Workgroup: NETCONF Working Group
Internet-Draft:
draft-ietf-netconf-tls-client-server-28
Published: 24 May 2022
Intended Status: Standards Track
Expires: 25 November 2022
Authors: K. Watsen
Watsen Networks

YANG Groupings for TLS Clients and TLS Servers

Abstract

This document defines three YANG 1.1 modules: the first defines features and groupings common to both TLS clients and TLS servers, the second defines a grouping for a generic TLS client, and the third defines a grouping for a generic TLS server.

Editorial Note (To be removed by RFC Editor)

This draft contains 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:

* AAAA --> the assigned RFC value for draft-ietf-netconf-crypto-types
* BBBB --> the assigned RFC value for draft-ietf-netconf-trust-anchors
* CCCC --> the assigned RFC value for draft-ietf-netconf-keystore
* DDDD --> the assigned RFC value for draft-ietf-netconf-tcp-client-server
* FFFF --> 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:

* 2022-05-24 --> the publication date of this draft

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

* Appendix B. Change Log

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines three YANG 1.1 [RFC7950] modules: the first defines features and groupings common to both TLS clients and TLS servers, the second defines a grouping for a generic TLS client, and the third defines a grouping for a generic TLS server.

Any version of TLS may be configured. TLS 1.0 [RFC2246] and TLS 1.1 [RFC4346] are historic and hence the YANG "feature" statements enabling them are marked "status obsolete". TLS 1.2 [RFC5246] is obsoleted by TLS 1.3 [RFC8446] but still in common use, and hence its "feature" statement is marked "status deprecated". All the feature statements for 1.0, 1.1, and 1.3 have "description" statements stating that it is NOT RECOMMENDED to enable obsolete protocol versions.

It is intended that the YANG groupings will be used by applications needing to configure TLS client and server protocol stacks. For instance, these groupings are used to help define the data model for HTTPS [RFC2818] and NETCONF over TLS [RFC7589] based clients and servers in [I-D.ietf-netconf-http-client-server] and [I-D.ietf-netconf-netconf-client-server] respectively.

The client and server YANG modules in this document each define one grouping, which is focused on just TLS-specific configuration, and specifically avoids any transport-level configuration, such as what ports to listen-on or connect-to. This affords applications the opportunity to define their own strategy for how the underlying TCP connection is established. For instance, applications supporting NETCONF Call Home [RFC8071] could use the "tls-server-grouping" grouping for the TLS parts it provides, while adding data nodes for the TCP-level call-home configuration.

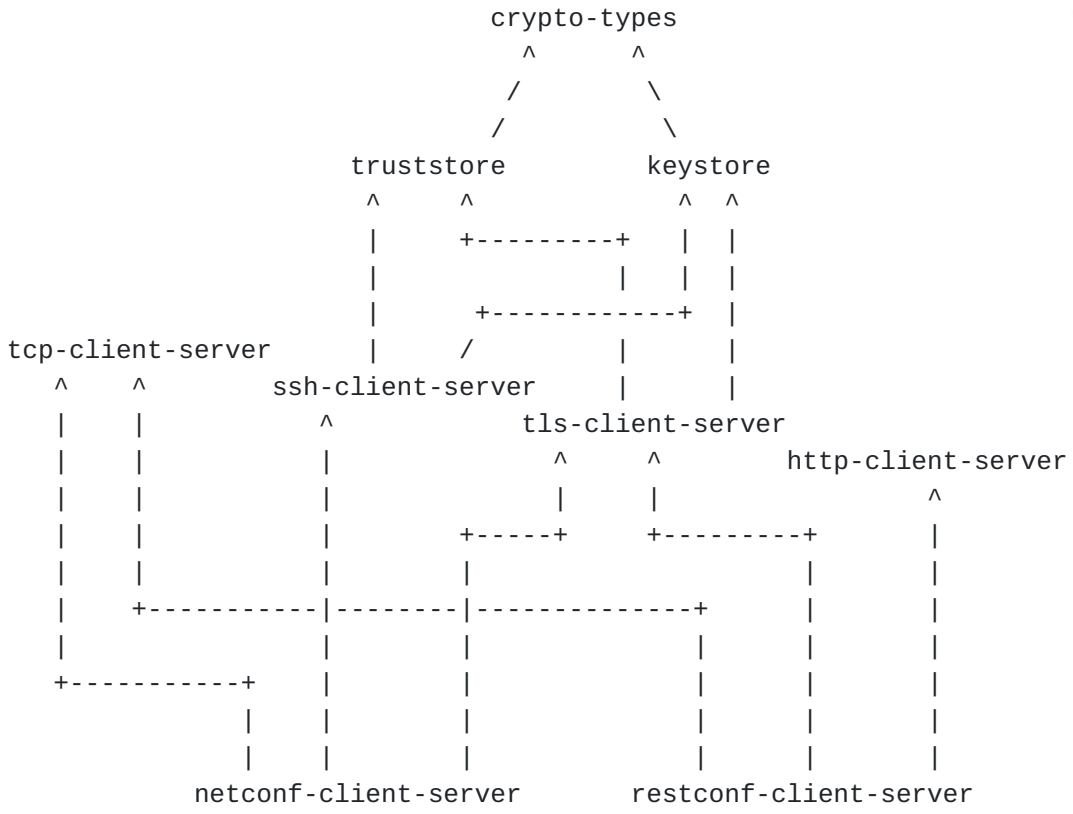
1.1. Relation to other RFCs

This document presents one or more YANG modules [RFC7950] that are part of a collection of RFCs that work together to, ultimately, enable the configuration of the clients and servers of both the NETCONF [RFC6241] and RESTCONF [RFC8040] protocols.

The modules have been defined in a modular fashion to enable their use by other efforts, some of which are known to be in progress at the time of this writing, with many more expected to be defined in time.

The normative dependency relationship between the various RFCs in the collection is presented in the below diagram. The labels in the

diagram represent the primary purpose provided by each RFC. Hyperlinks to each RFC are provided below the diagram.



Label in Diagram	Originating RFC
crypto-types	[I-D.ietf-netconf-crypto-types]
truststore	[I-D.ietf-netconf-trust-anchors]
keystore	[I-D.ietf-netconf-keystore]
tcp-client-server	[I-D.ietf-netconf-tcp-client-server]
ssh-client-server	[I-D.ietf-netconf-ssh-client-server]
tls-client-server	[I-D.ietf-netconf-tls-client-server]
http-client-server	[I-D.ietf-netconf-http-client-server]
netconf-client-server	[I-D.ietf-netconf-netconf-client-server]
restconf-client-server	[I-D.ietf-netconf-restconf-client-server]

Table 1: Label to RFC Mapping

1.2. Specification Language

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.

1.3. Adherence to the NMDA

This document is compliant with the Network Management Datastore Architecture (NMDA) [RFC8342]. For instance, as described in [I-D.ietf-netconf-trust-anchors] and [I-D.ietf-netconf-keystore], trust anchors and keys installed during manufacturing are expected to appear in <operational>.

1.4. Conventions

Various examples used in this document use a placeholder value for binary data that has been base64 encoded (e.g., "BASE64VALUE="). This placeholder value is used as real base64 encoded structures are often many lines long and hence distracting to the example being presented.

2. The "ietf-tls-common" Module

The TLS common model presented in this section contains features and groupings common to both TLS clients and TLS servers. The "hello-params-grouping" grouping can be used to configure the list of TLS algorithms permitted by the TLS client or TLS server. The lists of algorithms are ordered such that, if multiple algorithms are permitted by the client, the algorithm that appears first in its list that is also permitted by the server is used for the TLS transport layer connection. The ability to restrict the algorithms allowed is provided in this grouping for TLS clients and TLS servers that are capable of doing so and may serve to make TLS clients and TLS servers compliant with local security policies. This model supports both TLS 1.2 [RFC5246] and TLS 1.3 [RFC8446].

Thus, in order to support both TLS1.2 and TLS1.3, the cipher-suites part of the "hello-params-grouping" grouping should include three parameters for configuring its permitted TLS algorithms, which are: TLS Cipher Suites, TLS SignatureScheme, TLS Supported Groups. Note that TLS1.2 only uses TLS Cipher Suites.

2.1. Data Model Overview

This section provides an overview of the "ietf-tls-common" module in terms of its features, identities, and groupings.

2.1.1. Features

The following diagram lists all the "feature" statements defined in the "ietf-tls-common" module:

```
Features:
  +-- tls10
  +-- tls11
  +-- tls12
  +-- tls13
  +-- hello-params
  +-- public-key-generation
```

The diagram above uses syntax that is similar to but not defined in [RFC8340].

2.1.2. Identities

The following diagram illustrates the relationship amongst the "identity" statements defined in the "ietf-tls-common" module:

```
Identities:
  +-- tls-version-base
     +-- tls10
     +-- tls11
     +-- tls12
     +-- tls13
```

The diagram above uses syntax that is similar to but not defined in [RFC8340].

Comments:

- The diagram shows that there are two base identities.
- One base identity is used to specific TLS versions, while the other is used to specify cipher-suites.
- These base identities are "abstract", in the object oriented programming sense, in that they only define a "class" of things, rather than a specific thing.

2.1.3. Groupings

The "ietf-tls-common" module defines the following "grouping" statement:

- hello-params-grouping

This grouping is presented in the following subsection.

2.1.3.1. The "hello-params-grouping" Grouping

The following tree diagram [RFC8340] illustrates the "hello-params-grouping" grouping:

```
  grouping hello-params-grouping:
    +-- tls-versions
    |  +-- tls-version*   identityref
    +-- cipher-suites
       +-- cipher-suite*   identityref
```

Comments:

*This grouping is used by both the "tls-client-grouping" and the "tls-server-grouping" groupings defined in Section 3.1.2.1 and Section 4.1.2.1, respectively.

*This grouping enables client and server configurations to specify the TLS versions and cipher suites that are to be used when establishing TLS sessions.

*The "cipher-suites" list is "ordered-by user".

2.1.4. Protocol-accessible Nodes

The following tree diagram [RFC8340] lists all the protocol-accessible nodes defined in the "ietf-tls-common" module, without expanding the "grouping" statements:

```
module: ietf-tls-common

  rpcs:
    +---x generate-public-key {public-key-generation}?
       +---w input
       |  +---w algorithm
       |  |       tlscsa:cipher-suite-algorithm-ref
       |  +---w bits?                uint16
       |  +---w (private-key-encoding)?
       |     +--:(cleartext)
       |     |  +---w cleartext?      empty
       |     +--:(encrypt) {ct:private-key-encryption}?
       |     |  +---w encrypt-with
       |     |     +---w ks:encrypted-by-choice-grouping
       |     +--:(hide) {ct:hidden-keys}?
       |        +---w hide?           empty
       +--ro output
          +---u ct:asymmetric-key-pair-grouping
```

The following tree diagram [RFC8340] lists all the protocol-accessible nodes defined in the "ietf-tls-common" module, with all "grouping" statements expanded, enabling the module's full structure to be seen:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

module: ietf-tls-common

  rpcs:
    +---x generate-public-key {public-key-generation}?
       +---w input
       |  +---w algorithm
       |  |       tlscsa:cipher-suite-algorithm-ref
       |  +---w bits?                uint16
       |  +---w (private-key-encoding)?
       |     +--:(cleartext)
       |     |  +---w cleartext?      empty
       |     +--:(encrypt) {ct:private-key-encryption}?
       |     |  +---w encrypt-with
       |     |     +---w (encrypted-by-choice)
       |     |        +--:(symmetric-key-ref)
       |     |        |        {central-keystore-supported,symmetric\
-keys}?
       |     |        |  +---w symmetric-key-ref?
       |     |        |          ks:symmetric-key-ref
       |     |        +--:(asymmetric-key-ref)
       |     |                 {central-keystore-supported,asymmetri\
c-keys}?
       |     |           +---w asymmetric-key-ref?
       |     |                   ks:asymmetric-key-ref
       |     +--:(hide) {ct:hidden-keys}?
       |        +---w hide?           empty
       +--ro output
          +--ro public-key-format              identityref
          +--ro public-key                     binary
          +--ro private-key-format?            identityref
          +--ro (private-key-type)
             +--:(cleartext-private-key)
             |  +--ro cleartext-private-key?   binary
             +--:(hidden-private-key) {hidden-keys}?
             |  +--ro hidden-private-key?      empty
             +--:(encrypted-private-key) {private-key-encryption}?
                +--ro encrypted-private-key
                   +--ro encrypted-by
                   +--ro encrypted-value-format    identityref
                   +--ro encrypted-value           binary
```

Comments:

*Protocol-accessible nodes are those nodes that are accessible when the module is "implemented", as described in Section 5.6.5 of [RFC7950].

*The protocol-accessible nodes for the "ietf-tls-common" module are limited to the RPC "generate-public-key", which is additionally constrained by the feature "public-key-generation".

*The "encrypted-by-choice-grouping" grouping is discussed in Section 2.1.3.1 of [I-D.ietf-netconf-keystore].

*The "asymmetric-key-pair-grouping" grouping is discussed in Section 2.1.4.5 of [I-D.ietf-netconf-crypto-types].

2.2. Example Usage

The following example illustrates the "hello-params-grouping' grouping when populated with some data.

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

<!-- The outermost element below doesn't exist in the data model. -->
<!--  It simulates if the "grouping" were a "container" instead.  -->

<hello-params
   xmlns="urn:ietf:params:xml:ns:yang:ietf-tls-common"
   xmlns:tlscmn="urn:ietf:params:xml:ns:yang:ietf-tls-common"
   xmlns:tlscsa="urn:ietf:params:xml:ns:yang:iana-tls-cipher-suite-a\
lgs">
  <tls-versions>
    <tls-version>tlscmn:tls11</tls-version>
    <tls-version>tlscmn:tls12</tls-version>
  </tls-versions>
  <cipher-suites>
    <cipher-suite>tlscsa:tls-ecdhe-ecdsa-with-aes-256-cbc-sha</ciphe\
r-suite>
    <cipher-suite>tlscsa:tls-dhe-rsa-with-aes-128-cbc-sha256</cipher\
-suite>
    <cipher-suite>tlscsa:tls-rsa-with-3des-ede-cbc-sha</cipher-suite>
  </cipher-suites>
</hello-params>
```

The following example illustrates the "generate-public-key" RPC.

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <generate-public-key
    xmlns="urn:ietf:params:xml:ns:yang:ietf-tls-common"
    xmlns:tlscsa="urn:ietf:params:xml:ns:yang:iana-tls-cipher-suite-\
algs">
    <algorithm>tlscsa:tls-ecdhe-psk-with-aes-128-gcm-sha256</algorit\
hm>
    <bits>521</bits>
    <encrypt-with>
      <asymmetric-key-ref>hidden-asymmetric-key</asymmetric-key-ref>
    </encrypt-with>
  </generate-public-key>
</rpc>
```

2.3. YANG Module

This YANG module has a normative references to [RFC4346], [RFC5288], [RFC5289], [RFC8422], and FIPS PUB 180-4.

This YANG module has a informative references to [RFC2246], [RFC4346], [RFC5246], and [RFC8446].

<CODE BEGINS> file "ietf-tls-common@2022-05-24.yang"

```
module ietf-tls-common {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tls-common";
  prefix tlscmn;

  import iana-tls-cipher-suite-algs {
    prefix tlscsa;
    reference
      "RFC FFFF: YANG Groupings for TLS Clients and SSH Servers";
  }

  import ietf-crypto-types {
    prefix ct;
    reference
      "RFC AAAA: YANG Data Types and Groupings for Cryptography";
  }

  import ietf-keystore {
    prefix ks;
    reference
      "RFC CCCC: A YANG Data Model for a Keystore";
  }

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

  contact
    "WG List:  NETCONF WG list <mailto:netconf@ietf.org>
     WG Web:   https://datatracker.ietf.org/wg/netconf
     Author:   Kent Watsen <mailto:kent+ietf@watsen.net>
     Author:   Jeff Hartley <mailto:jeff.hartley@commscope.com>
     Author:   Gary Wu <mailto:garywu@cisco.com>";

   description
    "This module defines a common features and groupings for
     Transport Layer Security (TLS).

