Corrections to the ISO/IEC 29150 Signcryption ASN.1 Schema

Phillip H. Griffin
phil@phillipgriffin.com

Abstract. In this note, defects in the schema of the first edition of the ISO/IEC 29150 Signcryption standard are described, and a corrected ASN.1 module is proposed. An example signcryption algorithm identifier value is defined and binary and markup representations of this value are presented. Although the schema errors are small and do not affect the textual content of the standard, programming language code generation and other tools cannot process the schema unless it is correct.

1 Introduction

The ISO/IEC 29150 Signcryption standard [1] provides a schema for signcryption mechanism and cryptographic algorithm identification. The schema is defined as an ASN.1 module [2]. Syntax errors in the published schema prohibit its use by ASN.1 tools. These minor defects may lead to misinterpretation by readers and to the development of implementations that fail to interoperate.

Type `SCparameters` has two components that are meant to identify a key derivation function (`kdf`) and a hash function (`hash`). The published version of the ISO/IEC 29150 schema contains the following definition of type `SCparameters`:

```
SCparameters ::= SEQUENCE {
   kdf   SCKDFfunction,
   hash  SCHashFunction
}
```

For this definition of type `SCparameters` to be valid, `SCKDFfunction` and `SCHashFunction` must be valid ASN.1 types. However, `SCKDFfunction` and `SCHashFunction` are ASN.1 information object sets of class `ALGORITHM`, defined as follows:

```
SCHashFunction ALGORITHM ::= {
   {OID id-sha1 PARMS NullParms} |
   {OID id-sha256 PARMS NullParms } |
   {OID id-sha384 PARMS NullParms } |
   {OID id-sha512 PARMS NullParms },
   ... -- expect more hash functions here
}

SCKDFfunction ALGORITHM ::= {
   {OID id-kdf-kdf1 PARMS SCHashFunction} |
   {OID id-kdf-kdf2 PARMS SCHashFunction},
   ... -- expect additional KDF functions here
}
```

Other aspects of the published schema that are not errors can be improved. The schema does not define a signcryption algorithm identifier type for reference by implementers and other standards. The schema imports the `HashFunctionAlgs` information object set, though this set of algorithms is never used and can be eliminated. The `SCHashFunction` and `SCKDFfunction` information object sets described above duplicate the content of encryption algorithm information object sets already defined in the ISO/IEC 18033 standard [3].

These object sets can be referenced and not redefined. Their redefinition in ISO/IEC 29150 requires additional information object identifiers (OIDs) to be imported into the module, and for the creation of duplicate definitions for the `id-kdf-kdf1` and the `id-kdf-kdf2` key derivation functions. These definitions can be eliminated.

2 Schema

The following ASN.1 schema contains corrections to the schema published in ISO/IEC 29150:2011. This module contains valid syntax that can be used as input to ASN.1 syntax checking, schema validation, and programming language code generation tools. The ISO/IEC 29150 module information object identifier is reused here for clarity.

```
Signcryption {
  iso(1) standard(0) signcryption(29150)
      asn1-module(0) signcryption-mechanisms(0) version(1)
}
  DEFINITIONS EXPLICIT TAGS ::= BEGIN

IMPORTS

   HashFunction, KeyDerivationFunction
      FROM EncryptionAlgorithms-2 {
         iso(1) standard(0) encryption-algorithms(18033) part(2)
            asn1-module(0) algorithm-object-identifiers(0) };


SigncryptionAlgorithmIdentifier ::=
                 AlgorithmIdentifier {{ SigncryptionMechanism }}

SigncryptionMechanism ALGORITHM ::= {
   { OID signcryption-mechanism-dlsc    PARMS  SCparameters } |
   { OID signcryption-mechanism-ecdlsc  PARMS  SCparameters } |
   { OID signcryption-mechanism-ifsc    PARMS  SCparameters } |
   { OID signcryption-mechanism-ets     PARMS  SCparameters },

   ... -- Expect additional signcryption mechanisms --
}

SCparameters ::= SEQUENCE {
   kdf   KeyDerivationFunction,
   hash  HashFunction
}

-- Cryptographic algorithm identification --

OID ::= OBJECT IDENTIFIER  -- Alias --

is29150 OID ::= { iso(1) standard(0) signcryption(29150) }

mechanism OID ::= { is29150 mechanisms(1) }

signcryption-mechanism-dlsc    OID ::= { mechanism dlsc(1) }
signcryption-mechanism-ecdlsc  OID ::= { mechanism ecdlsc(2) }
signcryption-mechanism-ifsc    OID ::= { mechanism ifsc(3) }
signcryption-mechanism-ets     OID ::= { mechanism ets(4) }

AlgorithmIdentifier { ALGORITHM:IOSet } ::= SEQUENCE {
   algorithm   ALGORITHM.&id({IOSet}),
   parameters  ALGORITHM.&Type({IOSet}{@algorithm})  OPTIONAL
}

ALGORITHM ::= CLASS {
   &id    OBJECT IDENTIFIER  UNIQUE,
   &Type  OPTIONAL
}
  WITH SYNTAX { OID &id [PARMS &Type] }

END  -- Signcryption --
```

