67:58:9f:67:11:24:13:f4:86:86:8d:29:
89:c5:4e:86:22:12:86:11:94:0e:f4:c6:26:3e:0f:8e:06:8d:
5a:60:30:d0:a9:a8:bf:76:3f:88:34:79:a8:da:78:1b:71:9f:
8c:33:59:8d:fb:6b:cf:96:45:4f:be:54:e5:15:c6:d3:9b:7d:
ea:d9:61:53:75:91:3d:c5:10:7d:a2:5d:00:cd:4a:77:ba:96:
6c:51:57:a4:68:75:43:27:ec:0b:49:4a:4d:25:c9:38:fd:cc:
33:1b:da:70:bf:1b:c3:d4:59:dd:8a:05:fe:87:c5:8e:59:16:
ef:33:4b:88:14:f4:8e:3f:65:43:eb:ea:a3:9c:5c:eb:dc:81:
d7:df:7b:a5:1e:4d:84:5c:cd:31:e2:02:a6:37:cf:81:4f:b5:
91:41:87:04:92:f3:c1:5d:62:2e:52:f1:86:ae:8d:13:bf:b6:
c7:56:36:ef:e6:97:b6:05:cc:39:db:49:af:b5:3e:ec:ca:37:
2e:a4:51:c6:d7:03:2d:c8:69:3b:58:f7:91:ed:d4:88:0e:9c:
05:7f:fe:8c:5f:0c:18:31:39:4b:ad:3c:25:4d:26:24:42:45:
99:18:df:0e:ac:93:47:0b:47:60:58:53:63:0f:0b:b0:67:a5:
07:12:ca:a1:64:e9:a3:be:16:de:f6:70:8e:23:8d:61:d7:8d:
4b:31:6f:79:48:8c:b0:be:01:48:f2:4e:3d:2a:4f:e0:55:90:
72:3e:d3:0c:5c:f7:f8:15:45:e4:10:df:ad:9c:d0:23:c3:bb:
a3:52:70:08:e2:fa:ae:ba:b0:74:35:dd:a6:4b:fb:9a:b7:3c:
28:17:87:08:70:47:42:5e:58:3a:a6:84:ac:94:34:41:5c:3c:
d1:ac:0a:b4:bf:a1:c6:da:c2:59:a3:22:cc:a6:e3:e9:d5:92:
15:80:bb:2e:24:91:d3:8a:02:13:e5:51:05:f5:55:4a:78:41:
d5:e7:62:1d:b7:d5:1f:e5:34:f7:b1:ae:c6:0f:ec:38:c2:a8:
23:8e:ff:5d:b6:87:8a:4f:bf:77:d6:c1:ae:a1:c8:88:d5:66:
e1:77:06:ca:91:10:db:14:20:4c:a0:8f:d8:8b:1b:71:66:b8:
96:09:08:6a:ec:df:c1:4b:d6:91:03:8c:66:e2:c8:1d:c9:0e:
f3:99:3e:0a:b4:60:83:8a:bc:3d:ca:19:00:b3:fd:b0:5e:84:
61:b7:23:04:db:64:35:06:9a:ab:4a:03:47:a2:79:6c:d8:0b:
9e:c9:77:bb:47:5e:db:66:e4:f3:33:eb:8c:e2:49:a4:d6:a1:
c9:61:97:4a:e6:3a:ab:16:64:b3:df:16:5a:de:e5:f9:ba:5d:
7d:eb:04:f5:f4:f0:f0:7d:e4:1a:74:fc:7d:03:16:a4:ca:f6:
e0:05:95:e0:fa:9d:80:07:58:b4:12:5e:34:43:04:ad:90:9f:
3f:be:31:ca:3d:d3:c9:d0:b7:91:c7:5c:d0:2b:81:73:34:bf:
ca:a5:6e:23:4f:b3:f3:b4:bf:03:f4:bd:af:fd:d7:09:8b:65:
a3:0c:76:dc:1e:7c:97:d2:be:85:d4:65:6d:f9:3d:6e:ae:6c:
57:f4:10:40:21:d6:04:2d:9b:9b:e5:95:90:9c:52:a8:ad:61:
```

```
8b:cd:b0:12:c1:13:26:c3:4d:8e:22:82:82:9b:fe:6d:01:e7:
3c:65:79:b4:79:9f:9e:b0:10:dd:5e:6a:57:43:8c:6b:41:d5:
e6:ab:94:ba:c7:67:a5:b4:41:d8:10:0c:fd:29:77:e2:0b:cd:
29:80:2e:ae:5e:a5:85:a3:a2:09:31:51:82:98:0b:2c:7a:6b:
96:ef:8d:c0:f5:1f:98:b4:f6:22:b6:21:6e:36:e3:bb:18:da:
1d:24:46:0d:65:28:b6:6a
```

-----BEGIN CERTIFICATE-----

MIIGLTCCAWegAwIBAgIUQ4VjomkBMsw5z7xAVxtfo8zHiEUwCwYJYIZIAWUDBAMU  
MEIxCzAJBgNVBAYTAkZSMQ4wDAYDVQQHDAVQYXJpczEjMCEGA1UECgwaQm9ndXMg  
U0xILURTQS1TSEeYLTeyOHMgQ0EwHhcNMjQxMDE2MTM0MjEyWhcNMzQxMDE0MTM0  
MjEyWjBCMQswCQYDVQQGEwJGUjEOMAwGA1UEBwwFUGFyaXNzAhBgNVBAoMGkVj  
Z3VzIFNMSC1EU0EtU0hBMi0xMjhzIENBMDAwCwYJYIZIAWUDBAMUAYEAK4EJ7Hd8  
qk4fAkzPz5SXZ2GAUJKA9CVq8rB6+AKJtJSjYzBhMB0GA1UdDgQWBbTNWtaq/sQR  
x6RyaT8L6L0LIXsZ7TAfBgNVHSMEGDAWgBTNWtaq/sQRx6RyaT8L6L0LIXsZ7TAP  
BgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAWIBBjALBglghkgBZQMEAxQDgh6x  
AKqgUd6wwwTQzfsSRqIXIMntqz/cV6X7Rfbw03/jWoy1hx4fCxWfq1ZoQ37qIwUh  
0TPLhGFVfj10GDzqjGgKjZr7NXRpyWI1fw40ARYQQZcT/8WkZa4Pv5sy0ios14Yt  
Seu6rppw5zVnPwp+0t0LZk74RbLm2HCr+3Jg64WuYjykvzx65d1KJOJ00LU7w6zp  
JvhsyjvhRhV/GMVBQJBzuRljhiM6sn8S0l+7wxBsTrJi7jtLXeJpJHQ+boHiaEjI  
JyW8sqaqK51WlwJIhy+1QoLXgwISUI6DS37iTuzFd7u57JeH6bwSvZlwV1eBXpt  
KufCwyA3zqsPb0rJ0fMo0XWBMX8B4gnIVoFQz076ghpgPoe/YcqqQCeVv/hPBLH9  
H3/OKfoVX0+UmvbwDH8Jf+y2NiaDaaotaZ4XehWqm1FDwZB8yWk6WrHud8ko5yHY  
kwqAGZxet2FfFGyaACKqTbiGA7WDSunzWnbMozvKE5T3VpZWM90Z2T2NVauZ5QAK  
9//07ghHjU0z90M61RLvBACZYqFezV+fkPPCjJWbikbsVE4TIF1fy9lhseLEntL1  
J1YfU1mcJ0xqeSsdavKTONjres3XisiY1Idhv3k8KmRCD1sVtL3Ax8TeIEy72A9h  
Lqpn4af/DbfdBc9cywXGJuDZSMtFdieIUUnfTBZ1jBqEggnz107EKhepe8B3JP1P  
AJgS7RDnZ8N9VHGpYgd/9PKAKxs0DPpfxBKFHF/mhI30Eueu9e/r1l9ib4c6NWfK  
2K21VQsNBpHTnRqWLMfYsQ6PBz971v61dmIZg/bSCDU7nx0K9xTSRVBwXJHmtQ9L  
