

NETCONF Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 6, 2018

K. Watsen
Juniper Networks
June 4, 2018

Common YANG Data Types for Cryptography
draft-ietf-netconf-crypto-types-00

Abstract

This document defines YANG identities and typedefs useful for cryptographic applications.

Editorial Note (To be removed by RFC Editor)

This draft contains many placeholder values that need to be replaced with finalized values at the time of publication. This note summarizes all of the substitutions that are needed. No other RFC Editor instructions are specified elsewhere in this document.

Artwork in this document contains shorthand references to drafts in progress. Please apply the following replacements:

- o "XXXX" --> the assigned RFC value for this draft

Artwork in this document contains placeholder values for the date of publication of this draft. Please apply the following replacement:

- o "2018-06-04" --> the publication date of this draft

The following Appendix section is to be removed prior to publication:

- o Appendix A. Change Log

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 6, 2018.

Copyright Notice

Copyright (c) 2018 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. YANG Module	3
3. Security Considerations	11
4. IANA Considerations	11
4.1. The IETF XML Registry	11
4.2. The YANG Module Names Registry	12
5. References	12
5.1. Normative References	12
5.2. Informative References	13
Appendix A. Change Log	14
A.1. I-D to 00	14
Acknowledgements	14
Author's Address	14

1. Introduction

This document defines a YANG 1.1 [RFC7950] module specifying identities and typedefs useful for cryptography.

As the YANG module only defines identities and typedefs, this draft does not present a YANG tree diagram [RFC8340] or any examples illustrating usage of the module.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP

14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. YANG Module

This module has normative references for [RFC4253], [RFC5280], [RFC5480], [RFC5652], and [ITU.X690.2015] and has informational references to [RFC6234] and [RFC8017]

```
<CODE BEGINS> file "ietf-crypto-types@2018-06-04.yang"
module ietf-crypto-types {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-crypto-types";
  prefix "ct";

  organization
    "IETF NETCONF (Network Configuration) Working Group";

  contact
    "WG Web: <http://datatracker.ietf.org/wg/netconf/>
    WG List: <mailto:netconf@ietf.org>

    Author: Kent Watsen
            <mailto:kwatsen@juniper.net>";

  description
    "This module defines common YANG types for cryptographic
    applications.

    Copyright (c) 2018 IETF Trust and the persons identified
    as authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with
    or without modification, is permitted pursuant to, and
    subject to the license terms contained in, the Simplified
    BSD License set forth in Section 4.c of the IETF Trust's
    Legal Provisions Relating to IETF Documents
    (http://trustee.ietf.org/license-info).
```

```
}

/*****/
/* Identities for Hashing Algorithms */
/*****/

identity hash-algorithm {
  description
    "A base identity for hash algorithm verification.";
}

identity sha-256 {
  base "hash-algorithm";
  description "The SHA-256 algorithm.";
  reference "RFC 6234: US Secure Hash Algorithms.";
}

/*****/
/* Identities for Key Algorithms */
/*****/

identity key-algorithm {
  description
    "Base identity from which all key-algorithms are derived.";
}

identity rsa1024 {
  base key-algorithm;
  description
    "The RSA algorithm using a 1024-bit key.";
  reference
    "RFC 8017:
    PKCS #1: RSA Cryptography Specifications Version 2.2.";
}

identity rsa2048 {
  base key-algorithm;
  description
    "The RSA algorithm using a 2048-bit key.";
  reference
    "RFC 8017:
    PKCS #1: RSA Cryptography Specifications Version 2.2.";
}

identity rsa3072 {
  base key-algorithm;
  description
```

```
    "The RSA algorithm using a 3072-bit key.";
  reference
    "RFC 8017:
      PKCS #1: RSA Cryptography Specifications Version 2.2.";
}

identity rsa4096 {
  base key-algorithm;
  description
    "The RSA algorithm using a 4096-bit key.";
  reference
    "RFC 8017:
      PKCS #1: RSA Cryptography Specifications Version 2.2.";
}

identity rsa7680 {
  base key-algorithm;
  description
    "The RSA algorithm using a 7680-bit key.";
  reference
    "RFC 8017:
      PKCS #1: RSA Cryptography Specifications Version 2.2.";
}

identity rsal5360 {
  base key-algorithm;
  description
    "The RSA algorithm using a 15360-bit key.";
  reference
    "RFC 8017:
      PKCS #1: RSA Cryptography Specifications Version 2.2.";
}

identity secp192r1 {
  base key-algorithm;
  description
    "The secp192r1 algorithm.";
  reference
    "RFC 5480: Elliptic Curve Cryptography Subject Public
      Key Information.";
}

identity secp256r1 {
  base key-algorithm;
  description
    "The secp256r1 algorithm.";
  reference
    "RFC 5480: Elliptic Curve Cryptography Subject Public
```

```
        Key Information.";
    }

    identity secp384r1 {
        base key-algorithm;
        description
            "The secp384r1 algorithm.";
        reference
            "RFC 5480: Elliptic Curve Cryptography Subject Public
            Key Information.";
    }

    identity secp521r1 {
        base key-algorithm;
        description
            "The secp521r1 algorithm.";
        reference
            "RFC 5480: Elliptic Curve Cryptography Subject Public
            Key Information.";
    }

    /*
     * Typedefs for identityrefs to above base identities
     */
    typedef hash-algorithm-ref {
        type identityref {
            base "hash-algorithm";
        }
        description
            "This typedef enables importing modules to easily define an
            identityref to the 'hash-algorithm' base identity.";
    }

    typedef key-algorithm-ref {
        type identityref {
            base "key-algorithm";
        }
        description
            "This typedef enables importing modules to easily define an
            identityref to the 'key-algorithm' base identity.";
    }

    /*
     * Typedefs for ASN.1 structures from RFC 5280
     */
```

```
typedef x509 {
  type binary;
  description
    "A Certificate structure, as specified in RFC 5280,
    encoded using ASN.1 distinguished encoding rules (DER),
    as specified in ITU-T X.690.";
  reference
    "RFC 5280:
    Internet X.509 Public Key Infrastructure Certificate
    and Certificate Revocation List (CRL) Profile
    ITU-T X.690:
    Information technology - ASN.1 encoding rules:
    Specification of Basic Encoding Rules (BER),
    Canonical Encoding Rules (CER) and Distinguished
    Encoding Rules (DER).";
}
```

```
typedef crl {
  type binary;
  description
    "A CertificateList structure, as specified in RFC 5280,
    encoded using ASN.1 distinguished encoding rules (DER),
    as specified in ITU-T X.690.";
  reference
    "RFC 5280:
    Internet X.509 Public Key Infrastructure Certificate
    and Certificate Revocation List (CRL) Profile
    ITU-T X.690:
    Information technology - ASN.1 encoding rules:
    Specification of Basic Encoding Rules (BER),
    Canonical Encoding Rules (CER) and Distinguished
    Encoding Rules (DER).";
}
```

```
/*
*****
/*   Typedefs for ASN.1 structures from 5652   */
*****
/*
```

```
typedef cms {
  type binary;
  description
    "A ContentInfo structure, as specified in RFC 5652,
    encoded using ASN.1 distinguished encoding rules (DER),
    as specified in ITU-T X.690.";
  reference
    "RFC 5652:
    Cryptographic Message Syntax (CMS)
    ITU-T X.690:
```

```
        Information technology - ASN.1 encoding rules:
        Specification of Basic Encoding Rules (BER),
        Canonical Encoding Rules (CER) and Distinguished
        Encoding Rules (DER).";
    }

typedef data-content-cms {
    type cms;
    description
        "A CMS structure whose top-most content type MUST be the
        data content type, as described by Section 4 in RFC 5652.";
    reference
        "RFC 5652: Cryptographic Message Syntax (CMS)";
}

typedef signed-data-cms {
    type cms;
    description
        "A CMS structure whose top-most content type MUST be the
        signed-data content type, as described by Section 5 in
        RFC 5652.";
    reference
        "RFC 5652: Cryptographic Message Syntax (CMS)";
}

typedef enveloped-data-cms {
    type cms;
    description
        "A CMS structure whose top-most content type MUST be the
        enveloped-data content type, as described by Section 6
        in RFC 5652.";
    reference
        "RFC 5652: Cryptographic Message Syntax (CMS)";
}

typedef digested-data-cms {
    type cms;
    description
        "A CMS structure whose top-most content type MUST be the
        digested-data content type, as described by Section 7
        in RFC 5652.";
    reference
        "RFC 5652: Cryptographic Message Syntax (CMS)";
}

typedef encrypted-data-cms {
    type cms;
    description
```

```
        "A CMS structure whose top-most content type MUST be the
        encrypted-data content type, as described by Section 8
        in RFC 5652.";
    reference
        "RFC 5652: Cryptographic Message Syntax (CMS)";
}

typedef authenticated-data-cms {
    type cms;
    description
        "A CMS structure whose top-most content type MUST be the
        authenticated-data content type, as described by Section 9
        in RFC 5652.";
    reference
        "RFC 5652: Cryptographic Message Syntax (CMS)";
}

/*****
/*   Typedefs for structures related to RFC 4253   */
*****/

typedef ssh-host-key {
    type binary;
    description
        "The binary public key data for this SSH key, as
        specified by RFC 4253, Section 6.6, i.e.:"

        string    certificate or public key format
                identifier
        byte[n]   key/certificate data.";
    reference
        "RFC 4253: The Secure Shell (SSH) Transport Layer
        Protocol";
}

/*****
/*   Typedefs for ASN.1 structures related to RFC 5280   */
*****/

typedef trust-anchor-cert-x509 {
    type x509;
    description
        "A Certificate structure that MUST encode a self-signed
        root certificate.";
}

typedef end-entity-cert-x509 {
    type x509;
```

```
description
  "A Certificate structure that MUST encode a certificate
  that is neither self-signed nor having Basic constraint
  CA true.";
}

/*****
/*  Typedefs for ASN.1 structures related to RFC 5652  */
*****/

typedef trust-anchor-cert-cms {
  type signed-data-cms;
  description
    "A CMS SignedData structure that MUST contain the chain of
    X.509 certificates needed to authenticate the certificate
    presented by a client or end-entity.

    The CMS MUST contain only a single chain of certificates.
    The client or end-entity certificate MUST only authenticate
    to last intermediate CA certificate listed in the chain.

    In all cases, the chain MUST include a self-signed root
    certificate. In the case where the root certificate is
    itself the issuer of the client or end-entity certificate,
    only one certificate is present.

    This CMS structure MAY (as applicable where this type is
    used) also contain suitably fresh (as defined by local
    policy) revocation objects with which the device can
    verify the revocation status of the certificates.

    This CMS encodes the degenerate form of the SignedData
    structure that is commonly used to disseminate X.509
    certificates and revocation objects (RFC 5280).";
  reference
    "RFC 5280:
    Internet X.509 Public Key Infrastructure Certificate
    and Certificate Revocation List (CRL) Profile.";
}

typedef end-entity-cert-cms {
  type signed-data-cms;
  description
    "A CMS SignedData structure that MUST contain the end
    entity certificate itself, and MAY contain any number
    of intermediate certificates leading up to a trust
    anchor certificate. The trust anchor certificate
    MAY be included as well.
```