     Copyright (c) 2022 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 Revised
     BSD License set forth in Section 4.c of the IETF Trust's
     Legal Provisions Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC FFFF
     (https://www.rfc-editor.org/info/rfcFFFF); see the RFC
     itself for full legal notices.
```

```
   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 (RFC 2119)
   (RFC 8174) when, and only when, they appear in all
   capitals, as shown here.";

revision 2022-05-24 {
  description
    "Initial version";
  reference
    "RFC FFFF: YANG Groupings for TLS Clients and TLS Servers";
}

// Features

feature tls10 {
  status "obsolete";
  description
    "TLS Protocol Version 1.0 is supported.  TLS 1.0 is obsolete
     and thus it is NOT RECOMMENDED to enable this feature.";
  reference
    "RFC 2246: The TLS Protocol Version 1.0";
}

feature tls11 {
  status "obsolete";
  description
    "TLS Protocol Version 1.1 is supported.  TLS 1.1 is obsolete
     and thus it is NOT RECOMMENDED to enable this feature.";
  reference
    "RFC 4346: The Transport Layer Security (TLS) Protocol
               Version 1.1";
}

feature tls12 {
  status "deprecated";
  description
    "TLS Protocol Version 1.2 is supported  TLS 1.2 is obsolete
     and thus it is NOT RECOMMENDED to enable this feature.";
  reference
    "RFC 5246: The Transport Layer Security (TLS) Protocol
               Version 1.2";
}

feature tls13 {
  description
    "TLS Protocol Version 1.3 is supported.";
```

```
  reference
    "RFC 8446: The Transport Layer Security (TLS)
               Protocol Version 1.3";
}

feature hello-params {
  description
    "TLS hello message parameters are configurable.";
}

feature public-key-generation {
  description
    "Indicates that the server implements the
     'generate-public-key' RPC.";
}

// Identities

identity tls-version-base {
  description
    "Base identity used to identify TLS protocol versions.";
}

identity tls10 {
  if-feature "tls10";
  base tls-version-base;
  status "obsolete";
  description
    "TLS Protocol Version 1.0.";
  reference
    "RFC 2246: The TLS Protocol Version 1.0";
}

identity tls11 {
  if-feature "tls11";
  base tls-version-base;
  status "obsolete";
  description
    "TLS Protocol Version 1.1.";
  reference
    "RFC 4346: The Transport Layer Security (TLS) Protocol
               Version 1.1";
}

identity tls12 {
  if-feature "tls12";
  base tls-version-base;
  status "deprecated";
  description
```

```
    "TLS Protocol Version 1.2.";
  reference
    "RFC 5246: The Transport Layer Security (TLS) Protocol
               Version 1.2";
}

identity tls13 {
  if-feature "tls13";
  base tls-version-base;
  description
    "TLS Protocol Version 1.3.";
  reference
    "RFC 8446: The Transport Layer Security (TLS)
               Protocol Version 1.3";
}

typedef epsk-supported-hash {
  type enumeration {
    enum sha-256 {
      description
        "The SHA-256 Hash.";
    }
    enum sha-384 {
      description
        "The SHA-384 Hash.";
    }
  }
  description
    "As per Section 4.2.11 of RFC 8446, the hash algorithm
     supported by an instance of an External Pre-Shared
     Key (EPSK).";
  reference
    "RFC 8446: The Transport Layer Security (TLS)
               Protocol Version 1.3
     I-D.ietf-tls-external-psk-importer: Importing
               External PSKs for TLS
     I-D.ietf-tls-external-psk-guidance: Guidance
               for External PSK Usage in TLS";
}


// Groupings

grouping hello-params-grouping {
  description
    "A reusable grouping for TLS hello message parameters.";
  reference
    "RFC 5246: The Transport Layer Security (TLS) Protocol
               Version 1.2
```

```
     RFC 8446: The Transport Layer Security (TLS) Protocol
               Version 1.3";
    container tls-versions {
      description
        "Parameters regarding TLS versions.";
      leaf-list tls-version {
        type identityref {
          base tls-version-base;
        }
        description
          "Acceptable TLS protocol versions.

           If this leaf-list is not configured (has zero elements)
           the acceptable TLS protocol versions are implementation-
           defined.";
      }
    }
    container cipher-suites {
      description
        "Parameters regarding cipher suites.";
      leaf-list cipher-suite {
        type identityref {
          base tlscsa:cipher-suite-alg-base;
        }
        ordered-by user;
        description
          "Acceptable cipher suites in order of descending
           preference.  The configured host key algorithms should
           be compatible with the algorithm used by the configured
           private key.  Please see Section 5 of RFC FFFF for
           valid combinations.

           If this leaf-list is not configured (has zero elements)
           the acceptable cipher suites are implementation-
           defined.";
        reference
          "RFC FFFF: YANG Groupings for TLS Clients and TLS Servers";
      }
    }
  } // hello-params-grouping

  rpc generate-public-key {
    if-feature "public-key-generation";
    description
      "Requests the device to generate an public key using
       the specified key algorithm.";
    input {
      leaf algorithm {
        type tlscsa:cipher-suite-algorithm-ref;
```

```
  mandatory true;
  description
    "The cipher suite algorithm that the generated key is
     to work with.  Implementations derive the public key
     algorithm from the cipher suite algorithm.  Example:
     cipher suite 'tls-rsa-with-aes-256-cbc-sha256' maps
     to the RSA public key.";
}
leaf bits {
  type uint16;
  description
    "Specifies the number of bits in the key to create.
     For RSA keys, the minimum size is 1024 bits and
     the default is 3072 bits. Generally, 3072 bits is
     considered sufficient. DSA keys must be exactly 1024
     bits as specified by FIPS 186-2.  For elliptical
     keys, the 'bits' value determines the key length
     of the curve (e.g., 256, 384 or 521), where valid
     values supported by the server are conveyed via an
     unspecified mechanism.  For some public algorithms,
     the keys have a fixed length and the 'bits' value,
     if specified, will be ignored.";
}
choice private-key-encoding {
  default cleartext;
  description
    "A choice amongst optional private key handling.";
  case cleartext {
    leaf cleartext {
      type empty;
      description
        "Indicates that the private key is to be returned
         as a cleartext value.";
    }
  }
  case encrypt {
    if-feature "ct:private-key-encryption";
    container encrypt-with {
      description
         "Indicates that the key is to be encrypted using
          the specified symmetric or asymmetric key.";
      uses ks:encrypted-by-choice-grouping;
    }
  }
  case hide {
    if-feature "ct:hidden-keys";
    leaf hide {
      type empty;
      description
```

```
              "Indicates that the private key is to be hidden.

               Unlike the 'cleartext' and 'encrypt' options, the
               key returned is a placeholder for an internally
               stored key.  See the 'Support for Built-in Keys'
               section in RFC CCCC for information about hidden
               keys.";
          }
        }
      }
    }
    output {
      uses ct:asymmetric-key-pair-grouping;
    }
  } // end generate-public-key

}
```

<CODE ENDS>

3. The "ietf-tls-client" Module

This section defines a YANG 1.1 [RFC7950] module called "ietf-tls-client". A high-level overview of the module is provided in Section 3.1. Examples illustrating the module's use are provided in Examples (Section 3.2). The YANG module itself is defined in Section 3.3.

3.1. Data Model Overview

This section provides an overview of the "ietf-tls-client" module in terms of its features and groupings.

3.1.1. Features

The following diagram lists all the "feature" statements defined in the "ietf-tls-client" module:

```
Features:
  +-- tls-client-keepalives
  +-- client-ident-x509-cert
  +-- client-ident-raw-public-key
  +-- client-ident-psk
  +-- server-auth-x509-cert
  +-- server-auth-raw-public-key
  +-- server-auth-psk
```

The diagram above uses syntax that is similar to but not defined in [RFC8340].

3.1.2. Groupings

The "ietf-tls-client" module defines the following "grouping" statement:

*tls-client-grouping

This grouping is presented in the following subsection.

3.1.2.1. The "tls-client-grouping" Grouping

The following tree diagram [RFC8340] illustrates the "tls-client-grouping" grouping:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

  grouping tls-client-grouping:
    +-- client-identity!
    |  +-- (auth-type)
    |     +--:(certificate) {client-ident-x509-cert}?
    |     |  +-- certificate
    |     |     +---u ks:local-or-keystore-end-entity-cert-with-key-\
grouping
    |     +--:(raw-public-key) {client-ident-raw-public-key}?
    |     |  +-- raw-private-key
    |     |     +---u ks:local-or-keystore-asymmetric-key-grouping
    |     +--:(tls12-psk) {client-ident-tls12-psk}?
    |     |  +-- tls12-psk
    |     |     +---u ks:local-or-keystore-symmetric-key-grouping
    |     |     +-- id?
    |     |             string
    |     +--:(tls13-epsk) {client-ident-tls13-epsk}?
    |        +-- tls13-epsk
    |           +---u ks:local-or-keystore-symmetric-key-grouping
    |           +-- external-identity
    |           |       string
    |           +-- hash
    |           |       tlscmn:epsk-supported-hash
    |           +-- context?
    |           |       string
    |           +-- target-protocol?
    |           |       uint16
    |           +-- target-kdf?
    |                   uint16
    +-- server-authentication
    |  +-- ca-certs! {server-auth-x509-cert}?
    |  |  +---u ts:local-or-truststore-certs-grouping
    |  +-- ee-certs! {server-auth-x509-cert}?
    |  |  +---u ts:local-or-truststore-certs-grouping
    |  +-- raw-public-keys! {server-auth-raw-public-key}?
    |  |  +---u ts:local-or-truststore-public-keys-grouping
    |  +-- tls12-psks?        empty {server-auth-tls12-psk}?
    |  +-- tls13-epsks?       empty {server-auth-tls13-epsk}?
    +-- hello-params {tlscmn:hello-params}?
    |  +---u tlscmn:hello-params-grouping
    +-- keepalives {tls-client-keepalives}?
       +-- peer-allowed-to-send?   empty
       +-- test-peer-aliveness!
          +-- max-wait?       uint16
          +-- max-attempts?   uint8
```

Comments:

* The "client-identity" node, which is optionally configured (as client authentication MAY occur at a higher protocol layer), configures identity credentials, each enabled by a "feature" statement defined in Section 3.1.1.

* The "server-authentication" node configures trust anchors for authenticating the TLS server, with each option enabled by a "feature" statement.

* The "hello-params" node, which must be enabled by a feature, configures parameters for the TLS sessions established by this configuration.

* The "keepalives" node, which must be enabled by a feature, configures a "presence" container for testing the aliveness of the TLS server. The aliveness-test occurs at the TLS protocol layer.

* For the referenced grouping statement(s):

  - The "local-or-keystore-end-entity-cert-with-key-grouping" grouping is discussed in Section 2.1.3.6 of [I-D.ietf-netconf-keystore].
  - The "local-or-keystore-asymmetric-key-grouping" grouping is discussed in Section 2.1.3.4 of [I-D.ietf-netconf-keystore].
  - The "local-or-keystore-symmetric-key-grouping" grouping is discussed in Section 2.1.3.3 of [I-D.ietf-netconf-keystore].
  - The "local-or-truststore-certs-grouping" grouping is discussed in Section 2.1.3.1 of [I-D.ietf-netconf-trust-anchors].
  - The "local-or-truststore-public-keys-grouping" grouping is discussed in Section 2.1.3.2 of [I-D.ietf-netconf-trust-anchors].
  - The "hello-params-grouping" grouping is discussed in Section 2.1.3.1 in this document.

3.1.3. Protocol-accessible Nodes

The "ietf-tls-client" module defines only "grouping" statements that are used by other modules to instantiate protocol-accessible nodes.

3.2. Example Usage

This section presents two examples showing the "tls-client-grouping" grouping populated with some data. These examples are effectively the same except the first configures the client identity using a local key while the second uses a key configured in a keystore. Both examples are consistent with the examples presented in Section 2 of

[I-D.ietf-netconf-trust-anchors] and Section 3.2 of [I-D.ietf-netconf-keystore].

The following configuration example uses local-definitions for the client identity and server authentication:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

<!-- The outermost element below doesn't exist in the data model. -->
<!--  It simulates if the "grouping" were a "container" instead.  -->

<tls-client
  xmlns="urn:ietf:params:xml:ns:yang:ietf-tls-client"
  xmlns:ct="urn:ietf:params:xml:ns:yang:ietf-crypto-types">
    <!-- how this client will authenticate itself to the server -->
    <client-identity>
        <certificate>
            <local-definition>
                <public-key-format>ct:subject-public-key-info-format\
</public-key-format>
                <public-key>BASE64VALUE=</public-key>
                <private-key-format>ct:rsa-private-key-format</priva\
te-key-format>
                <cleartext-private-key>BASE64VALUE=</cleartext-priva\
te-key>
                <cert-data>BASE64VALUE=</cert-data>
            </local-definition>
        </certificate>
        <!-- TESTED, BUT COMMENTED OUT DUE TO ONLY ONE ALLOWED AT A \
TIME
        <raw-private-key>
         <local-definition>
            <public-key-format>ct:subject-public-key-info-format</pu\
blic-key-format>
            <public-key>BASE64VALUE=</public-key>
            <private-key-format>ct:rsa-private-key-format</private-k\
ey-format>
            <cleartext-private-key>BASE64VALUE=</cleartext-private-k\
ey>
          </local-definition>
        </raw-private-key>
        -->
        <!-- USE ONLY ONE AT A TIME
        <tls12-psk>
            <local-definition>
                <key-format>ct:octet-string-key-format</key-format>
                <cleartext-key>BASE64VALUE=</cleartext-key>
            </local-definition>
            <id>example_id_string</id>
        </tls12-psk>
        -->
        <!-- USE ONLY ONE AT A TIME
        <tls13-epsk>
            <local-definition>
                <key-format>ct:octet-string-key-format</key-format>
```

```
                <cleartext-key>BASE64VALUE=</cleartext-key>
            </local-definition>
            <external-identity>example_external_id</external-identit\
y>
            <hash>sha-256</hash>
            <context>example_context_string</context>
            <target-protocol>8443</target-protocol>
            <target-kdf>12345</target-kdf>
        </tls13-epsk>
        -->
    </client-identity>
    <!-- which certificates will this client trust -->
    <server-authentication>
        <ca-certs>
            <local-definition>
                <certificate>
                    <name>Server Cert Issuer #1</name>
                    <cert-data>BASE64VALUE=</cert-data>
                </certificate>
                <certificate>
                    <name>Server Cert Issuer #2</name>
                    <cert-data>BASE64VALUE=</cert-data>
                </certificate>
            </local-definition>
        </ca-certs>
        <ee-certs>
            <local-definition>
                <certificate>
                    <name>My Application #1</name>
                    <cert-data>BASE64VALUE=</cert-data>
                </certificate>
                <certificate>
                    <name>My Application #2</name>
                    <cert-data>BASE64VALUE=</cert-data>
                </certificate>
            </local-definition>
        </ee-certs>
        <raw-public-keys>
            <local-definition>
                <public-key>
                    <name>corp-fw1</name>
                    <public-key-format>ct:subject-public-key-info-fo\
rmat</public-key-format>
                    <public-key>BASE64VALUE=</public-key>
                </public-key>
                <public-key>
                    <name>corp-fw2</name>
                    <public-key-format>ct:subject-public-key-info-fo\
rmat</public-key-format>
```

```
                    <public-key>BASE64VALUE=</public-key>
                </public-key>
            </local-definition>
        </raw-public-keys>
        <tls12-psks/>
        <tls13-epsks/>
    </server-authentication>
    <keepalives>
        <test-peer-aliveness>
            <max-wait>30</max-wait>
            <max-attempts>3</max-attempts>
        </test-peer-aliveness>
    </keepalives>
</tls-client>
```

The following configuration example uses keystore-references for the client identity and truststore-references for server authentication: from the keystore:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

<!-- The outermost element below doesn't exist in the data model. -->
<!--  It simulates if the "grouping" were a "container" instead.  -->

<tls-client xmlns="urn:ietf:params:xml:ns:yang:ietf-tls-client">
    <!-- how this client will authenticate itself to the server -->
    <client-identity>
        <certificate>
            <keystore-reference>
                <asymmetric-key>rsa-asymmetric-key</asymmetric-key>
                <certificate>ex-rsa-cert</certificate>
            </keystore-reference>
        </certificate>
        <!-- TESTED, BUT COMMENTED OUT DUE TO ONLY ONE ALLOWED AT A \
TIME
        <raw-private-key>
          <keystore-reference>raw-private-key</keystore-reference>
        </raw-private-key>
        -->
        <!-- USE ONLY ONE AT A TIME
        <tls12-psk>
            <keystore-reference>encrypted-symmetric-key</keystore-re\
ference>
            <id>example_id_string</id>
        </tls12-psk>
        -->
        <!-- USE ONLY ONE AT A TIME
        <tls13-epsk>
            <keystore-reference>encrypted-symmetric-key</keystore-re\
ference>
            <external-identity>example_external_id</external-identit\
y>
            <hash>sha-256</hash>
            <context>example_context_string</context>
            <target-protocol>8443</target-protocol>
            <target-kdf>12345</target-kdf>
        </tls13-epsk>
        -->
    </client-identity>
    <!-- which certificates will this client trust -->
    <server-authentication>
        <ca-certs>
            <truststore-reference>trusted-server-ca-certs</truststor\
e-reference>
        </ca-certs>
        <ee-certs>
            <truststore-reference>trusted-server-ee-certs</truststor\
e-reference>
```

```
        </ee-certs>
        <raw-public-keys>
            <truststore-reference>Raw Public Keys for TLS Servers</t\
ruststore-reference>
        </raw-public-keys>
        <tls12-psks/>
        <tls13-epsks/>
    </server-authentication>
    <keepalives>
        <test-peer-aliveness>
            <max-wait>30</max-wait>
            <max-attempts>3</max-attempts>
        </test-peer-aliveness>
    </keepalives>
</tls-client>
```

3.3. YANG Module

This YANG module has normative references to [I-D.ietf-netconf-trust-anchors] and [I-D.ietf-netconf-keystore], and Informative references to [RFC5246], [RFC8446], [I-D.ietf-tls-external-psk-importer] and [I-D.ietf-tls-external-psk-guidance].