73nv08e9Anr6i4PNMQew94p5xGgZ3gH4cxptisdUyEuaQFPjS+S90lJQxt7eGdee  
qIhw8XCmEVWwR15AN7KQXJF2vSAdJntxM4G4R+/sfnjSJStL4m4BgdQS/0D/4NeQ  
KYWA5kr1WzJstwUcICfgmFeA56KXy5H02cGjX9wkf7j1XNqRg+WuJGVzhGpbyT+X  
UX3MP9Y54XHxVI0fTzNwzAf4A3C+jIHhXXPbNL58PWnAzHKQz2U4NXXEWrh3ipgjI  
e93AMPS0KkX8BeYc76/zUwMvdrV78a19Fj0xtcJPm1V7DSL2CEs4smd02fjxZQPW  
Wh8fi8vaePx7UqXXGzWyzQZ+Hh2LYECRdC+RyCBHxAH1LxDC6quE9vYu/HfBhSiQ  
pRHc7Qd4wnScYIZpQDwXmzr16GUiwn/ZiL5DajGQ1SM365Nw5Lw01E+vpMFv8zAb  
xuh18dh7pE5uab6C0EICorple4db6ReUFpVIKXWAX0h4u3S60hpMxkw/KXwVSjjEV  
6IswiDPX2pF5SQDzXGLxyjYiyZcX+CnxQRH4Pt1JTiyj8W/qTVDbK4cFr0YT3gV9  
vjONZ1K6ba9L7gELx1YhfRa9GY0QyBRRi/uDwaXKaVqu2fGn3PdTn/ajQ5T70IYf  
Kg9Qz428N1H0jq+A/rWA+ENZ6jrXoqS2czpaa0inMaPTQjr8LrAp0meKmtEm1QgL  
YT9x7rGW9EKm1ztQYWwVyjEx3A382F+hJtPiQ80T0UpQLWRXvwKoXFRK1DdF8gn9  
z1NnGemSpM0bggksTSkwgMEji8ocOMYRj6I8LH+GJcn+oxr8gqtp6bU3sQ6amRDN  
p7ZSn8bkbjgjkM0UuMLGqVguikxS39Xu1f0glemiQ90IEwiHQLJBfJoePNZycNg  
hZIBMHwg6ykrZ1W3SErfj7rfqLzZRvzrBKjD1La7HQUZSJuuJWMtutbTXuV6QLYF  
dKGwerfXtGfW1qz1BW9TRabt4AyZDDLGiftCexF01CXcAXy7Tk9P11QosPtIZoc6  
0NoYv6oTDGrTxz4RJKPoQLNXKBWAK9YsHWDnr1LWznxfz+JjR0LGnhN5YzmB4Z1  
IxsUH80ETZjRzfVPHQBV+/jHkvXuXsXzJIQi7hFIkUtR94eonKCaSLyT9Twcftms  
FRwft/m5Zp/05VhK+X5cP6NaIFS+V3R0ZYAN9DCpDVPmcVL5fvQCJOW0IQU8Ey5n  
AL1kVIuCTGT4Ukay8jddMkmKvh10IafMmhkpyVeq/ttK7+ChBhpfWEyXrv6sFqDj  
p2Dvtr+AZzXIbP4RFhi9BJAytnVKE1WyLsbfl7c11jzxq0we2sJP/CTyks5k3e9w  
eq4mBwFhn+Yu/uQ1jNXu4r7904/E3FxFQFouqhTEDrWBE1XQhYEWPC4D8Cs10bb5  
zv/A9U13YIYDJf/dV8v9KP3ijrt8+01GnCWONHTP0rhFvv3BKmuOMEjDp0FnBHho  
nYECnfSTWh9Hqzo0Xk4tQyv0UrxYNFIVUzYZyBC8V3yVs4bufmifc7IJME/4kK4L  
jft00Ucb6NEDhZItimCrMPPqJL436ZC2LfYIH7z9E1r9qS18q1gQ2W07J3Ux9HSO  
6HAAo2PxjLSXiivQ+OCybk9KltXwPf5z4ci6+6iWvwHCY3D63Zf1yY8ABF36wDlo  
uuXcqs9vSWqQ+ICoVcreHSA+NbqokR/HjVGy30vg9x6JYfgJ87fEhWDtiYq+U4i  
GMppfeNohghA+kUbpT1joaoZyoM9LksTTVgmYvLvPGsTzJmVicLH9a8I76AhGkvp  