The CMS MUST contain a single end entity certificate.
The CMS MUST NOT contain any spurious certificates.

This CMS structure MAY (as applicable where this type is used) also contain suitably fresh (as defined by local policy) revocation objects with which the device can verify the revocation status of the certificates.

```
This CMS encodes the degenerate form of the SignedData
structure that is commonly used to disseminate X.509
certificates and revocation objects (RFC 5280).";
reference
  "RFC 5280:
   Internet X.509 Public Key Infrastructure Certificate
   and Certificate Revocation List (CRL) Profile.";
}
}
<CODE ENDS>
```

3. Security Considerations

In order to use YANG identities for algorithm identifiers, only the most commonly used RSA key lengths are supported for the RSA algorithm. Additional key lengths can be defined in another module or added into a future version of this document.

This document limits the number of elliptical curves supported. This was done to match industry trends and IETF best practice (e.g., matching work being done in TLS 1.3). If additional algorithms are needed, they can be defined by another module or added into a future version of this document.

The YANG module defined in this document specifies only typedefs and identities, and hence there are no YANG-specific security considerations that need to be addressed.

4. IANA Considerations

4.1. The IETF XML Registry

This document registers one URI in the "ns" subregistry of the IETF XML Registry [RFC3688]. Following the format in [RFC3688], the following registration is requested:

```
URI: urn:ietf:params:xml:ns:yang:ietf-crypto-types
Registrant Contact: The NETCONF WG of the IETF.
XML: N/A, the requested URI is an XML namespace.
```

4.2. The YANG Module Names Registry

This document registers one YANG module in the YANG Module Names registry [RFC6020]. Following the format in [RFC6020], the the following registration is requested:

```
name:          ietf-crypto-types
namespace:     urn:ietf:params:xml:ns:yang:ietf-crypto-types
prefix:        ct
reference:     RFC XXXX
```

5. References

5.1. Normative References

- [ITU.X690.2015] International Telecommunication Union, "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, August 2015, <<https://www.itu.int/rec/T-REC-X.690/>>.
- [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>>.
- [RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <<https://www.rfc-editor.org/info/rfc4253>>.
- [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>>.
- [RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, DOI 10.17487/RFC5480, March 2009, <<https://www.rfc-editor.org/info/rfc5480>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.

- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [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>>.

5.2. Informative References

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <<https://www.rfc-editor.org/info/rfc6234>>.
- [RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <<https://www.rfc-editor.org/info/rfc8017>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.

Appendix A. Change Log

A.1. I-D to 00

- o Removed groupings and notifications.
- o Added typedefs for identityrefs.
- o Added typedefs for other RFC 5280 structures.
- o Added typedefs for other RFC 5652 structures.
- o Added convenience typedefs for RFC 4253, RFC 5280, and RFC 5652.

Acknowledgements

The authors would like to thank for following for lively discussions on list and in the halls (ordered by last name): Martin Bjorklund, Balazs Kovacs, Eric Voit, and Liang Xia.

Author's Address

Kent Watsen
Juniper Networks

EMail: kwatsen@juniper.net