<CODE BEGINS> file "ietf-tls-client@2022-05-24.yang"

```
module ietf-tls-client {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tls-client";
  prefix tlsc;

  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control Model";
  }

  import ietf-crypto-types {
    prefix ct;
    reference
      "RFC AAAA: YANG Data Types and Groupings for Cryptography";
  }

  import ietf-truststore {
    prefix ts;
    reference
      "RFC BBBB: A YANG Data Model for a Truststore";
  }

  import ietf-keystore {
    prefix ks;
    reference
      "RFC CCCC: A YANG Data Model for a Keystore";
  }

  import ietf-tls-common {
    prefix tlscmn;
    revision-date 2022-05-24; // stable grouping definitions
    reference
      "RFC FFFF: YANG Groupings for TLS Clients and TLS Servers";
  }

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

  contact
    "WG List:  NETCONF WG list <mailto:netconf@ietf.org>
     WG Web:   https://datatracker.ietf.org/wg/netconf
     Author:   Kent Watsen <mailto:kent+ietf@watsen.net>
     Author:   Jeff Hartley <mailto:jeff.hartley@commscope.com>
     Author:   Gary Wu <mailto:garywu@cisco.com>";

  description
    "This module defines reusable groupings for TLS clients that
     can be used as a basis for specific TLS client instances.
```

```
   Copyright (c) 2022 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 Revised
   BSD License set forth in Section 4.c of the IETF Trust's
   Legal Provisions Relating to IETF Documents
   (https://trustee.ietf.org/license-info).

   This version of this YANG module is part of RFC FFFF
   (https://www.rfc-editor.org/info/rfcFFFF); see the RFC
   itself for full legal notices.

   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 (RFC 2119)
   (RFC 8174) when, and only when, they appear in all
   capitals, as shown here.";

 revision 2022-05-24 {
   description
     "Initial version";
   reference
     "RFC FFFF: YANG Groupings for TLS Clients and TLS Servers";
 }

 // Features

 feature tls-client-keepalives {
   description
     "Per socket TLS keepalive parameters are configurable for
      TLS clients on the server implementing this feature.";
 }

 feature client-ident-x509-cert {
   description
     "Indicates that the client supports identifying itself
      using X.509 certificates.";
   reference
     "RFC 5280:
        Internet X.509 Public Key Infrastructure Certificate
        and Certificate Revocation List (CRL) Profile";
 }

 feature client-ident-raw-public-key {
   description
     "Indicates that the client supports identifying itself
      using raw public keys.";
```

```
  reference
    "RFC 7250:
       Using Raw Public Keys in Transport Layer Security (TLS)
       and Datagram Transport Layer Security (DTLS)";
}

feature client-ident-tls12-psk {
  description
    "Indicates that the client supports identifying itself
     using TLS-1.2 PSKs (pre-shared or pairwise-symmetric keys).";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for Transport Layer Security
       (TLS)";
}

feature client-ident-tls13-epsk {
  description
    "Indicates that the client supports identifying itself
     using TLS-1.3 External PSKs (pre-shared keys).";
  reference
    "RFC 8446:
       The Transport Layer Security (TLS) Protocol Version 1.3";
}

feature server-auth-x509-cert {
  description
    "Indicates that the client supports authenticating servers
     using X.509 certificates.";
  reference
    "RFC 5280:
       Internet X.509 Public Key Infrastructure Certificate
       and Certificate Revocation List (CRL) Profile";
}

feature server-auth-raw-public-key {
  description
    "Indicates that the client supports authenticating servers
     using raw public keys.";
  reference
    "RFC 7250:
       Using Raw Public Keys in Transport Layer Security (TLS)
       and Datagram Transport Layer Security (DTLS)";
}

feature server-auth-tls12-psk {
  description
    "Indicates that the client supports authenticating servers
     using PSKs (pre-shared or pairwise-symmetric keys).";
```

```
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for Transport Layer Security
       (TLS)";
}

feature server-auth-tls13-epsk {
  description
    "Indicates that the client supports authenticating servers
     using TLS-1.3 External PSKs (pre-shared keys).";
  reference
    "RFC 8446:
       The Transport Layer Security (TLS) Protocol Version 1.3";
}

// Groupings

grouping tls-client-grouping {
  description
    "A reusable grouping for configuring a TLS client without
     any consideration for how an underlying TCP session is
     established.

     Note that this grouping uses fairly typical descendant
     node names such that a stack of 'uses' statements will
     have name conflicts.  It is intended that the consuming
     data model will resolve the issue (e.g., by wrapping
     the 'uses' statement in a container called
     'tls-client-parameters').  This model purposely does
     not do this itself so as to provide maximum flexibility
     to consuming models.";

  container client-identity {
    nacm:default-deny-write;
    presence
      "Indicates that a TLS-level client identity has been
       configured.  This statement is present so the mandatory
       descendant do not imply that this node must be configured.";
    description
      "Identity credentials the TLS client MAY present when
       establishing a connection to a TLS server.  If not
       configured, then client authentication is presumed to
       occur a protocol layer above TLS.  When configured,
       and requested by the TLS server when establishing a
       TLS session, these credentials are passed in the
       Certificate message defined in Section 7.4.2 of
       RFC 5246 and Section 4.4.2 in RFC 8446.";
    reference
      "RFC 5246: The Transport Layer Security (TLS)
```

```
            Protocol Version 1.2
   RFC 8446: The Transport Layer Security (TLS)
            Protocol Version 1.3
   RFC CCCC: A YANG Data Model for a Keystore";
choice auth-type {
  mandatory true;
  description
    "A choice amongst authentication types, of which one must
     be enabled (via its associated 'feature') and selected.";
  case certificate {
    if-feature "client-ident-x509-cert";
    container certificate {
      description
        "Specifies the client identity using a certificate.";
      uses
        ks:local-or-keystore-end-entity-cert-with-key-grouping{
        refine "local-or-keystore/local/local-definition" {
          must 'public-key-format'
             + ' = "ct:subject-public-key-info-format"';
        }
        refine "local-or-keystore/keystore/keystore-reference"
             + "/asymmetric-key" {
          must 'deref(.)/../ks:public-key-format'
             + ' = "ct:subject-public-key-info-format"';
        }
      }
    }
  }
  case raw-public-key {
    if-feature "client-ident-raw-public-key";
    container raw-private-key {
      description
        "Specifies the client identity using a raw
         private key.";
      uses ks:local-or-keystore-asymmetric-key-grouping {
        refine "local-or-keystore/local/local-definition" {
          must 'public-key-format'
             + ' = "ct:subject-public-key-info-format"';
        }
        refine "local-or-keystore/keystore"
             + "/keystore-reference" {
          must 'deref(.)/../ks:public-key-format'
             + ' = "ct:subject-public-key-info-format"';
        }
      }
    }
  }
  case tls12-psk {
    if-feature "client-ident-tls12-psk";
```

```
  container tls12-psk {
    description
      "Specifies the client identity using a PSK (pre-shared
       or pairwise-symmetric key).";
    uses ks:local-or-keystore-symmetric-key-grouping;
    leaf id {
      type string;
      description
        "The key 'psk_identity' value used in the TLS
         'ClientKeyExchange' message.";
      reference
        "RFC 4279: Pre-Shared Key Ciphersuites for
                   Transport Layer Security (TLS)";
    }
  }
}
case tls13-epsk {
  if-feature "client-ident-tls13-epsk";
  container tls13-epsk {
    description
      "An External Pre-Shared Key (EPSK) is established
      or provisioned out-of-band, i.e., not from a TLS
      connection.  An EPSK is a tuple of (Base Key,
      External Identity, Hash).  External PSKs MUST NOT
      be imported for (D)TLS 1.2 or prior versions.  When
      PSKs are provisioned out of band, the PSK identity
      and the KDF hash algorithm to be used with the PSK
      MUST also be provisioned.

      The structure of this container is designed
      to satisfy the requirements of RFC 8446
      Section 4.2.11, the recommendations from I-D
      ietf-tls-external-psk-guidance Section 6,
      and the EPSK input fields detailed in I-D
      draft-ietf-tls-external-psk-importer
      Section 3.1.  The base-key is based upon
      ks:local-or-keystore-symmetric-key-grouping
      in order to provide users with flexible and
      secure storage options.";
    reference
      "RFC 8446: The Transport Layer Security (TLS)
                 Protocol Version 1.3
       I-D.ietf-tls-external-psk-importer:
                 Importing External PSKs for TLS
       I-D.ietf-tls-external-psk-guidance:
                 Guidance for External PSK Usage in TLS";
    uses ks:local-or-keystore-symmetric-key-grouping;
    leaf external-identity {
      type string;
```

```
  mandatory true;
  description
    "As per Section 4.2.11 of RFC 8446, and Section 4.1
     of I-D. ietf-tls-external-psk-guidance:
     A sequence of bytes used to identify an EPSK. A
     label for a pre-shared key established externally.";
  reference
    "RFC 8446: The Transport Layer Security (TLS)
               Protocol Version 1.3
     I-D.ietf-tls-external-psk-guidance:
               Guidance for External PSK Usage in TLS";
}
leaf hash {
  type tlscmn:epsk-supported-hash;
  mandatory true;
  description
    "As per Section 4.2.11 of RFC 8446, for externally
     established PSKs, the Hash algorithm MUST be set
     when the PSK is established or default to SHA-256
     if no such algorithm is defined.  The server MUST
     ensure that it selects a compatible PSK (if any)
     and cipher suite.  Each PSK MUST only be used with
     a single hash function.";
  reference
    "RFC 8446: The Transport Layer Security (TLS)
               Protocol Version 1.3";
}
leaf context {
  type string;
  description
    "As per Section 4.1 of I-D.
     ietf-tls-external-psk-guidance: Context may include
     information about peer roles or identities to
     mitigate Selfie-style reflection attacks [Selfie].
     If the EPSK is a key derived from some other
     protocol or sequence of protocols, context
     MUST include a channel binding for the deriving
     protocols [RFC5056].  The details of this binding
     are protocol specific.";
  reference
    "I-D.ietf-tls-external-psk-importer:
               Importing External PSKs for TLS
     I-D.ietf-tls-external-psk-guidance:
               Guidance for External PSK Usage in TLS";
}
leaf target-protocol {
  type uint16;
  description
    "As per Section 3.1 of I-D.
```

```
            ietf-tls-external-psk-guidance:
            The protocol for which a PSK is imported for use.";
          reference
            "I-D.ietf-tls-external-psk-importer:
                      Importing External PSKs for TLS";
        }
        leaf target-kdf {
          type uint16;
          description
            "As per Section 3.1 of I-D.
             ietf-tls-external-psk-guidance:
             The specific Key Derivation Function (KDF) for which
             a PSK is imported for use.";
          reference
            "I-D.ietf-tls-external-psk-importer:
                      Importing External PSKs for TLS";
        }
      }
    }
  }
} // container client-identity

container server-authentication {
  nacm:default-deny-write;
  must 'ca-certs or ee-certs or raw-public-keys or tls12-psks
    or tls13-epsks';
  description
    "Specifies how the TLS client can authenticate TLS servers.
     Any combination of credentials is additive and unordered.

     Note that no configuration is required for PSK (pre-shared
     or pairwise-symmetric key) based authentication as the key
     is necessarily the same as configured in the '../client-
     identity' node.";
  container ca-certs {
    if-feature "server-auth-x509-cert";
    presence
      "Indicates that CA certificates have been configured.
       This statement is present so the mandatory descendant
       nodes do not imply that this node must be configured.";
    description
      "A set of certificate authority (CA) certificates used by
       the TLS client to authenticate TLS server certificates.
       A server certificate is authenticated if it has a valid
       chain of trust to a configured CA certificate.";
    reference
      "RFC BBBB: A YANG Data Model for a Truststore";
    uses ts:local-or-truststore-certs-grouping;
  }
```

```
container ee-certs {
  if-feature "server-auth-x509-cert";
  presence
    "Indicates that EE certificates have been configured.
     This statement is present so the mandatory descendant
     nodes do not imply that this node must be configured.";
  description
    "A set of server certificates (i.e., end entity
     certificates) used by the TLS client to authenticate
     certificates presented by TLS servers.  A server
     certificate is authenticated if it is an exact
     match to a configured server certificate.";
  reference
    "RFC BBBB: A YANG Data Model for a Truststore";
  uses ts:local-or-truststore-certs-grouping;
}
container raw-public-keys {
  if-feature "server-auth-raw-public-key";
  presence
    "Indicates that raw public keys have been configured.
     This statement is present so the mandatory descendant
     nodes do not imply that this node must be configured.";
  description
    "A set of raw public keys used by the TLS client to
     authenticate raw public keys presented by the TLS
     server.  A raw public key is authenticated if it
     is an exact match to a configured raw public key.";
  reference
    "RFC BBBB: A YANG Data Model for a Truststore";
  uses ts:local-or-truststore-public-keys-grouping {
    refine "local-or-truststore/local/local-definition"
         + "/public-key" {
      must 'public-key-format'
         + ' = "ct:subject-public-key-info-format"';
    }
    refine "local-or-truststore/truststore"
         + "/truststore-reference" {
      must 'deref(.)/../*/ts:public-key-format'
         + ' = "ct:subject-public-key-info-format"';
    }
  }
}
leaf tls12-psks {
  if-feature "server-auth-tls12-psk";
  type empty;
  description
    "Indicates that the TLS client can authenticate TLS servers
     using configure PSKs (pre-shared or pairwise-symmetric
     keys).
```

```
        No configuration is required since the PSK value is the
        same as PSK value configured in the 'client-identity'
        node.";
    }
    leaf tls13-epsks {
      if-feature "server-auth-tls13-epsk";
      type empty;
      description
        "Indicates that the TLS client can authenticate TLS servers
         using configured external PSKs (pre-shared keys).

         No configuration is required since the PSK value is the
         same as PSK value configured in the 'client-identity'
         node.";
    }
  } // container server-authentication

  container hello-params {
    nacm:default-deny-write;
    if-feature "tlscmn:hello-params";
    uses tlscmn:hello-params-grouping;
    description
      "Configurable parameters for the TLS hello message.";
  } // container hello-params

  container keepalives {
    nacm:default-deny-write;
    if-feature "tls-client-keepalives";
    description
      "Configures the keepalive policy for the TLS client.";
    leaf peer-allowed-to-send {
      type empty;
      description
        "Indicates that the remote TLS server is allowed to send
         HeartbeatRequest messages, as defined by RFC 6520
         to this TLS client.";
      reference
        "RFC 6520: Transport Layer Security (TLS) and Datagram
         Transport Layer Security (DTLS) Heartbeat Extension";
    }
    container test-peer-aliveness {
      presence
        "Indicates that the TLS client proactively tests the
         aliveness of the remote TLS server.";
      description
        "Configures the keep-alive policy to proactively test
         the aliveness of the TLS server.  An unresponsive
         TLS server is dropped after approximately max-wait
```

```
           * max-attempts seconds.  The TLS client MUST send
           HeartbeatRequest messages, as defined by RFC 6520.";
        reference
          "RFC 6520: Transport Layer Security (TLS) and Datagram
           Transport Layer Security (DTLS) Heartbeat Extension";
        leaf max-wait {
          type uint16 {
            range "1..max";
          }
          units "seconds";
          default "30";
          description
            "Sets the amount of time in seconds after which if
             no data has been received from the TLS server, a
             TLS-level message will be sent to test the
             aliveness of the TLS server.";
        }
        leaf max-attempts {
          type uint8;
          default "3";
          description
            "Sets the maximum number of sequential keep-alive
             messages that can fail to obtain a response from
             the TLS server before assuming the TLS server is
             no longer alive.";
        }
      }
    }
  } // grouping tls-client-grouping

}
```

<CODE ENDS>

4. The "ietf-tls-server" Module

This section defines a YANG 1.1 module called "ietf-tls-server". A high-level overview of the module is provided in Section 4.1. Examples illustrating the module's use are provided in Examples (Section 4.2). The YANG module itself is defined in Section 4.3.