9BxNRnKIIOuqtdz+0+aNuVGNRfRwE2iikWqcghZk/DpaKhmm/pI0ZeJqnKWTJCG0  
tlC4BDECHN9PuJy20xlmJqRAM/2b+wIvyAeMH2aK9vPFC3T0dcSUNIBgU8FCCS0h  
+yW0/8EAMPHIrc5ixh3X1MwPeyoAvrPzyD/liK9tGZAXcZbWjFs0uIW1QvL7F6CD  
u2phhvDvH9vOAC+Qu4H11lWhZYcl2vK1H2avdwBUt0cvIJegQiRNoV/PhJjWaoD  
ELMDLa0XfWGR1uG5Lj1UJ4qkkYe6M1QoUg1G80djQG0VdhFRKbtfl0owbwA0pthC  
xDKgNhtVBjCHji4ER/ElyPvUWHK2XLmBGMX/Fqv+uAEK+0qTPZvFgtUfv5Xqqjbv  
xfjYq/fKyEncMPs0nYHifGwGeDSpqkR0n0K1xZGfQcTxeX4NzTbVITJdgk2zgA1y

GasqDt70Is5It7JEAvgZsb953UkLvz74uaXjKI2PibPYvJfLLvjAj/AQzQAv37y7  
q+B33tIEF45w8AfhncWl+5HuPe70mJ1nEAQ6pvID/OgFU+4AKTyE/zX035N0ghbs  
WCVDgQGYaNNkUe2X7cIGHuUNdc8RMLD3D8HSwFFDXUJw+sH5KuuirwAHy5nKy5pQ  
hcNjdt0t9e/U8Ml1pEuISzKBw0Xv6gLwFojtChGTARwNoju6/UmspkFzGsKDvkG  
c/3DvjfHJikRYtQg4AbyaMNX27+F5i/L8YGIHCEompCAvx5kPbJsPuzbqVox067  
jIdsgSAVqH8buvvcusvdf08ADRM7iJ/IE0MCyfb6zEU7pd3y+g5QDE3UvxNSK6byj  
+m1ccvpihhfi25eIymxMrWgrV8/1tpIuAi6C0Vyf047p5Y12fGwDv+Ur38nKsYzs  
hucJld5zV07sr2JHRXNg/Qky2Vtz3mdEOSij/x2PImEESIT78EQEDwEbrb+f/zQs  
gz3WhTybgu9Hx6ui4p6scevWXqfY4HlTOSkVDqa5VjmTFn8KSABtNgoqShHvgNdD  
xPAG4qJjmuYtxf1G1qiDRSK1x1Xczz+EjgtpfNzgmBofphTWQtMPkUtsPy/5ZCW7  
5IO5RICzbmfyPlijYXoaBGHYooznQ9fr9JBikDDcwVWz60toCa9iedf2CWGJt2s3  
Pgl01dfjBbFL80Ufaz7wa+sqjR2u9ofGcPJ0+pJGHdZ+1qsa094Rcb7woeMFgk46  
o57SK8SSDqNwED/fxMxSl/dMplp7z0h0WkcSQnPYWwl+Malom3f20XJyoyLi2W7F  
/PIw1YXFWlB5EKafFVAXpIfXy9q5Xzer/n8JJeXDHsDWECCgISAQbzzQvUb+vK3f  
JSeN9A0MTbIwsXC0qiWfgLlgt3myJb6l3+7tjKyHyWk/6uXPTdFEc3+nTptpZN/a  
ildTEQ5U/a/KTG3grVYff8UHAIVkswlTr6Tb4aHE4cDWcNQt6NS90JTHkzlkVBt  
pTB9/h5h0KEmu2r4MmMFN2W7I5cGE8bWRrWD/d0bo5TsZ46cu56vC9/oK01F/6SM  
2fnjMN0g8j2tT9C5Kxe/0Eq0A42iHxb6/ofrPFd9+Hj5LXTUgthT4JG2g29zecz  
yoPthHUQ4F76pw+hm2ch0JqwkINoPJmXaUIRLFG5b1wDHy7ueLc6FNvYnRdpmq2e  
gNXX3v47G06mfZ87bzBndKH0//toreTs39bAkZiJhBqiLGnidGHAKSVhJaetB+/  
8W9ntj/VwlwFQRDNBqXo/uIeUuNcRrnE6Riqe0BLEIJ4rD1Z/SRAAHWrWuHvRGh  
wb3yqcy+rgVSe72GY9aevVI8Jdyku308DAQEWQzpbTEmw1CsmPtLScVp7dguw3zS  
btN2WhMMgijPQFw0FiTogl0q8IeJI5ktfmqFod2reBvmz3a8/iayJqWn4dReO/8g  
rYRzWyay0hXJxAKd+7Irz7Xyo36Z3vnZk/eLFuMET8S8TWebP7oteXpH8erYNS9d  
6/ezrgzgYvj2LNApkYr6aL8gV+95DXFi96clx3fyA0gtlXN7usD1Ynu7DQa2iHsk  
tH5IuaZtkng9h05oRNZFI8l7BAJ+x0B/oEH8JI7lQxn0ZbKl53MnA7RSDt4zEmLt  
tsMrGc2gaQvLY+uFg6EWqStywefGY3+kQW4ZYTt4uttqGFz0sV2lXd84/V+Az8/w  
leGxvHouLP8EAF7HeRxH4KdX3hvmARN608+g2GkW8p5F5rF9n/dHJdkfUApu3dpT  
4E1SkTOHij8373rrGpigVeD55fIDH+Lr5TBsDEt1pM9Ah9owSSXhJf04zkQg43V/  
JSt73bIC1+IPlqS7zwzfFudbkUYxvE0YtsozoVvmcJUDQHmpEqkdCeg419R9w6gl  
bMKqC3gZWxbLiiRPSnrKh2iFmyIXU0r9KK5F97a6dt5Jzp+kSLG78br4iI4UH18t  
U3m/Mg78GSCxuhJoXYzYPDzWY4oui+R8dQUngQngW76Hd9WziHTbzV9ZEFycR0HU  
fb827Ptwlb+nG9mo7v3XkU1ysdFyhwsCWCijy7FyNgRHM6Y5mTT6c2rhuSEXegRb  
I2Rln78U5o10cBueGa+bmD5vEy41pZCnxiSkttAKoWdrQM97xQOH4qd2ihBbTnXB  
Pq03Hv9GwaixbsT+ZYFhZ22DUZ8iWB+i4Tnd1DN0IpDLk79lplqNktuemmAell9d  
Zh0484L7E1rqP0kfXde0fxiZONMeSYMmq0zAE5ivos8tKkpKfjL8ILWEwC/WDEBa  
rTtb/NXzjF70zRX7aNRgxA76nPF+C8KVz+Efa0u0i30bBUW0ZWLYJE/JMfWeGzrT  
zUcFk+CRiZ9+h1CpCkso3wBVAX9Y9tSKF8JgG1YqSZyNESV+QudgkCD3PhIle4IF  
SdUviM9z2wl+D/F9xqQP3D1fJaQr4XR9cFqltGdsZnTEhgEwr9Xp+klyODsAlld77  
xq7uyNCvshSPndoyX57nhXapGnzTaYsCSzz/UTuggGnw1QEqrqUqVn0oJCvjfXb  
RWMLT4r7ltsmZtq44s9+FUfIEANGjDu/Rgwp5n2AQjrcjTi0SC0slqE3cR0ccgAC  
/6R5/3RaMbqm0iQIv45BtEhvvE0FMX25ygZgdvun0a0vrdCnywcCCLq3zqsGVihd  
MXks2xBSVUx1UxD0H1805RU1x0B4EjzSDInzYN3x74vsfoqBLFibH3vw091H10lf  
EfrtenIchGwGD3ZEQ0YvJBs/ZkY858Z/4wYbXnmzm1mcINPNkLP0wndjidRSVkdAP  
TNnw1UNCshXbTz0Vy2BsIvj74MRDHNBxnRCb9nbD10jx2GKzs4/04mm1/eMKI+Z0  
mw+1LKEJAc4nJpSnkMD0DoKYQ0SHnTRXc7W3Nfqjr0fPCUgnedPGGwR6CN+meA9q  
LlzlqxYWrE9NbQbWRd5o0izyIjJhJObQ5WKpSf66hq3Lxr4pawtLzUxZTr0XbJvJ  
1tnNn6oBjMmj3a9rX+n1GCRtk0EUnlaGBC47okIh+AruBXExVfdWmV9yGIci/21P  