4.1. Data Model Overview

This section provides an overview of the "ietf-tls-server" module in terms of its features and groupings.

4.1.1. Features

The following diagram lists all the "feature" statements defined in the "ietf-tls-server" module:

```
Features:
  +-- tls-server-keepalives
  +-- server-ident-x509-cert
  +-- server-ident-raw-public-key
  +-- server-ident-psk
  +-- client-auth-supported
  +-- client-auth-x509-cert
  +-- client-auth-raw-public-key
  +-- client-auth-psk
```

The diagram above uses syntax that is similar to but not defined in [RFC8340].

4.1.2. Groupings

The "ietf-tls-server" module defines the following "grouping" statement:

```
  *tls-server-grouping
```

This grouping is presented in the following subsection.

4.1.2.1. The "tls-server-grouping" Grouping

The following tree diagram [RFC8340] illustrates the "tls-server-grouping" grouping:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

  grouping tls-server-grouping:
    +-- server-identity
    |  +-- (auth-type)
    |     +--:(certificate) {server-ident-x509-cert}?
    |     |  +-- certificate
    |     |     +---u ks:local-or-keystore-end-entity-cert-with-key-\
grouping
    |     +--:(raw-private-key) {server-ident-raw-public-key}?
    |     |  +-- raw-private-key
    |     |     +---u ks:local-or-keystore-asymmetric-key-grouping
    |     +--:(tls12-psk) {server-ident-tls12-psk}?
    |     |  +-- tls12-psk
    |     |     +---u ks:local-or-keystore-symmetric-key-grouping
    |     |     +-- id_hint?
    |     |             string
    |     +--:(tls13-epsk) {server-ident-tls13-epsk}?
    |        +-- tls13-epsk
    |           +---u ks:local-or-keystore-symmetric-key-grouping
    |           +-- external-identity
    |           |       string
    |           +-- hash
    |           |       tlscmn:epsk-supported-hash
    |           +-- context?
    |           |       string
    |           +-- target-protocol?
    |           |       uint16
    |           +-- target-kdf?
    |                   uint16
    +-- client-authentication! {client-auth-supported}?
    |  +-- ca-certs! {client-auth-x509-cert}?
    |  |  +---u ts:local-or-truststore-certs-grouping
    |  +-- ee-certs! {client-auth-x509-cert}?
    |  |  +---u ts:local-or-truststore-certs-grouping
    |  +-- raw-public-keys! {client-auth-raw-public-key}?
    |  |  +---u ts:local-or-truststore-public-keys-grouping
    |  +-- tls12-psks?        empty {client-auth-tls12-psk}?
    |  +-- tls13-epsks?       empty {client-auth-tls13-epsk}?
    +-- hello-params {tlscmn:hello-params}?
    |  +---u tlscmn:hello-params-grouping
    +-- keepalives {tls-server-keepalives}?
       +-- peer-allowed-to-send?   empty
       +-- test-peer-aliveness!
          +-- max-wait?       uint16
          +-- max-attempts?   uint8
```

Comments:

* The "server-identity" node configures identity credentials, each of which is enabled by a "feature".

* The "client-authentication" node, which is optionally configured (as client authentication MAY occur at a higher protocol layer), configures trust anchors for authenticating the TLS client, with each option enabled by a "feature" statement.

* The "hello-params" node, which must be enabled by a feature, configures parameters for the TLS sessions established by this configuration.

* The "keepalives" node, which must be enabled by a feature, configures a flag enabling the TLS client to test the aliveness of the TLS server, as well as a "presence" container for testing the aliveness of the TLSi client. The aliveness-tests occurs at the TLS protocol layer.

* For the referenced grouping statement(s):

  - The "local-or-keystore-end-entity-cert-with-key-grouping" grouping is discussed in Section 2.1.3.6 of [I-D.ietf-netconf-keystore].
  - The "local-or-keystore-asymmetric-key-grouping" grouping is discussed in Section 2.1.3.4 of [I-D.ietf-netconf-keystore].
  - The "local-or-keystore-symmetric-key-grouping" grouping is discussed in Section 2.1.3.3 of [I-D.ietf-netconf-keystore].
  - The "local-or-truststore-public-keys-grouping" grouping is discussed in Section 2.1.3.2 of [I-D.ietf-netconf-trust-anchors].
  - The "local-or-truststore-certs-grouping" grouping is discussed in Section 2.1.3.1 of [I-D.ietf-netconf-trust-anchors].
  - The "hello-params-grouping" grouping is discussed in Section 2.1.3.1 in this document.

4.1.3. Protocol-accessible Nodes

The "ietf-tls-server" module defines only "grouping" statements that are used by other modules to instantiate protocol-accessible nodes.

4.2. Example Usage

This section presents two examples showing the "tls-server-grouping" grouping populated with some data. These examples are effectively the same except the first configures the server identity using a local key while the second uses a key configured in a keystore. Both examples are consistent with the examples presented in Section 2 of

[I-D.ietf-netconf-trust-anchors] and Section 3.2 of [I-D.ietf-netconf-keystore].

The following configuration example uses local-definitions for the server identity and client authentication:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

<!-- The outermost element below doesn't exist in the data model. -->
<!--  It simulates if the "grouping" were a "container" instead.  -->

<tls-server
  xmlns="urn:ietf:params:xml:ns:yang:ietf-tls-server"
  xmlns:ct="urn:ietf:params:xml:ns:yang:ietf-crypto-types">
    <!-- how this server will authenticate itself to the client -->
    <server-identity>
        <certificate>
            <local-definition>
                <public-key-format>ct:subject-public-key-info-format\
</public-key-format>
                <public-key>BASE64VALUE=</public-key>
                <private-key-format>ct:rsa-private-key-format</priva\
te-key-format>
                <cleartext-private-key>BASE64VALUE=</cleartext-priva\
te-key>
                <cert-data>BASE64VALUE=</cert-data>
            </local-definition>
        </certificate>
        <!-- TESTED, BUT COMMENTED OUT DUE TO ONLY ONE ALLOWED AT A \
TIME
        <raw-private-key>
          <local-definition>
            <public-key-format>ct:subject-public-key-info-format</pu\
blic-key-format>
            <public-key>BASE64VALUE=</public-key>
            <private-key-format>ct:rsa-private-key-format</private-k\
ey-format>
            <cleartext-private-key>BASE64VALUE=</cleartext-private-k\
ey>
          </local-definition>
        </raw-private-key>
        -->
        <!-- USE ONLY ONE AT A TIME
        <tls12-psk>
            <local-definition>
                <key-format>ct:octet-string-key-format</key-format>
                <cleartext-key>BASE64VALUE=</cleartext-key>
            </local-definition>
            <id_hint>example_id_hint</id_hint>
        </tls12-psk>
        -->
        <!-- USE ONLY ONE AT A TIME
        <tls13-epsk>
            <local-definition>
                <key-format>ct:octet-string-key-format</key-format>
```

```
                <cleartext-key>BASE64VALUE=</cleartext-key>
            </local-definition>
            <external-identity>example_external_id</external-identit\
y>
            <hash>sha-256</hash>
            <context>example_context_string</context>
            <target-protocol>8443</target-protocol>
            <target-kdf>12345</target-kdf>
        </tls13-epsk>
        -->
    </server-identity>
    <!-- which certificates will this server trust -->
    <client-authentication>
        <ca-certs>
            <local-definition>
                <certificate>
                    <name>Identity Cert Issuer #1</name>
                    <cert-data>BASE64VALUE=</cert-data>
                </certificate>
                <certificate>
                    <name>Identity Cert Issuer #2</name>
                    <cert-data>BASE64VALUE=</cert-data>
                </certificate>
            </local-definition>
        </ca-certs>
        <ee-certs>
            <local-definition>
                <certificate>
                    <name>Application #1</name>
                    <cert-data>BASE64VALUE=</cert-data>
                </certificate>
                <certificate>
                    <name>Application #2</name>
                    <cert-data>BASE64VALUE=</cert-data>
                </certificate>
            </local-definition>
        </ee-certs>
        <raw-public-keys>
            <local-definition>
                <public-key>
                    <name>User A</name>
                    <public-key-format>ct:subject-public-key-info-fo\
rmat</public-key-format>
                    <public-key>BASE64VALUE=</public-key>
                </public-key>
                <public-key>
                    <name>User B</name>
                    <public-key-format>ct:subject-public-key-info-fo\
rmat</public-key-format>
```

```
                    <public-key>BASE64VALUE=</public-key>
                </public-key>
            </local-definition>
        </raw-public-keys>
        <tls12-psks/>
        <tls13-epsks/>
    </client-authentication>
    <keepalives>
        <peer-allowed-to-send/>
    </keepalives>
</tls-server>
```

The following configuration example uses keystore-references for the server identity and truststore-references for client authentication: from the keystore:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

<!-- The outermost element below doesn't exist in the data model. -->
<!--  It simulates if the "grouping" were a "container" instead.  -->

<tls-server xmlns="urn:ietf:params:xml:ns:yang:ietf-tls-server">
    <!-- how this server will authenticate itself to the client -->
    <server-identity>
        <certificate>
            <keystore-reference>
                <asymmetric-key>rsa-asymmetric-key</asymmetric-key>
                <certificate>ex-rsa-cert</certificate>
            </keystore-reference>
        </certificate>
        <!-- TESTED, BUT COMMENTED OUT DUE TO ONLY ONE ALLOWED AT A \
TIME
        <raw-private-key>
          <keystore-reference>raw-private-key</keystore-reference>
        </raw-private-key>
        -->
        <!-- USE ONLY ONE AT A TIME
        <tls12-psk>
            <keystore-reference>encrypted-symmetric-key</keystore-re\
ference>
            <id_hint>example_id_hint</id_hint>
        </tls12-psk>
        -->
        <!-- USE ONLY ONE AT A TIME
        <tls13-epsk>
            <keystore-reference>encrypted-symmetric-key</keystore-re\
ference>
            <external-identity>example_external_id</external-identit\
y>
            <hash>sha-256</hash>
            <context>example_context_string</context>
            <target-protocol>8443</target-protocol>
            <target-kdf>12345</target-kdf>
        </tls13-epsk>
        -->
    </server-identity>
    <!-- which certificates will this server trust -->
    <client-authentication>
        <ca-certs>
            <truststore-reference>trusted-client-ca-certs</truststor\
e-reference>
        </ca-certs>
        <ee-certs>
            <truststore-reference>trusted-client-ee-certs</truststor\
e-reference>
```

```
        </ee-certs>
        <raw-public-keys>
            <truststore-reference>Raw Public Keys for TLS Clients</t\
ruststore-reference>
        </raw-public-keys>
        <tls12-psks/>
        <tls13-epsks/>
    </client-authentication>
    <keepalives>
        <peer-allowed-to-send/>
    </keepalives>
</tls-server>
```

4.3. YANG Module

This YANG module has normative references to [I-D.ietf-netconf-trust-anchors] and [I-D.ietf-netconf-keystore], and Informative references to [RFC5246], [RFC8446], [I-D.ietf-tls-external-psk-importer] and [I-D.ietf-tls-external-psk-guidance].

<CODE BEGINS> file "ietf-tls-server@2022-05-24.yang"

```
module ietf-tls-server {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tls-server";
  prefix tlss;

  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control Model";
  }

  import ietf-crypto-types {
    prefix ct;
    reference
      "RFC AAAA: YANG Data Types and Groupings for Cryptography";
  }

  import ietf-truststore {
    prefix ts;
    reference
      "RFC BBBB: A YANG Data Model for a Truststore";
  }

  import ietf-keystore {
    prefix ks;
    reference
      "RFC CCCC: A YANG Data Model for a Keystore";
  }

  import ietf-tls-common {
    prefix tlscmn;
    revision-date 2022-05-24; // stable grouping definitions
    reference
      "RFC FFFF: YANG Groupings for TLS Clients and TLS Servers";
  }

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

  contact
    "WG List:  NETCONF WG list <mailto:netconf@ietf.org>
     WG Web:   https://datatracker.ietf.org/wg/netconf
     Author:   Kent Watsen <mailto:kent+ietf@watsen.net>
     Author:   Jeff Hartley <mailto:jeff.hartley@commscope.com>
     Author:   Gary Wu <mailto:garywu@cisco.com>";

  description
    "This module defines reusable groupings for TLS servers that
     can be used as a basis for specific TLS server instances.
```

```
   Copyright (c) 2022 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 Revised
   BSD License set forth in Section 4.c of the IETF Trust's
   Legal Provisions Relating to IETF Documents
   (https://trustee.ietf.org/license-info).

   This version of this YANG module is part of RFC FFFF
   (https://www.rfc-editor.org/info/rfcFFFF); see the RFC
   itself for full legal notices.

   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 (RFC 2119)
   (RFC 8174) when, and only when, they appear in all
   capitals, as shown here.";

 revision 2022-05-24 {
   description
     "Initial version";
   reference
     "RFC FFFF: YANG Groupings for TLS Clients and TLS Servers";
 }

 // Features

 feature tls-server-keepalives {
   description
     "Per socket TLS keepalive parameters are configurable for
      TLS servers on the server implementing this feature.";
 }

 feature server-ident-x509-cert {
   description
     "Indicates that the server supports identifying itself
      using X.509 certificates.";
   reference
     "RFC 5280:
        Internet X.509 Public Key Infrastructure Certificate
        and Certificate Revocation List (CRL) Profile";
 }

 feature server-ident-raw-public-key {
   description
     "Indicates that the server supports identifying itself
      using raw public keys.";
```

```
  reference
    "RFC 7250:
       Using Raw Public Keys in Transport Layer Security (TLS)
       and Datagram Transport Layer Security (DTLS)";
}

feature server-ident-tls12-psk {
  description
    "Indicates that the server supports identifying itself
     using TLS-1.2 PSKs (pre-shared or pairwise-symmetric keys).";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for Transport Layer Security
       (TLS)";
}

feature server-ident-tls13-epsk {
  description
    "Indicates that the server supports identifying itself
     using TLS-1.3 External PSKs (pre-shared keys).";
  reference
    "RFC 8446:
       The Transport Layer Security (TLS) Protocol Version 1.3";
}

feature client-auth-supported {
  description
    "Indicates that the configuration for how to authenticate
     clients can be configured herein.  TLS-level client
     authentication may not be needed when client authentication
     is expected to occur only at another protocol layer.";
}

feature client-auth-x509-cert {
  description
    "Indicates that the server supports authenticating clients
     using X.509 certificates.";
  reference
    "RFC 5280:
       Internet X.509 Public Key Infrastructure Certificate
       and Certificate Revocation List (CRL) Profile";
}

feature client-auth-raw-public-key {
  description
    "Indicates that the server supports authenticating clients
     using raw public keys.";
  reference
    "RFC 7250:
```

```
       Using Raw Public Keys in Transport Layer Security (TLS)
       and Datagram Transport Layer Security (DTLS)";
}

feature client-auth-tls12-psk {
  description
    "Indicates that the server supports authenticating clients
     using PSKs (pre-shared or pairwise-symmetric keys).";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for Transport Layer Security
       (TLS)";
}

feature client-auth-tls13-epsk {
  description
    "Indicates that the server supports authenticating clients
     using TLS-1.3 External PSKs (pre-shared keys).";
  reference
    "RFC 8446:
       The Transport Layer Security (TLS) Protocol Version 1.3";
}

// Groupings

grouping tls-server-grouping {
  description
    "A reusable grouping for configuring a TLS server without
     any consideration for how underlying TCP sessions are
     established.