fMLCMoRdTB3aWRJxSJg3aMhsFIy2jNRJ5fYrDwSsZhV3xNAYbeNdEk2dNMZMNs+W  
K12u17F0yfBEtvDGRTJ0t0JC0/m1w1FUPrhKcA6CLjkHvGapkZND8n/tpGHYNfrg  
n4YAYydbax47+NH653jm0EYn1YDUNA+PvxwnR2A/p7XE7bPCFTc3s4vRwacbRyRz  
ziJ02vvIP6F1TXln0YrbcXnUXX2hrGwTEdGY0/bMo0KT4REGUSw8TLdrXQf6qAhy  
TJomC68oHHBVsr3Ipgg9pbRi/3cHE4SwEH7zMzchQS7N09p05vqtP+7zBTmNZSDc  
lEmY50mhJrM6PclpH+ScKX0bkQJwJ4t33xh+UFBYBhv8N2tMAHHq7oJM4oukp4H4  
h1cHUNnQv/SFxo+bz+RR7tFrCq0neal/5Grrg1mC+OUyxmuTVxhh54mx/6f3MYtU  
Md8wyAsvflxNHZnizWGxtSgUNj82DrQnOMhhaOCVjSY81INdlp+mN5ZZ2xCKX5C2  
RPF+bIZEJUAK/O/XXJe6G0yVnuOekLkCWDADYLeUMPV4taTqN4J69XNsDd0BynLM  
jM2/b/p/yzknGlmacVHY87NA09pmg/Ty1KWPtaB/csLI5xtBNv77bYHYq4ozQRi/

Qskaii16JZ7gt0VG7qs7VzqPZJZReh9m1f1S1UB3UWn1br0815VTkAmw/F+MytUt  
QKspwiExgHW5DM1XRv1+4fyVY8GRrRCQry2ihQJV0aEQdtskrDcdNb+KCSkht9rV  
Jm0Abnc/ZOCIawK36YL4x628BeoddaS6w9T7Q66ZKDoZ/YRTS4SKs3aupt2pu/5W  
wn0UBW16pK99082AxN2HWFQhniHyYKNCpt5VMY7JfAGu/YdnUkO6eqTuI59vC1Lb  
0BJBGMQtSoWENlmmI544j1HCiCOF0txgULZ5mYSwapmzG6wnyF1Ngo087ueExw1y  
rIDIG1W7BXseM/SjDD1bK+2k9s+1FY9YvqC7mzUnzHt4qu6rD/regruV1De2RP8h  
4WRBc0Yi2bCJYSS0UwGZF0t56d3gPQRJPdUCHE10vSbZm7AyLmoiuHD1xu1RT+6g  
Nyl18xddNdKm03FDi28imxp9oMX3f34kepNnuQtMhGHY3W1vYHtjVkfGzRyuJRip  
zyGqvNVwSHU4pxBevLyh4CdPbBi0QPiAAXQf/NKCWLPE8xzx5WZhwGxjTDu2YXoV  
nb51S8MENa0nA/nMUGLQ0HTB4sjORht2QqA7/1w8BMdzPas2tBzvR36ZeQyHnVTJ  
RUphKUM0ck6m2SQsMHR1PRaHkQNYPNk789GLahCHGJLJDeWqY0UKYIPCgRE4tsPN  
+LBx20BbBMVXK1U82z+CJuvbCbcl8miQNL55QSWXndGXDq9MrkAhYV7zvpnao4Ix  
mJZbHIYgSGuvkt/nLfUN11UESz1vEEeYafMGi6CaiHwKooSNcUpfI3Qu7bsoMtIz  
Nkt3Q0f41Bb+sHPkFKX1PD6g80BCHC/Dw/i7B1pWIG1Pjxj9jz99hErlyyGZmYR  
FutRwikGMIS65IGYVmhWqZFdwu/r5uWgy5vjN46j+q1GzW0d0qFtXd9Lz3w5zSSu  
hkCwP9N3HVhUSHG5fSXAiHnXNseqLNq/24aC//kPItBacYxbsiPqysvutlEtXkPa  
/RiERYKVMed1aC1law/5LEDORU0W0GusVyTe4sHrmWWRnnpsbG7HN6suToCACWDV  