     Note that this grouping uses fairly typical descendant
     node names such that a stack of 'uses' statements will
     have name conflicts.  It is intended that the consuming
     data model will resolve the issue (e.g., by wrapping
     the 'uses' statement in a container called
     'tls-server-parameters').  This model purposely does
     not do this itself so as to provide maximum flexibility
     to consuming models.";

  container server-identity {
    nacm:default-deny-write;
    description
      "A locally-defined or referenced end-entity certificate,
       including any configured intermediate certificates, the
       TLS server will present when establishing a TLS connection
       in its Certificate message, as defined in Section 7.4.2
       in RFC 5246 and Section 4.4.2 in RFC 8446.";
    reference
```

```
  "RFC 5246: The Transport Layer Security (TLS) Protocol
             Version 1.2
   RFC 8446: The Transport Layer Security (TLS) Protocol
             Version 1.3
   RFC CCCC: A YANG Data Model for a Keystore";
choice auth-type {
  mandatory true;
  description
    "A choice amongst authentication types, of which one must
     be enabled (via its associated 'feature') and selected.";
  case certificate {
    if-feature "server-ident-x509-cert";
    container certificate {
      description
        "Specifies the server identity using a certificate.";
      uses
        ks:local-or-keystore-end-entity-cert-with-key-grouping{
        refine "local-or-keystore/local/local-definition" {
          must 'public-key-format'
             + ' = "ct:subject-public-key-info-format"';
        }
        refine "local-or-keystore/keystore/keystore-reference"
             + "/asymmetric-key" {
          must 'deref(.)/../ks:public-key-format'
             + ' = "ct:subject-public-key-info-format"';
        }
      }
    }
  }
  case raw-private-key {
    if-feature "server-ident-raw-public-key";
    container raw-private-key {
      description
        "Specifies the server identity using a raw
         private key.";
      uses ks:local-or-keystore-asymmetric-key-grouping {
        refine "local-or-keystore/local/local-definition" {
          must 'public-key-format'
             + ' = "ct:subject-public-key-info-format"';
        }
        refine "local-or-keystore/keystore/keystore-reference"{
          must 'deref(.)/../ks:public-key-format'
             + ' = "ct:subject-public-key-info-format"';
        }
      }
    }
  }
  case tls12-psk {
    if-feature "server-ident-tls12-psk";
```

```
  container tls12-psk {
    description
      "Specifies the server identity using a PSK (pre-shared
       or pairwise-symmetric key).";
    uses ks:local-or-keystore-symmetric-key-grouping;
    leaf id_hint {
      type string;
      description
        "The key 'psk_identity_hint' value used in the TLS
         'ServerKeyExchange' message.";
      reference
        "RFC 4279: Pre-Shared Key Ciphersuites for
                   Transport Layer Security (TLS)";
    }
  }
}
case tls13-epsk {
  if-feature "server-ident-tls13-epsk";
  container tls13-epsk {
    description
      "An External Pre-Shared Key (EPSK) is established
      or provisioned out-of-band, i.e., not from a TLS
      connection.  An EPSK is a tuple of (Base Key,
      External Identity, Hash).  External PSKs MUST
      NOT be imported for (D)TLS 1.2 or prior versions.
      When PSKs are provisioned out of band, the PSK
      identity and the KDF hash algorithm to be used
      with the PSK MUST also be provisioned.

      The structure of this container is designed
      to satisfy the requirements of RFC 8446
      Section 4.2.11, the recommendations from
      I-D ietf-tls-external-psk-guidance Section 6,
      and the EPSK input fields detailed in
      I-D draft-ietf-tls-external-psk-importer
      Section 3.1.  The base-key is based upon
      ks:local-or-keystore-symmetric-key-grouping
      in order to provide users with flexible and
      secure storage options.";
    reference
      "RFC 8446: The Transport Layer Security (TLS)
                 Protocol Version 1.3
       I-D.ietf-tls-external-psk-importer: Importing
                 External PSKs for TLS
       I-D.ietf-tls-external-psk-guidance: Guidance
                 for External PSK Usage in TLS";
    uses ks:local-or-keystore-symmetric-key-grouping;
    leaf external-identity {
      type string;
```

```
  mandatory true;
  description
    "As per Section 4.2.11 of RFC 8446, and Section 4.1
     of I-D. ietf-tls-external-psk-guidance: A sequence
     of bytes used to identify an EPSK. A label for a
     pre-shared key established externally.";
  reference
    "RFC 8446: The Transport Layer Security (TLS)
               Protocol Version 1.3
     I-D.ietf-tls-external-psk-guidance:
               Guidance for External PSK Usage in TLS";
}
leaf hash {
  type tlscmn:epsk-supported-hash;
  mandatory true;
  description
    "As per Section 4.2.11 of RFC 8446, for externally
     established PSKs, the Hash algorithm MUST be set
     when the PSK is established or default to SHA-256
     if no such algorithm is defined.  The server MUST
     ensure that it selects a compatible PSK (if any)
     and cipher suite.  Each PSK MUST only be used
     with a single hash function.";
  reference
    "RFC 8446: The Transport Layer Security (TLS)
               Protocol Version 1.3";
}
leaf context {
  type string;
  description
    "As per Section 4.1 of I-D.
     ietf-tls-external-psk-guidance: Context
     may include information about peer roles or
     identities to mitigate Selfie-style reflection
     attacks [Selfie].  If the EPSK is a key derived
     from some other protocol or sequence of protocols,
     context MUST include a channel binding for the
     deriving protocols [RFC5056].  The details of
     this binding are protocol specific.";
  reference
    "I-D.ietf-tls-external-psk-importer:
               Importing External PSKs for TLS
     I-D.ietf-tls-external-psk-guidance:
               Guidance for External PSK Usage in TLS";
}
leaf target-protocol {
  type uint16;
  description
    "As per Section 3.1 of I-D.
```

```
            ietf-tls-external-psk-guidance: The protocol
            for which a PSK is imported for use.";
          reference
            "I-D.ietf-tls-external-psk-importer:
                      Importing External PSKs for TLS";
        }
        leaf target-kdf {
          type uint16;
          description
            "As per Section 3.1 of I-D.
             ietf-tls-external-psk-guidance: The specific Key
             Derivation Function (KDF) for which a PSK is
             imported for use.";
          reference
            "I-D.ietf-tls-external-psk-importer:
                      Importing External PSKs for TLS";
        }
      }
    }
  }
} // container server-identity

container client-authentication {
  if-feature "client-auth-supported";
  nacm:default-deny-write;
  must 'ca-certs or ee-certs or raw-public-keys or tls12-psks
    or tls13-epsks';
  presence
    "Indicates that client authentication is supported (i.e.,
     that the server will request clients send certificates).
     If not configured, the TLS server SHOULD NOT request the
     TLS clients provide authentication credentials.";
  description
    "Specifies how the TLS server can authenticate TLS clients.
     Any combination of credentials is additive and unordered.

     Note that no configuration is required for PSK (pre-shared
     or pairwise-symmetric key) based authentication as the key
     is necessarily the same as configured in the '../server-
     identity' node.";
  container ca-certs {
    if-feature "client-auth-x509-cert";
    presence
      "Indicates that CA certificates have been configured.
       This statement is present so the mandatory descendant
       nodes do not imply that this node must be configured.";
    description
      "A set of certificate authority (CA) certificates used by
       the TLS server to authenticate TLS client certificates.
```

```
     A client certificate is authenticated if it has a valid
     chain of trust to a configured CA certificate.";
  reference
    "RFC BBBB: A YANG Data Model for a Truststore";
  uses ts:local-or-truststore-certs-grouping;
}
container ee-certs {
  if-feature "client-auth-x509-cert";
  presence
    "Indicates that EE certificates have been configured.
     This statement is present so the mandatory descendant
     nodes do not imply that this node must be configured.";
  description
    "A set of client certificates (i.e., end entity
     certificates) used by the TLS server to authenticate
     certificates presented by TLS clients. A client
     certificate is authenticated if it is an exact
     match to a configured client certificate.";
  reference
    "RFC BBBB: A YANG Data Model for a Truststore";
  uses ts:local-or-truststore-certs-grouping;
}
container raw-public-keys {
  if-feature "client-auth-raw-public-key";
  presence
    "Indicates that raw public keys have been configured.
     This statement is present so the mandatory descendant
     nodes do not imply that this node must be configured.";
  description
    "A set of raw public keys used by the TLS server to
     authenticate raw public keys presented by the TLS
     client.  A raw public key is authenticated if it
     is an exact match to a configured raw public key.";
  reference
    "RFC BBBB: A YANG Data Model for a Truststore";
  uses ts:local-or-truststore-public-keys-grouping {
    refine "local-or-truststore/local/local-definition"
         + "/public-key" {
      must 'public-key-format'
         + ' = "ct:subject-public-key-info-format"';
    }
    refine "local-or-truststore/truststore"
         + "/truststore-reference" {
      must 'deref(.)/../*/ts:public-key-format'
         + ' = "ct:subject-public-key-info-format"';
    }
  }
}
leaf tls12-psks {
```

```
    if-feature "client-auth-tls12-psk";
    type empty;
    description
      "Indicates that the TLS server can authenticate TLS clients
       using configured PSKs (pre-shared or pairwise-symmetric
       keys).

       No configuration is required since the PSK value is the
       same as PSK value configured in the 'server-identity'
       node.";
  }
  leaf tls13-epsks {
    if-feature "client-auth-tls13-epsk";
    type empty;
    description
      "Indicates that the TLS 1.3 server can authenticate TLS
       clients using configured external PSKs (pre-shared keys).

       No configuration is required since the PSK value is the
       same as PSK value configured in the 'server-identity'
       node.";
  }
} // container client-authentication

container hello-params {
  nacm:default-deny-write;
  if-feature "tlscmn:hello-params";
  uses tlscmn:hello-params-grouping;
  description
    "Configurable parameters for the TLS hello message.";
} // container hello-params

container keepalives {
  nacm:default-deny-write;
  if-feature "tls-server-keepalives";
  description
    "Configures the keepalive policy for the TLS server.";
  leaf peer-allowed-to-send {
    type empty;
    description
      "Indicates that the remote TLS client is allowed to send
       HeartbeatRequest messages, as defined by RFC 6520
       to this TLS server.";
    reference
      "RFC 6520: Transport Layer Security (TLS) and Datagram
       Transport Layer Security (DTLS) Heartbeat Extension";
  }
  container test-peer-aliveness {
    presence
```

```
          "Indicates that the TLS server proactively tests the
           aliveness of the remote TLS client.";
        description
          "Configures the keep-alive policy to proactively test
           the aliveness of the TLS client.  An unresponsive
           TLS client is dropped after approximately max-wait
           * max-attempts seconds.";
        leaf max-wait {
          type uint16 {
            range "1..max";
          }
          units "seconds";
          default "30";
          description
            "Sets the amount of time in seconds after which if
             no data has been received from the TLS client, a
             TLS-level message will be sent to test the
             aliveness of the TLS client.";
        }
        leaf max-attempts {
          type uint8;
          default "3";
          description
            "Sets the maximum number of sequential keep-alive
             messages that can fail to obtain a response from
             the TLS client before assuming the TLS client is
             no longer alive.";
        }
      }
    } // container keepalives
  } // grouping tls-server-grouping

}
```

<CODE ENDS>

5. Security Considerations

5.1. The "iana-tls-cipher-suite-algs" Module

The "iana-tls-cipher-suite-algs" YANG module defines a data model that is designed to be accessed via YANG based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. Both of these protocols have mandatory-to-implement secure transport layers (e.g., SSH, TLS) with mutual authentication.

The NETCONF access control model (NACM) [RFC8341] provides the means to restrict access for particular users to a pre-configured subset of all available protocol operations and content.

This YANG module defines YANG identities, for a public IANA-maintained registry, and a single protocol-accessible read-only node for the subset of those identities supported by a server.

YANG identities are not security-sensitive, as they are statically defined in the publicly-accessible YANG module.

The protocol-accessible read-only node for the algorithms supported by a server is mildly sensitive, but not to the extent that special NACM annotations are needed to prevent read-access to regular authenticated administrators.

This module does not define any writable-nodes, RPCs, actions, or notifications, and thus the security consideration for such is not provided here.

5.2. The "ietf-tls-common" YANG Module

The "ietf-tls-common" YANG module defines "grouping" statements that are designed to be accessed via YANG based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. Both of these protocols have mandatory-to-implement secure transport layers (e.g., SSH, TLS) with mutual authentication.

The NETCONF access control model (NACM) [RFC8341] provides the means to restrict access for particular users to a pre-configured subset of all available protocol operations and content.

Since the module in this document only define groupings, these considerations are primarily for the designers of other modules that use these groupings.

None of the readable data nodes defined in this YANG module are considered sensitive or vulnerable in network environments. The NACM

"default-deny-all" extension has not been set for any data nodes defined in this module.

None of the writable data nodes defined in this YANG module are considered sensitive or vulnerable in network environments. The NACM "default-deny-write" extension has not been set for any data nodes defined in this module.

This module does not define any RPCs, actions, or notifications, and thus the security consideration for such is not provided here.

5.3. The "ietf-tls-client" YANG Module

The "ietf-tls-client" YANG module defines "grouping" statements that are designed to be accessed via YANG based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. Both of these protocols have mandatory-to-implement secure transport layers (e.g., SSH, TLS) with mutual authentication.

The NETCONF access control model (NACM) [RFC8341] provides the means to restrict access for particular users to a pre-configured subset of all available protocol operations and content.

Since the module in this document only define groupings, these considerations are primarily for the designers of other modules that use these groupings.

None of the readable data nodes defined in this YANG module are considered sensitive or vulnerable in network environments. The NACM "default-deny-all" extension has not been set for any data nodes defined in this module.

Please be aware that this module uses the "key" and "private-key" nodes from the "ietf-crypto-types" module [I-D.ietf-netconf-crypto-types], where said nodes have the NACM extension "default-deny-all" set, thus preventing unrestricted read-access to the cleartext key values.

All the writable data nodes defined by this module may be considered sensitive or vulnerable in some network environments. For instance, any modification to a key or reference to a key may dramatically alter the implemented security policy. For this reason, the NACM extension "default-deny-write" has been set for all data nodes defined in this module.

This module does not define any RPCs, actions, or notifications, and thus the security consideration for such is not provided here.

5.4. The "ietf-tls-server" YANG Module

The "ietf-tls-server" YANG module defines "grouping" statements that are designed to be accessed via YANG based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. Both of these protocols have mandatory-to-implement secure transport layers (e.g., SSH, TLS) with mutual authentication.

The NETCONF access control model (NACM) [RFC8341] provides the means to restrict access for particular users to a pre-configured subset of all available protocol operations and content.

Since the module in this document only define groupings, these considerations are primarily for the designers of other modules that use these groupings.

None of the readable data nodes defined in this YANG module are considered sensitive or vulnerable in network environments. The NACM "default-deny-all" extension has not been set for any data nodes defined in this module.

Please be aware that this module uses the "key" and "private-key" nodes from the "ietf-crypto-types" module [I-D.ietf-netconf-crypto-types], where said nodes have the NACM extension "default-deny-all" set, thus preventing unrestricted read-access to the cleartext key values.

All the writable data nodes defined by this module may be considered sensitive or vulnerable in some network environments. For instance, any modification to a key or reference to a key may dramatically alter the implemented security policy. For this reason, the NACM extension "default-deny-write" has been set for all data nodes defined in this module.

This module does not define any RPCs, actions, or notifications, and thus the security consideration for such is not provided here.

6. IANA Considerations

6.1. The "IETF XML" Registry

This document registers four URIs in the "ns" subregistry of the IETF XML Registry [RFC3688]. Following the format in [RFC3688], the following registrations are requested:

```
URI: urn:ietf:params:xml:ns:yang:iana-tls-cipher-suite-algs
Registrant Contact: IANA
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-tls-common
Registrant Contact: The IESG
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-tls-client
Registrant Contact: The IESG
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-tls-server
Registrant Contact: The IESG
XML: N/A, the requested URI is an XML namespace.
```

6.2. The "YANG Module Names" Registry

This document registers four YANG modules in the YANG Module Names registry [RFC6020]. Following the format in [RFC6020], the following registrations are requested:

```
name:         iana-tls-cipher-suite-algs
namespace:    urn:ietf:params:xml:ns:yang:iana-tls-cipher-suite-algs
prefix:       tlscsa
reference:    RFC FFFF

name:         ietf-tls-common
namespace:    urn:ietf:params:xml:ns:yang:ietf-tls-common
prefix:       tlscmn
reference:    RFC FFFF

name:         ietf-tls-client
namespace:    urn:ietf:params:xml:ns:yang:ietf-tls-client
prefix:       tlsc
reference:    RFC FFFF

name:         ietf-tls-server
namespace:    urn:ietf:params:xml:ns:yang:ietf-tls-server
prefix:       tlss
reference:    RFC FFFF
```

6.3. The "iana-tls-cipher-suite-algs" Module

IANA is requested to maintain a YANG module called "iana-tls-cipher-suite-algs" that shadows the "TLS Cipher Suites" sub-registry of the "Transport Layer Security (TLS) Parameters" registry [IANA-CIPHER-ALGS].

This registry defines a YANG identity for each cipher suite algorithm, and a "base" identity from which all of the other identities are derived.

An initial version of this module can be found in Appendix A.1.

*Please note that this module was created on June 2st, 2021, and that additional entries may have been added in the interim before this document's publication. If this is that case, IANA may either publish just an updated module containing the new entries, or publish the initial module as is immediately followed by a "revision" containing the additional algorithm names.

*Please also note that the "status" statement has been set to "deprecated", if the "RECOMMENDED" column in the registry had the value 'N', and to "obsolete", if the "References" column included Moving single-DES and IDEA TLS ciphersuites to Historic reference.

7. References

7.1. Normative References

[I-D.ietf-netconf-crypto-types] Watsen, K., "YANG Data Types and Groupings for Cryptography", Work in Progress, Internet-Draft, draft-ietf-netconf-crypto-types-22, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-crypto-types-22>.

[I-D.ietf-netconf-keystore] Watsen, K., "A YANG Data Model for a Keystore", Work in Progress, Internet-Draft, draft-ietf-netconf-keystore-24, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-keystore-24>.

[I-D.ietf-netconf-trust-anchors] Watsen, K., "A YANG Data Model for a Truststore", Work in Progress, Internet-Draft, draft-ietf-netconf-trust-anchors-17, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-trust-anchors-17>.

[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>.

[RFC5288] Salowey, J., Choudhury, A., and D. McGrew, "AES Galois Counter Mode (GCM) Cipher Suites for TLS", RFC 5288, DOI

10.17487/RFC5288, August 2008, <https://www.rfc-editor.org/info/rfc5288>.

[RFC5289] Rescorla, E., "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)", RFC 5289, DOI 10.17487/RFC5289, August 2008, <https://www.rfc-editor.org/info/rfc5289>.

[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>.

[RFC7589] Badra, M., Luchuk, A., and J. Schoenwaelder, "Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication", RFC 7589, DOI 10.17487/RFC7589, June 2015, <https://www.rfc-editor.org/info/rfc7589>.

[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>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8422] Nir, Y., Josefsson, S., and M. Pegourie-Gonnard, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier", RFC 8422, DOI 10.17487/RFC8422, August 2018, <https://www.rfc-editor.org/info/rfc8422>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

7.2. Informative References

[I-D.ietf-netconf-http-client-server]
Watsen, K., "YANG Groupings for HTTP Clients and HTTP Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-http-client-server-09, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-http-client-server-09>.

[I-D.ietf-netconf-netconf-client-server]
Watsen, K., "NETCONF Client and Server Models", Work in Progress, Internet-Draft, draft-ietf-netconf-netconf-client-server-25, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-netconf-client-server-25>.

[I-D.ietf-netconf-restconf-client-server]
Watsen, K., "RESTCONF Client and Server Models", Work in Progress, Internet-Draft, draft-ietf-netconf-restconf-client-server-25, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-restconf-client-server-25>.

[I-D.ietf-netconf-ssh-client-server]
Watsen, K., "YANG Groupings for SSH Clients and SSH Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-ssh-client-server-27, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-ssh-client-server-27>.

[I-D.ietf-netconf-tcp-client-server] Watsen, K. and M. Scharf, "YANG Groupings for TCP Clients and TCP Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-tcp-client-server-12, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-tcp-client-server-12>.

[I-D.ietf-netconf-tls-client-server]
Watsen, K., "YANG Groupings for TLS Clients and TLS Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-tls-client-server-27, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-tls-client-server-27>.

[I-D.ietf-tls-external-psk-guidance] Housley, R., Hoyland, J., Sethi, M., and C. A. Wood, "Guidance for External PSK Usage in TLS", Work in Progress, Internet-Draft, draft-ietf-tls-external-psk-guidance-06, 4 February 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-external-psk-guidance-06>.

[I-D.ietf-tls-external-psk-importer] Benjamin, D. and C. A. Wood, "Importing External PSKs for TLS", Work in Progress, Internet-Draft, draft-ietf-tls-external-psk-importer-08, 22 April 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-external-psk-importer-08>.

[IANA-CIPHER-ALGS] (IANA), I. A. N. A., "IANA "TLS Cipher Suites" Sub-registry of the "Transport Layer Security (TLS)

Parameters" Registry", <https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-4>.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, DOI 10.17487/RFC2246, January 1999, <https://www.rfc-editor.org/info/rfc2246>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, DOI 10.17487/RFC4346, April 2006, <https://www.rfc-editor.org/info/rfc4346>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8071] Watsen, K., "NETCONF Call Home and RESTCONF Call Home", RFC 8071, DOI 10.17487/RFC8071, February 2017, <https://www.rfc-editor.org/info/rfc8071>.

[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>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

Appendix A. YANG Modules for IANA

The module contained in this section was generated by scripts using the contents of the associated sub-registry as they existed on June 2nd, 2021.

A.1. Initial Module for the "TLS Cipher Suites" Registry

A.1.1. Data Model Overview

This section provides an overview of the "iana-tls-cipher-suite-algs" module in terms of its identities and protocol-accessible nodes.

A.1.1.1. Identities

The following diagram lists the base "identity" statements defined in the module, of which there is just one, and illustrates that all the derived identity statements are generated from the associated IANA-maintained registry [IANA-CIPHER-ALGS].

```
Identities:
  +-- cipher-suite-alg-base
    +-- <identity-name from IANA registry>
```

The diagram above uses syntax that is similar to but not defined in [RFC8340].

A.1.1.2. Typedefs

The following diagram illustrates the "typedef" statements defined in the "iana-tls-cipher-suite-algs" module:

```
Typedefs:
  identityref
    +-- cipher-suite-algorithm-ref
```

The diagram above uses syntax that is similar to but not defined in [RFC8340].

Comments:

*The typedef defined in the "iana-tls-cipher-suite-algs" module extends the "identityref" type defined in [RFC7950].

A.1.1.3. Protocol-accessible Nodes

The following tree diagram [RFC8340] lists all the protocol-accessible nodes defined in the "iana-tls-cipher-suite-alg" module:

```
module: iana-tls-cipher-suite-algs
  +--ro supported-algorithms
     +--ro supported-algorithm*   cipher-suite-algorithm-ref
```

Comments:

*Protocol-accessible nodes are those nodes that are accessible when the module is "implemented", as described in Section 5.6.5 of [RFC7950].

A.1.2. Example Usage

The following example illustrates operational state data indicating the TLS cipher suite algorithms supported by the server:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

<supported-algorithms
  xmlns="urn:ietf:params:xml:ns:yang:iana-tls-cipher-suite-algs"
  xmlns:tlscsa="urn:ietf:params:xml:ns:yang:iana-tls-cipher-suite-al\
gs">
  <supported-algorithm>tlscsa:tls-ecdhe-ecdsa-with-aes-256-cbc-sha</\
supported-algorithm>
  <supported-algorithm>tlscsa:tls-dhe-rsa-with-aes-128-cbc-sha256</s\
upported-algorithm>
  <supported-algorithm>tlscsa:tls-rsa-with-3des-ede-cbc-sha</support\
ed-algorithm>
  <supported-algorithm>tlscsa:tls-ecdhe-psk-with-aes-256-gcm-sha384<\
/supported-algorithm>
  <supported-algorithm>tlscsa:tls-dhe-psk-with-chacha20-poly1305-sha\
256</supported-algorithm>
  <supported-algorithm>tlscsa:tls-eccpwd-with-aes-256-gcm-sha384</su\
pported-algorithm>
  <supported-algorithm>tlscsa:tls-psk-with-aes-256-ccm</supported-al\
gorithm>
  <supported-algorithm>tlscsa:tls-dhe-psk-with-camellia-256-cbc-sha3\
84</supported-algorithm>
  <supported-algorithm>tlscsa:tls-ecdh-rsa-with-aes-256-cbc-sha384</\
supported-algorithm>
  <supported-algorithm>tlscsa:tls-ecdh-rsa-with-3des-ede-cbc-sha</su\
pported-algorithm>
  <supported-algorithm>tlscsa:tls-dh-dss-with-aes-128-gcm-sha256</su\
pported-algorithm>
</supported-algorithms>
```