EAtRmyR/ILJ9d7XhM6IuwHpi+6q8qLoH7yfEacBL2v+JgBOCHyVZ00DcEfRd3sWk  
oNVHwBntHtNnSrB224Ut30/rbhesnsxnDXQDEFuI097H4AVVSAG8vnqCLPtePffK  
LEIG7VD/PCsHxI3RE1eqJmeDAht5iATF7wpuyPikzZNXu0o5S57BF2dUn4Vei6QV  
84G6LYVvkqJnqEQybg1KAAXjAHXKe0gvUj0VZCCilz4tG7+mCm1Tw4glwtC30MdHx  
6tpXHBu73r0FR/QZ5MQGhYdUI3Zs4T0owcAlALM001Gv2d8Pi7i1bchT/o1ZuvE0  
AAVOv1GbWRBZBw9fJ5mffGujFEAy2uSjbxXG0z/t4/ktFazRqBFBLsxyq6TV9Jyu  
16990eIcJ6j/PpJ95HY41P6imW4daxFw497yTR9N5cxEQ/hCyJkRxiki7vkT1QgV  
cfwOyoKXsRH7uYwnPL6k19hPPA08gl3PGAELKMod8Pe6cYDrndpY6ZG4hnHQcdIT  
PLd158T/J/cv8j8k1cbfbNDdCu7eSxZmb2j01LH5aWcMxBkgLC10+KfiAAYTyS0d  
T3Z0AyhGebeAstrS0QpWR1/DgZruF5ENSfQjPzbbVUjYfKp/bG/6yqwXyqNiTd5g  
X031o5YzNVMkBpmPMNakuAc94dnKB5UcFDGDtJLk5wHFfreeHtdCjMb9Qc2qTvws  
ERpuANtbJW6Wyc1DrGi+wNMsPBvUtpwqoJ+bFqMq3e0ALLmdk11lgd6pqbiWRMRD  
MJMhTDxCo6r+jeWcsjsIhkbi8oic74I32od1+8TC00u/aDWoRCK914T5V2hgcCB  
Bj9f6rPheJn1LRxWC9/DHU4f9uointgzEyu76T+xF88zDoCFcnJyWk1wtIGb2FfW  
pJ/3khXjctDuIqFHsJDj8RS2mf/8w8s0A/gAdt19xE0dwutIc01BQJ3hgFw3zGWN  
aoqwmjXVLMzZo81D9+dcRnrhX7Kgk9cAyp46FUxhq/xi5D151iIq2X6PpGua6R2J  
K5zv1z82/J0c70Wmk87sMpFIRrAKsuMzGd+h+3gg4xNUE/P7ilrynro04f7rWOLE  
r7ZjVjJCz+N9xfDVb/ZkU0AXwIjwVI2cBY1S0WNoI4aGkTTynKTdF7omWn9zdx1b  
k1osiQdFJ0UrqoYamJhZKkbIjk91MNw66fYfwdPvChMwXDJFiBlNtk2o8fqJs0/k  
Qj4mYICTIXtGufRsvp/Gf8ZJyeFJyC0HNpNpFbj+ztreTcAvfLh9gZ7LafqhuIe  
YmRIQ119L/0kyKFPlKyNhn0VoTIBJo/NdYwVYqo9mg104AhzJF2vRV/oUJrjqWQ  
e/pdAXouAiG0MfksQIG0dQHLgzkpDdiw10z44NVf/rw18/JZJ8GObIY80GBYP9Q  
XVASNw6CwNovavj8F1+7IimDFEakAcr42Cx57c9AN0aoSH9mfQ6g/y8HwKNY7Cw6  
JzPjP1Ks1JKQKxWEEelxwDXDefclv/NbQkYXRF3BxKz8AWBqaV3MZQjgMcDbAe14  
cBgbk6/3sSwLH7Vo1rj5aZ/15jXLvAZ1ZBHVq9Tm03kxobDi04B4wvaHd0M0SKuL  
XjBS1jsCcs0+pPnays3abFkH0XPaCPDQPJ35UoN3YGdYn2cRJBP0hoaNKYnFToYi  
EoYR1A70xiY+D44GjVpgMNCpql92P4g0eajaeBtxN4wzWY37a8+WRU++VOUVxtOb  
ferZYYN1kT3FEH2iXQDNSne6lmxRV6RodUMn7AtJSk01yTj9zDMb2nC/G8PUWd2K  
Bf6HxY5ZFu8zS4gU9I4/ZUPr6q0cX0vcgdf6eUeTYRczTHiAqY3z4FPtZFBhwSS  
88FdYi5S8YaujR0/tsdWNU/ml7YFzDnbSa+1PuzKNy6kUcbXAY3IaTtY95Ht1Ig0  
nAV//oxfDBGx0UutPCVNJiRCRZkY3w6sk0cLR2BYU2MPC7BnpQcSyqFk6a0+Ft72  
cI4jjWHXjUsxb31IjLC+AUjyTj0qT+BVkHI+0wxc9/gVReQQ362c0CPDu6NScAji  
+q66sHQ13aZL+5q3PCGxHwhwR0JeWDqmHkyUNEFcPNGsCrS/ocbawlmjIsym4+nV  
khWAuy4kkd0KAhPlUQX1VUp4QdXnYh231R/1NPexrsYP7DjCqC00/122h4pPv3fW  
wa6hyIjVZuF3BsqRENsUIEygj9iLG3FmuJYJCGrs38FL1pEDjGbiyB3JDvOZPgq0  
YIOKvD3KGQCz/bBehGG3IwTbZDUGmqtKA0eieWzYC57Jd7tHXttm5PMz64ziSaTW  
oc1h10rm0QsWZLPfFlre5fm6XX3rBPX08PB95Bp0/H0DFqTK9uAF1eD6nYAHWLQS  
XjRDBK2Qnz++Mco908nQt5HHXNArgXM0v8qlbiNPs/00vvp0va/91wmLZaMMdtwe  
fJfSvoXUZW35PW6ubFf0EEAh1gQtm5v1lZCcUqitYYvNsBLBEyBDTY4igoKb/m0B  
5zx1ebR5n56wEN1ealdDjGtB1ear1LRHZ6W0QdgQDP0pd+ILzSmALq5epYWjogkx

```
UYKYCyx6a5bvjcD1H5i09iK2IW4247sY2h0kRg1lKLZq
-----END CERTIFICATE-----
```

## Acknowledgments

Much of the structure and text of this document is based on [RFC8410] and [RFC9881]. The remainder comes from [RFC9814]. Thanks to the authors of those documents, and the ones they based their work on, for making our work easier. "Copying always makes things easier and less error prone" [RFC8411]. Thanks to Sean Turner for helpful text and to Markku-Juhani O. Saarinen for side-channel clarifications.

## Authors' Addresses

### Kaveh Bashiri

BSI

Email: [kaveh.bashiri.ietf@gmail.com](mailto:kaveh.bashiri.ietf@gmail.com)

### Scott Fluhrer

Cisco Systems

Email: [sfluhrer@cisco.com](mailto:sfluhrer@cisco.com)

### Stefan-Lukas Gazdag

genua GmbH

Email: [ietf@gazdag.de](mailto:ietf@gazdag.de)

### Daniel Van Geest

CryptoNext Security

Email: [daniel.vangeest@cryptonext-security.com](mailto:daniel.vangeest@cryptonext-security.com)

### Stavros Kousidis

BSI

Email: [kousidis.ietf@gmail.com](mailto:kousidis.ietf@gmail.com)