A.1.3. YANG Module

Following are the complete contents to the initial IANA-maintained YANG module. Please note that the date "2021-06-02" reflects the day on which the extraction occurred.

```
<CODE BEGINS> file "iana-tls-cipher-suite-algs@2021-06-02.yang"
```

```
module iana-tls-cipher-suite-algs {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:iana-tls-cipher-suite-algs";
  prefix tlscsa;

  organization
    "Internet Assigned Numbers Authority (IANA)";

  contact
    "Postal: ICANN
             12025 Waterfront Drive, Suite 300
             Los Angeles, CA  90094-2536
             United States of America
     Tel:    +1 310 301 5800
     Email:  iana@iana.org";

  description
    "This module defines identities for the Cipher Suite
     algorithms defined in the 'TLS Cipher Suites' sub-registry
     of the 'Transport Layer Security (TLS) Parameters' registry
     maintained by IANA.

     Copyright (c) 2021 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 Revised
     BSD License set forth in Section 4.c of the IETF Trust's
     Legal Provisions Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     The initial version of this YANG module is part of RFC FFFF
     (https://www.rfc-editor.org/info/rfcFFFF); see the RFC
     itself for full legal notices.";

  revision 2021-06-02 {
    description
      "Initial version";
    reference
      "RFC FFFF: YANG Groupings for TLS Clients and TLS Servers";
  }

  // Typedefs

  typedef cipher-suite-algorithm-ref {
    type identityref {
      base "cipher-suite-alg-base";
    }
    description
```

```
    "A reference to a TLS cipher suite algorithm identifier.";
}


// Identities

identity cipher-suite-alg-base {
  description
    "Base identity used to identify TLS cipher suites.";
}

identity tls-null-with-null-null {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-NULL-WITH-NULL-NULL";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-rsa-with-null-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-NULL-MD5";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-rsa-with-null-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-NULL-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-rsa-export-with-rc4-40-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-EXPORT-WITH-RC4-40-MD5";
  reference
    "RFC 4346:
       The TLS Protocol Version 1.1
```

```
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-rsa-with-rc4-128-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-RC4-128-MD5";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-rsa-with-rc4-128-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-RC4-128-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-rsa-export-with-rc2-cbc-40-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-EXPORT-WITH-RC2-CBC-40-MD5";
  reference
    "RFC 4346:
       The TLS Protocol Version 1.1";
}

identity tls-rsa-with-idea-cbc-sha {
  base cipher-suite-alg-base;
  status obsolete;
  description
    "TLS-RSA-WITH-IDEA-CBC-SHA";
  reference
    "RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)
     RFC 5469:
       DES and IDEA Cipher Suites for
```

```
       Transport Layer Security (TLS)";
}

identity tls-rsa-export-with-des40-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-EXPORT-WITH-DES40-CBC-SHA";
  reference
    "RFC 4346:
       The TLS Protocol Version 1.1";
}

identity tls-rsa-with-des-cbc-sha {
  base cipher-suite-alg-base;
  status obsolete;
  description
    "TLS-RSA-WITH-DES-CBC-SHA";
  reference
    "RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)
     RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-rsa-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-dss-export-with-des40-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-EXPORT-WITH-DES40-CBC-SHA";
  reference
    "RFC 4346:
       The TLS Protocol Version 1.1";
}

identity tls-dh-dss-with-des-cbc-sha {
  base cipher-suite-alg-base;
```

```
  status obsolete;
  description
    "TLS-DH-DSS-WITH-DES-CBC-SHA";
  reference
    "RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)
     RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-dh-dss-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-rsa-export-with-des40-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-EXPORT-WITH-DES40-CBC-SHA";
  reference
    "RFC 4346:
       The TLS Protocol Version 1.1";
}

identity tls-dh-rsa-with-des-cbc-sha {
  base cipher-suite-alg-base;
  status obsolete;
  description
    "TLS-DH-RSA-WITH-DES-CBC-SHA";
  reference
    "RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)
     RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-dh-rsa-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
```

```
  description
    "TLS-DH-RSA-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dhe-dss-export-with-des40-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-EXPORT-WITH-DES40-CBC-SHA";
  reference
    "RFC 4346:
       The TLS Protocol Version 1.1";
}

identity tls-dhe-dss-with-des-cbc-sha {
  base cipher-suite-alg-base;
  status obsolete;
  description
    "TLS-DHE-DSS-WITH-DES-CBC-SHA";
  reference
    "RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)
     RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-dhe-dss-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dhe-rsa-export-with-des40-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-EXPORT-WITH-DES40-CBC-SHA";
  reference
    "RFC 4346:
       The TLS Protocol Version 1.1";
```

```
}

identity tls-dhe-rsa-with-des-cbc-sha {
  base cipher-suite-alg-base;
  status obsolete;
  description
    "TLS-DHE-RSA-WITH-DES-CBC-SHA";
  reference
    "RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)
     RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-dhe-rsa-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-anon-export-with-rc4-40-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-EXPORT-WITH-RC4-40-MD5";
  reference
    "RFC 4346:
       The TLS Protocol Version 1.1
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-dh-anon-with-rc4-128-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-RC4-128-MD5";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}
```

```
identity tls-dh-anon-export-with-des40-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-EXPORT-WITH-DES40-CBC-SHA";
  reference
    "RFC 4346:
       The TLS Protocol Version 1.1";
}

identity tls-dh-anon-with-des-cbc-sha {
  base cipher-suite-alg-base;
  status obsolete;
  description
    "TLS-DH-ANON-WITH-DES-CBC-SHA";
  reference
    "RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)
     RFC 5469:
       DES and IDEA Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-dh-anon-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-krb5-with-des-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-WITH-DES-CBC-SHA";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-krb5-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
```

```
  description
    "TLS-KRB5-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-krb5-with-rc4-128-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-WITH-RC4-128-SHA";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-krb5-with-idea-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-WITH-IDEA-CBC-SHA";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-krb5-with-des-cbc-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-WITH-DES-CBC-MD5";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-krb5-with-3des-ede-cbc-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-WITH-3DES-EDE-CBC-MD5";
  reference
```

```
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-krb5-with-rc4-128-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-WITH-RC4-128-MD5";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-krb5-with-idea-cbc-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-WITH-IDEA-CBC-MD5";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-krb5-export-with-des-cbc-40-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-EXPORT-WITH-DES-CBC-40-SHA";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-krb5-export-with-rc2-cbc-40-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-EXPORT-WITH-RC2-CBC-40-SHA";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)";
```

```
}

identity tls-krb5-export-with-rc4-40-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-EXPORT-WITH-RC4-40-SHA";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-krb5-export-with-des-cbc-40-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-EXPORT-WITH-DES-CBC-40-MD5";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-krb5-export-with-rc2-cbc-40-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-EXPORT-WITH-RC2-CBC-40-MD5";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-krb5-export-with-rc4-40-md5 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-KRB5-EXPORT-WITH-RC4-40-MD5";
  reference
    "RFC 2712:
       Addition of Kerberos Cipher Suites to
       Transport Layer Security (TLS)
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}
```

```
identity tls-psk-with-null-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-NULL-SHA";
  reference
    "RFC 4785:
       Pre-Shared Key Cipher Suites with NULL Encryption for
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-null-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-NULL-SHA";
  reference
    "RFC 4785:
       Pre-Shared Key Cipher Suites with NULL Encryption for
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-null-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-NULL-SHA";
  reference
    "RFC 4785:
       Pre-Shared Key Cipher Suites with NULL Encryption for
       Transport Layer Security (TLS)";
}

identity tls-rsa-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-AES-128-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-dss-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-AES-128-CBC-SHA";
```

```
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-rsa-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-AES-128-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dhe-dss-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-AES-128-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dhe-rsa-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-AES-128-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-anon-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-AES-128-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-rsa-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
```

```
    "TLS-RSA-WITH-AES-256-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-dss-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-AES-256-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-rsa-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-AES-256-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dhe-dss-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-AES-256-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dhe-rsa-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-AES-256-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-anon-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
```

```
  description
    "TLS-DH-ANON-WITH-AES-256-CBC-SHA";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-rsa-with-null-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-NULL-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-rsa-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-rsa-with-aes-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-AES-256-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-dss-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-rsa-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
```

```
  status deprecated;
  description
    "TLS-DH-RSA-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dhe-dss-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-rsa-with-camellia-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-CAMELLIA-128-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dh-dss-with-camellia-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-CAMELLIA-128-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dh-rsa-with-camellia-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-CAMELLIA-128-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dhe-dss-with-camellia-128-cbc-sha {
```

```
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-CAMELLIA-128-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dhe-rsa-with-camellia-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-CAMELLIA-128-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dh-anon-with-camellia-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-CAMELLIA-128-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dhe-rsa-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-dss-with-aes-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-AES-256-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}
```

```
identity tls-dh-rsa-with-aes-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-AES-256-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dhe-dss-with-aes-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-AES-256-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dhe-rsa-with-aes-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-AES-256-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-anon-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}

identity tls-dh-anon-with-aes-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-AES-256-CBC-SHA256";
  reference
    "RFC 5246:
       The Transport Layer Security (TLS) Protocol Version 1.2";
}
```

```
identity tls-rsa-with-camellia-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-CAMELLIA-256-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dh-dss-with-camellia-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-CAMELLIA-256-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dh-rsa-with-camellia-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-CAMELLIA-256-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dhe-dss-with-camellia-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-CAMELLIA-256-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dhe-rsa-with-camellia-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-CAMELLIA-256-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
```

```
}

identity tls-dh-anon-with-camellia-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-CAMELLIA-256-CBC-SHA";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-psk-with-rc4-128-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-RC4-128-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-psk-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-psk-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-AES-128-CBC-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-psk-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
```

```
  status deprecated;
  description
    "TLS-PSK-WITH-AES-256-CBC-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-rc4-128-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-RC4-128-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-dhe-psk-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-AES-128-CBC-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-AES-256-CBC-SHA";
```

```
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-rc4-128-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-RC4-128-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-rsa-psk-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-AES-128-CBC-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-AES-256-CBC-SHA";
  reference
    "RFC 4279:
       Pre-Shared Key Ciphersuites for
```

```
       Transport Layer Security (TLS)";
}

identity tls-rsa-with-seed-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-SEED-CBC-SHA";
  reference
    "RFC 4162:
       Addition of SEED Ciphersuites to
       Transport Layer Security (TLS)";
}

identity tls-dh-dss-with-seed-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-SEED-CBC-SHA";
  reference
    "RFC 4162:
       Addition of SEED Ciphersuites to
       Transport Layer Security (TLS)";
}

identity tls-dh-rsa-with-seed-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-SEED-CBC-SHA";
  reference
    "RFC 4162:
       Addition of SEED Ciphersuites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-dss-with-seed-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-SEED-CBC-SHA";
  reference
    "RFC 4162:
       Addition of SEED Ciphersuites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-rsa-with-seed-cbc-sha {
  base cipher-suite-alg-base;
```

```
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-SEED-CBC-SHA";
  reference
    "RFC 4162:
       Addition of SEED Ciphersuites to
       Transport Layer Security (TLS)";
}

identity tls-dh-anon-with-seed-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-SEED-CBC-SHA";
  reference
    "RFC 4162:
       Addition of SEED Ciphersuites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}

identity tls-rsa-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}

identity tls-dhe-rsa-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-DHE-RSA-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}
```

```
identity tls-dhe-rsa-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  description
    "TLS-DHE-RSA-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}

identity tls-dh-rsa-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}

identity tls-dh-rsa-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}

identity tls-dhe-dss-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}

identity tls-dhe-dss-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}
```

```
identity tls-dh-dss-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}

identity tls-dh-dss-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}

identity tls-dh-anon-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}

identity tls-dh-anon-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5288:
       AES-GCM Cipher Suites for TLS";
}

identity tls-psk-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
```

```
}

identity tls-psk-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-dhe-psk-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-DHE-PSK-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-dhe-psk-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  description
    "TLS-DHE-PSK-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-rsa-psk-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-rsa-psk-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-AES-256-GCM-SHA384";
```

```
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-psk-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-psk-with-aes-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-AES-256-CBC-SHA384";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-psk-with-null-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-NULL-SHA256";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-psk-with-null-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-NULL-SHA384";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}
```

```
identity tls-dhe-psk-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-dhe-psk-with-aes-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-AES-256-CBC-SHA384";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-dhe-psk-with-null-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-NULL-SHA256";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-dhe-psk-with-null-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-NULL-SHA384";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-rsa-psk-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
```

```
    "TLS-RSA-PSK-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-rsa-psk-with-aes-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-AES-256-CBC-SHA384";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-rsa-psk-with-null-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-NULL-SHA256";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-rsa-psk-with-null-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-NULL-SHA384";
  reference
    "RFC 5487:
       Pre-Shared Key Cipher Suites for Transport Layer Security
       (TLS) with SHA-256/384 and AES Galois Counter Mode";
}

identity tls-rsa-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}
```

```
identity tls-dh-dss-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dh-rsa-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dhe-dss-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dhe-rsa-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dh-anon-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
```

```
}

identity tls-rsa-with-camellia-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-CAMELLIA-256-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dh-dss-with-camellia-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-CAMELLIA-256-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dh-rsa-with-camellia-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-CAMELLIA-256-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dhe-dss-with-camellia-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-CAMELLIA-256-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-dhe-rsa-with-camellia-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-CAMELLIA-256-CBC-SHA256";
  reference
    "RFC 5932:
```

```
       Camellia Cipher Suites for TLS";
}

identity tls-dh-anon-with-camellia-256-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-CAMELLIA-256-CBC-SHA256";
  reference
    "RFC 5932:
       Camellia Cipher Suites for TLS";
}

identity tls-sm4-gcm-sm3 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-SM4-GCM-SM3";
  reference
    "RFC 8998:
       ShangMi (SM) Cipher Suites for Transport Layer Security
       (TLS) Protocol Version 1.3";
}

identity tls-sm4-ccm-sm3 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-SM4-CCM-SM3";
  reference
    "RFC 8998:
       ShangMi (SM) Cipher Suites for Transport Layer Security
       (TLS) Protocol Version 1.3";
}

identity tls-empty-renegotiation-info-scsv {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-EMPTY-RENEGOTIATION-INFO-SCSV";
  reference
    "RFC 5746:
       Transport Layer Security (TLS)
       Renegotiation Indication Extension";
}

identity tls-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  description
```

```
    "TLS-AES-128-GCM-SHA256";
  reference
    "RFC 8446:
       The Transport Layer Security (TLS) Protocol Version 1.3";
}

identity tls-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  description
    "TLS-AES-256-GCM-SHA384";
  reference
    "RFC 8446:
       The Transport Layer Security (TLS) Protocol Version 1.3";
}

identity tls-chacha20-poly1305-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-CHACHA20-POLY1305-SHA256";
  reference
    "RFC 8446:
       The Transport Layer Security (TLS) Protocol Version 1.3";
}

identity tls-aes-128-ccm-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-AES-128-CCM-SHA256";
  reference
    "RFC 8446:
       The Transport Layer Security (TLS) Protocol Version 1.3";
}

identity tls-aes-128-ccm-8-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-AES-128-CCM-8-SHA256";
  reference
    "RFC 8446:
       The Transport Layer Security (TLS) Protocol Version 1.3";
}

identity tls-fallback-scsv {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-FALLBACK-SCSV";
  reference
```

```
    "RFC 7507:
       TLS Fallback Signaling Cipher Suite Value (SCSV)
       for Preventing Protocol Downgrade Attacks";
}

identity tls-ecdh-ecdsa-with-null-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-NULL-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdh-ecdsa-with-rc4-128-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-RC4-128-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-ecdh-ecdsa-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdh-ecdsa-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-AES-128-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
```

```
}

identity tls-ecdh-ecdsa-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-AES-256-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdhe-ecdsa-with-null-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-NULL-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdhe-ecdsa-with-rc4-128-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-RC4-128-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-ecdhe-ecdsa-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdhe-ecdsa-with-aes-128-cbc-sha {
```

```
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-AES-128-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdhe-ecdsa-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-AES-256-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdh-rsa-with-null-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-NULL-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdh-rsa-with-rc4-128-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-RC4-128-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-ecdh-rsa-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
```

```
    "TLS-ECDH-RSA-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdh-rsa-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-AES-128-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdh-rsa-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-AES-256-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdhe-rsa-with-null-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-NULL-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdhe-rsa-with-rc4-128-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-RC4-128-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier
```

```
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-ecdhe-rsa-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdhe-rsa-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-AES-128-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdhe-rsa-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-AES-256-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdh-anon-with-null-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ANON-WITH-NULL-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdh-anon-with-rc4-128-sha {
```

```
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ANON-WITH-RC4-128-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-ecdh-anon-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ANON-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdh-anon-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ANON-WITH-AES-128-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-ecdh-anon-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ANON-WITH-AES-256-CBC-SHA";
  reference
    "RFC 8422:
       Elliptic Curve Cryptography (ECC) Cipher Suites for
       Transport Layer Security (TLS) Versions 1.2 and Earlier";
}

identity tls-srp-sha-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
```

```
    "TLS-SRP-SHA-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 5054:
       Using SRP for TLS Authentication";
}

identity tls-srp-sha-rsa-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-SRP-SHA-RSA-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 5054:
       Using SRP for TLS Authentication";
}

identity tls-srp-sha-dss-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-SRP-SHA-DSS-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 5054:
       Using SRP for TLS Authentication";
}

identity tls-srp-sha-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-SRP-SHA-WITH-AES-128-CBC-SHA";
  reference
    "RFC 5054:
       Using SRP for TLS Authentication";
}

identity tls-srp-sha-rsa-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-SRP-SHA-RSA-WITH-AES-128-CBC-SHA";
  reference
    "RFC 5054:
       Using SRP for TLS Authentication";
}

identity tls-srp-sha-dss-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
```

```
  description
    "TLS-SRP-SHA-DSS-WITH-AES-128-CBC-SHA";
  reference
    "RFC 5054:
       Using SRP for TLS Authentication";
}

identity tls-srp-sha-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-SRP-SHA-WITH-AES-256-CBC-SHA";
  reference
    "RFC 5054:
       Using SRP for TLS Authentication";
}

identity tls-srp-sha-rsa-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-SRP-SHA-RSA-WITH-AES-256-CBC-SHA";
  reference
    "RFC 5054:
       Using SRP for TLS Authentication";
}

identity tls-srp-sha-dss-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-SRP-SHA-DSS-WITH-AES-256-CBC-SHA";
  reference
    "RFC 5054:
       Using SRP for TLS Authentication";
}

identity tls-ecdhe-ecdsa-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdhe-ecdsa-with-aes-256-cbc-sha384 {
```

```
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-AES-256-CBC-SHA384";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdh-ecdsa-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdh-ecdsa-with-aes-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-AES-256-CBC-SHA384";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdhe-rsa-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdhe-rsa-with-aes-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-AES-256-CBC-SHA384";
  reference
```

```
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdh-rsa-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdh-rsa-with-aes-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-AES-256-CBC-SHA384";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdhe-ecdsa-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-ECDHE-ECDSA-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdhe-ecdsa-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  description
    "TLS-ECDHE-ECDSA-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdh-ecdsa-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
```

```
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdh-ecdsa-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdhe-rsa-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdhe-rsa-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  description
    "TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdh-rsa-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
```

```
}

identity tls-ecdh-rsa-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 5289:
       TLS Elliptic Curve Cipher Suites with SHA-256/384
       and AES Galois Counter Mode";
}

identity tls-ecdhe-psk-with-rc4-128-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-RC4-128-SHA";
  reference
    "RFC 5489:
       ECDHE_PSK Ciphersuites for Transport Layer Security (TLS)
     RFC 6347:
       Datagram Transport Layer Security version 1.2";
}

identity tls-ecdhe-psk-with-3des-ede-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-3DES-EDE-CBC-SHA";
  reference
    "RFC 5489:
       ECDHE_PSK Ciphersuites for Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-aes-128-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-AES-128-CBC-SHA";
  reference
    "RFC 5489:
       ECDHE_PSK Ciphersuites for Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-aes-256-cbc-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
```

```
    "TLS-ECDHE-PSK-WITH-AES-256-CBC-SHA";
  reference
    "RFC 5489:
       ECDHE_PSK Ciphersuites for Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-aes-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-AES-128-CBC-SHA256";
  reference
    "RFC 5489:
       ECDHE_PSK Ciphersuites for Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-aes-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-AES-256-CBC-SHA384";
  reference
    "RFC 5489:
       ECDHE_PSK Ciphersuites for Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-null-sha {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-NULL-SHA";
  reference
    "RFC 5489:
       ECDHE_PSK Ciphersuites for Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-null-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-NULL-SHA256";
  reference
    "RFC 5489:
       ECDHE_PSK Ciphersuites for Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-null-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
```

```
  description
    "TLS-ECDHE-PSK-WITH-NULL-SHA384";
  reference
    "RFC 5489:
       ECDHE_PSK Ciphersuites for Transport Layer Security (TLS)";
}

identity tls-rsa-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-dss-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-dss-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
```

```
}

identity tls-dh-rsa-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-rsa-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-dss-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-dss-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-rsa-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
```

```
  description
    "TLS-DHE-RSA-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-rsa-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-anon-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-anon-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-ecdsa-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
```

```
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-ecdsa-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-ecdsa-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-ecdsa-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-rsa-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-rsa-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
```

```
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-rsa-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-rsa-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
```

```
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-rsa-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-rsa-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-rsa-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-rsa-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-dss-with-aria-128-gcm-sha256 {
```

```
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-dss-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-dss-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-dss-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-anon-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-ARIA-128-GCM-SHA256";
  reference
```

```
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-anon-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-ecdsa-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-ecdsa-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-ecdsa-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}
```

```
identity tls-ecdh-ecdsa-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-rsa-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-rsa-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-rsa-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-rsa-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-ARIA-256-GCM-SHA384";
```

```
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-psk-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-psk-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}
```

```
identity tls-rsa-psk-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-psk-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-psk-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
```

```
    "TLS-DHE-PSK-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-aria-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-ARIA-128-GCM-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-aria-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-ARIA-256-GCM-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-aria-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-ARIA-128-CBC-SHA256";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
```

```
}

identity tls-ecdhe-psk-with-aria-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-ARIA-256-CBC-SHA384";
  reference
    "RFC 6209:
       Addition of the ARIA Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-ecdsa-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-ecdsa-with-camellia-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-CAMELLIA-256-CBC-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-ecdsa-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-ecdsa-with-camellia-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
```

```
  description
    "TLS-ECDH-ECDSA-WITH-CAMELLIA-256-CBC-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-rsa-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-rsa-with-camellia-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-CAMELLIA-256-CBC-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-rsa-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-rsa-with-camellia-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-CAMELLIA-256-CBC-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
```

```
       Transport Layer Security (TLS)";
}

identity tls-rsa-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-rsa-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-rsa-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-rsa-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
```

```
  status deprecated;
  description
    "TLS-DH-RSA-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-rsa-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-RSA-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-dss-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-dss-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-DSS-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-dss-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
```

```
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-dss-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-DSS-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-anon-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dh-anon-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DH-ANON-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-ecdsa-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-ecdsa-with-camellia-256-gcm-sha384 {
```

```
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-ecdsa-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-ecdsa-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-ECDSA-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-rsa-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-rsa-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-RSA-WITH-CAMELLIA-256-GCM-SHA384";
  reference
```

```
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-rsa-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdh-rsa-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDH-RSA-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-psk-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-psk-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}
```

```
identity tls-dhe-psk-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-camellia-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-CAMELLIA-128-GCM-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-camellia-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-CAMELLIA-256-GCM-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-psk-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-CAMELLIA-128-CBC-SHA256";
```

```
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-psk-with-camellia-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-CAMELLIA-256-CBC-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-camellia-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-PSK-WITH-CAMELLIA-256-CBC-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}
```

```
identity tls-rsa-psk-with-camellia-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-CAMELLIA-256-CBC-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-camellia-128-cbc-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-CAMELLIA-128-CBC-SHA256";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-camellia-256-cbc-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-CAMELLIA-256-CBC-SHA384";
  reference
    "RFC 6367:
       Addition of the Camellia Cipher Suites to
       Transport Layer Security (TLS)";
}

identity tls-rsa-with-aes-128-ccm {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-AES-128-CCM";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-rsa-with-aes-256-ccm {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-AES-256-CCM";
```

```
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-dhe-rsa-with-aes-128-ccm {
  base cipher-suite-alg-base;
  description
    "TLS-DHE-RSA-WITH-AES-128-CCM";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-dhe-rsa-with-aes-256-ccm {
  base cipher-suite-alg-base;
  description
    "TLS-DHE-RSA-WITH-AES-256-CCM";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-rsa-with-aes-128-ccm-8 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-AES-128-CCM-8";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-rsa-with-aes-256-ccm-8 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-WITH-AES-256-CCM-8";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-dhe-rsa-with-aes-128-ccm-8 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-AES-128-CCM-8";
  reference
```

```
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-dhe-rsa-with-aes-256-ccm-8 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-DHE-RSA-WITH-AES-256-CCM-8";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-psk-with-aes-128-ccm {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-AES-128-CCM";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-psk-with-aes-256-ccm {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-AES-256-CCM";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-dhe-psk-with-aes-128-ccm {
  base cipher-suite-alg-base;
  description
    "TLS-DHE-PSK-WITH-AES-128-CCM";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-dhe-psk-with-aes-256-ccm {
  base cipher-suite-alg-base;
  description
    "TLS-DHE-PSK-WITH-AES-256-CCM";
  reference
    "RFC 6655:
```

```
       AES-CCM Cipher Suites for TLS";
}

identity tls-psk-with-aes-128-ccm-8 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-AES-128-CCM-8";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-psk-with-aes-256-ccm-8 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-AES-256-CCM-8";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-psk-dhe-with-aes-128-ccm-8 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-DHE-WITH-AES-128-CCM-8";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-psk-dhe-with-aes-256-ccm-8 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-DHE-WITH-AES-256-CCM-8";
  reference
    "RFC 6655:
       AES-CCM Cipher Suites for TLS";
}

identity tls-ecdhe-ecdsa-with-aes-128-ccm {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-AES-128-CCM";
  reference
```

```
    "RFC 7251:
       AES-CCM ECC Cipher Suites for TLS";
}

identity tls-ecdhe-ecdsa-with-aes-256-ccm {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-AES-256-CCM";
  reference
    "RFC 7251:
       AES-CCM ECC Cipher Suites for TLS";
}

identity tls-ecdhe-ecdsa-with-aes-128-ccm-8 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-AES-128-CCM-8";
  reference
    "RFC 7251:
       AES-CCM ECC Cipher Suites for TLS";
}

identity tls-ecdhe-ecdsa-with-aes-256-ccm-8 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-ECDSA-WITH-AES-256-CCM-8";
  reference
    "RFC 7251:
       AES-CCM ECC Cipher Suites for TLS";
}

identity tls-eccpwd-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECCPWD-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 8492:
       Secure Password Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-eccpwd-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
```

```
    "TLS-ECCPWD-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 8492:
       Secure Password Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-eccpwd-with-aes-128-ccm-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECCPWD-WITH-AES-128-CCM-SHA256";
  reference
    "RFC 8492:
       Secure Password Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-eccpwd-with-aes-256-ccm-sha384 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECCPWD-WITH-AES-256-CCM-SHA384";
  reference
    "RFC 8492:
       Secure Password Ciphersuites for
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-rsa-with-chacha20-poly1305-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-ECDHE-RSA-WITH-CHACHA20-POLY1305-SHA256";
  reference
    "RFC 7905:
       ChaCha20-Poly1305 Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-ecdsa-with-chacha20-poly1305-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-ECDHE-ECDSA-WITH-CHACHA20-POLY1305-SHA256";
  reference
    "RFC 7905:
       ChaCha20-Poly1305 Cipher Suites for
       Transport Layer Security (TLS)";
}
```

```
identity tls-dhe-rsa-with-chacha20-poly1305-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-DHE-RSA-WITH-CHACHA20-POLY1305-SHA256";
  reference
    "RFC 7905:
       ChaCha20-Poly1305 Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-psk-with-chacha20-poly1305-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-PSK-WITH-CHACHA20-POLY1305-SHA256";
  reference
    "RFC 7905:
       ChaCha20-Poly1305 Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-chacha20-poly1305-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-ECDHE-PSK-WITH-CHACHA20-POLY1305-SHA256";
  reference
    "RFC 7905:
       ChaCha20-Poly1305 Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-dhe-psk-with-chacha20-poly1305-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-DHE-PSK-WITH-CHACHA20-POLY1305-SHA256";
  reference
    "RFC 7905:
       ChaCha20-Poly1305 Cipher Suites for
       Transport Layer Security (TLS)";
}

identity tls-rsa-psk-with-chacha20-poly1305-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-RSA-PSK-WITH-CHACHA20-POLY1305-SHA256";
  reference
    "RFC 7905:
       ChaCha20-Poly1305 Cipher Suites for
```

```
       Transport Layer Security (TLS)";
}

identity tls-ecdhe-psk-with-aes-128-gcm-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-ECDHE-PSK-WITH-AES-128-GCM-SHA256";
  reference
    "RFC 8442:
       ECDHE_PSK with AES-GCM and AES-CCM Cipher Suites";
}

identity tls-ecdhe-psk-with-aes-256-gcm-sha384 {
  base cipher-suite-alg-base;
  description
    "TLS-ECDHE-PSK-WITH-AES-256-GCM-SHA384";
  reference
    "RFC 8442:
       ECDHE_PSK with AES-GCM and AES-CCM Cipher Suites";
}

identity tls-ecdhe-psk-with-aes-128-ccm-8-sha256 {
  base cipher-suite-alg-base;
  status deprecated;
  description
    "TLS-ECDHE-PSK-WITH-AES-128-CCM-8-SHA256";
  reference
    "RFC 8442:
       ECDHE_PSK with AES-GCM and AES-CCM Cipher Suites";
}

identity tls-ecdhe-psk-with-aes-128-ccm-sha256 {
  base cipher-suite-alg-base;
  description
    "TLS-ECDHE-PSK-WITH-AES-128-CCM-SHA256";
  reference
    "RFC 8442:
       ECDHE_PSK with AES-GCM and AES-CCM Cipher Suites";
}

// Protocol-accessible Nodes

container supported-algorithms {
  config false;
  description
    "A container for a list of cipher suite algorithms supported
     by the server.";
  leaf-list supported-algorithm {
    type cipher-suite-algorithm-ref;
```

```
      description
        "A cipher suite algorithm supported by the server.";
    }
  }

}
```

<CODE ENDS>

Appendix B. Change Log

This section is to be removed before publishing as an RFC.

B.1. 00 to 01

*Noted that '0.0.0.0' and '::' might have special meanings.

*Renamed "keychain" to "keystore".

B.2. 01 to 02

*Removed the groupings containing transport-level configuration. Now modules contain only the transport-independent groupings.

*Filled in previously incomplete 'ietf-tls-client' module.

*Added cipher suites for various algorithms into new 'ietf-tls-common' module.

B.3. 02 to 03

*Added a 'must' statement to container 'server-auth' asserting that at least one of the various auth mechanisms must be specified.

*Fixed description statement for leaf 'trusted-ca-certs'.

B.4. 03 to 04

*Updated title to "YANG Groupings for TLS Clients and TLS Servers"

*Updated leafref paths to point to new keystore path

*Changed the YANG prefix for ietf-tls-common from 'tlscom' to 'tlscmn'.

*Added TLS protocol verions 1.0 and 1.1.

*Made author lists consistent

*Now tree diagrams reference ietf-netmod-yang-tree-diagrams

*Updated YANG to use typedefs around leafrefs to common keystore paths

*Now inlines key and certificates (no longer a leafref to keystore)

B.5. 04 to 05

* Merged changes from co-author.

B.6. 05 to 06

* Updated to use trust anchors from trust-anchors draft (was keystore draft)

* Now Uses new keystore grouping enabling asymmetric key to be either locally defined or a reference to the keystore.

B.7. 06 to 07

* factored the tls-[client|server]-groupings into more reusable groupings.

* added if-feature statements for the new "x509-certificates" feature defined in draft-ietf-netconf-trust-anchors.

B.8. 07 to 08

* Added a number of compatibility matrices to Section 5 (thanks Frank!)

* Clarified that any configured "cipher-suite" values need to be compatible with the configured private key.

B.9. 08 to 09

* Updated examples to reflect update to groupings defined in the keystore draft.

* Add TLS keepalives features and groupings.

* Prefixed top-level TLS grouping nodes with 'tls-' and support mashups.

* Updated copyright date, boilerplate template, affiliation, and folding algorithm.

B.10. 09 to 10

* Reformatted the YANG modules.

B.11. 10 to 11

* Collapsed all the inner groupings into the top-level grouping.

* Added a top-level "demux container" inside the top-level grouping.

* Added NACM statements and updated the Security Considerations section.

* Added "presence" statements on the "keepalive" containers, as was needed to address a validation error that appeared after adding the "must" statements into the NETCONF/RESTCONF client/server modules.

* Updated the boilerplate text in module-level "description" statement to match copyeditor convention.

B.12. 11 to 12

* In server model, made 'client-authentication' a 'presence' node indicating that the server supports client authentication.

* In the server model, added a 'required-or-optional' choice to 'client-authentication' to better support protocols such as RESTCONF.

* In the server model, added a 'local-or-external' choice to 'client-authentication' to better support consuming data models that prefer to keep client auth with client definitions than in a model principally concerned with the "transport".

* In both models, removed the "demux containers", floating the nacm:default-deny-write to each descendant node, and adding a note to model designers regarding the potential need to add their own demux containers.

* Fixed a couple references (section 2 --> section 3)

B.13. 12 to 13

* Updated to reflect changes in trust-anchors drafts (e.g., s/trust-anchors/truststore/g + s/pinned.//)

B.14. 12 to 13

* Removed 'container' under 'client-identity' to match server model.

* Updated examples to reflect change grouping in keystore module.

B.15. 13 to 14

* Removed the "certificate" container from "client-identity" in the ietf-tls-client module.

* Updated examples to reflect ietf-crypto-types change (e.g., identities --> enumerations)

B.16. 14 to 15

* Updated "server-authentication" and "client-authentication" nodes from being a leaf of type "ts:certificates-ref" to a container that uses "ts:local-or-truststore-certs-grouping".

B.17. 15 to 16

* Removed unnecessary if-feature statements in the -client and -server modules.

* Cleaned up some description statements in the -client and -server modules.

* Fixed a canonical ordering issue in ietf-tls-common detected by new pyang.

B.18. 16 to 17

* Removed choice local-or-external by removing the 'external' case and flattening the 'local' case and adding a "client-auth-supported" feature.

* Removed choice required-or-optional.

* Updated examples to include the "*-key-format" nodes.

* Augmented-in "must" expressions ensuring that locally-defined public-key-format are "ct:tls-public-key-format" (must expr for ref'ed keys are TBD).

B.19. 17 to 18

* Removed the unused "external-client-auth-supported" feature.

* Made client-indentity optional, as there may be over-the-top auth instead.

* Added augment to uses of local-or-keystore-symmetric-key-grouping for a psk "id" node.

* Added missing presence container "psks" to ietf-tls-server's "client-authentication" container.

* Updated examples to reflect new "bag" addition to truststore.

*Removed feature-limited caseless 'case' statements to improve tree diagram rendering.

*Refined truststore/keystore groupings to ensure the key formats "must" be particular values.

*Switched to using truststore's new "public-key" bag (instead of separate "ssh-public-key" and "raw-public-key" bags).

*Updated client/server examples to cover ALL cases (local/ref x cert/raw-key/psk).

B.20. 18 to 19

*Updated the "keepalives" containers in part to address Michal Vasko's request to align with RFC 8071, and in part to better align to RFC 6520.

*Removed algorithm-mapping tables from the "TLS Common Model" section

*Removed the 'algorithm' node from the examples.

*Renamed both "client-certs" and "server-certs" to "ee-certs"

*Added a "Note to Reviewers" note to first page.

B.21. 19 to 20

*Modified the 'must' expression in the "ietf-tls-client:server-authention" node to cover the "raw-public-keys" and "psks" nodes also.

*Added a "must 'ca-certs or ee-certs or raw-public-keys or psks'" statement to the ietf-tls-server:client-authentication" node.

*Added "mandatory true" to "choice auth-type" and a "presence" statement to its ancestor.

*Expanded "Data Model Overview section(s) [remove "wall" of tree diagrams].

*Moved the "ietf-tls-common" module section to proceed the other two module sections.

*Updated the Security Considerations section.

B.22. 20 to 21

* Updated examples to reflect new "cleartext-" prefix in the crypto-types draft.

B.23. 21 to 22

* In both the "client-authentication" and "server-authentication" subtrees, replaced the "psks" node from being a P-container to a leaf of type "empty".

* Cleaned up examples (e.g., removed FIXMEs)

* Fixed issues found by the SecDir review of the "keystore" draft.

* Updated the "psk" sections in the "ietf-tls-client" and "ietf-tls-server" modules to more correctly reflect RFC 4279.

B.24. 22 to 23

* Addressed comments raised by YANG Doctor in the ct/ts/ks drafts.

B.25. 23 to 24

* Added missing reference to "FIPS PUB 180-4".

* Added identity "tls-1.3" and updated description statement in other identities indicating that the protocol version is obsolete and enabling the feature is NOT RECOMMENDED.

* Added XML-comment above examples explaining the reason for the unexpected top-most element's presence.

* Added missing "client-ident-raw-public-key" and "client-ident-psk" featutes.

* Aligned modules with `pyang -f` formatting.

* Fixed nits found by YANG Doctor reviews.

* Added a 'Contributors' section.

B.26. 24 to 25

* Added TLS 1.3 references.

* Clarified support for various TLS protocol versions.

* Moved algorithms in ietf-tls-common (plus more) to IANA-maintained modules

*Added "config false" lists for algorithms supported by the server.

*Fixed issues found during YANG Doctor review.

B.27. 25 to 26

*Replaced "base64encodedvalue==" with "BASE64VALUE=" in examples.

*Minor editorial nits

B.28. 26 to 27

*Fixed up the 'WG Web' and 'WG List' lines in YANG module(s)

*Fixed up copyright (i.e., s/Simplified/Revised/) in YANG module(s)

*Created identityref-based typedef for the IANA alg identity base.

*Major update to support TLS 1.3.

B.29. 27 to 28

*Fixed draft text to refer to new "identity" values (e.g., s/tls-1.3/tls13).

*Added ietf-tls-common:generate-public-key() RPC.

Acknowledgements

The authors would like to thank for following for lively discussions on list and in the halls (ordered by first name): Alan Luchuk, Andy Bierman, Balazs Kovacs, Benoit Claise, Bert Wijnen, David Lamparter, Dhruv Dhody, Gary Wu, Henk Birkholz, Juergen Schoenwaelder, Ladislav Lhotka, Liang Xia, Martin Bjoerklund, Mehmet Ersue, Michal Vasko, Phil Shafer, Radek Krejci, Sean Turner, and Tom Petch.

Contributors

Special acknowledgement goes to Gary Wu who contributed the "ietf-tls-common" module, and Tom Petch who carefully ensured that references were set correctly throughout.

Author's Address

Kent Watsen
Watsen Networks

Email: kent+ietf@watsen.net