3 Example

Type `SigncryptionAlgorithmIdentifier` is defined as the following parameterized type:

```
SigncryptionAlgorithmIdentifier ::=
                  AlgorithmIdentifier {{ SigncryptionMechanism }}
```

When expanded using the provided parameter, the information object set `SigncryptionMechanism`, this parameterized type becomes

```
SigncryptionAlgorithmIdentifier ::= ::= SEQUENCE {
   algorithm  ALGORITHM.&id({SigncryptionMechanism }),
   parameters ALGORITHM.&Type({SigncryptionMechanism }
                                          {@algorithm}) OPTIONAL
}
```

The information object set `SigncryptionMechanism` forms a table constraint on the `algorithm` and `parameters` components of type `SigncryptionAlgorithmIdentifier`. The types of these two components are based on the `&id` and `&Type` fields of information object class `ALGORITHM`.

An example value[1] of type `SigncryptionAlgorithmIdentifier` expressed using the ASN.1 XML Value Notation could be defined as follows:

```
 1  <SigncryptionAlgorithmIdentifier>
 2     <algorithm>1.0.29150.1.3</algorithm>
 3     <parameters>
 4        <SCparameters>
 5           <kdf>
 6              <algorithm>1.0.18033.2.5.1</algorithm>
 7              <parameters>
 8                 <HashFunction>
 9                    <algorithm>
10                       2.16.840.1.101.3.4.2.2
11                    </algorithm>
12                 </HashFunction>
13              </parameters>
14           </kdf>
15           <hash>
16              <algorithm>2.16.840.1.101.3.4.2.2</algorithm>
17           </hash>
18        </SCparameters>
19     </parameters>
20  </SigncryptionAlgorithmIdentifier>
```

On line 2, the integer factorization based signcryption (IFSC) mechanism is identified as the signcryption algorithm. The parameters associated with the IFSC algorithm on lines 3-19 consist of two cryptographic functions, a key derivation function (sometimes referred to as a mask generation function) and a hash or message digest function.

The first of these cryptographic functions, the key derivation function (KDF) is identified on line 6. It is the KDF1 family of functions defined in the ISO/IEC 18033-2 standard, which rely on the hash functions defined in the ISO/IEC 10118-3 standard. The parameters of the KDF1 algorithm are the SHA-384 hash function indicated on lines

[1] All of the encoded values in this document were produced using the ASN-1Step tool, an interactive application development and testing environment from OSS Nokalva (http://www.oss.com).

9-11. The second cryptographic function is the SHA-384 hash function identified on line 16.

The same value can be defined using the ASN.1 Basic Value Notation as follows:

```
value SigncryptionAlgorithmIdentifier ::= {
   algorithm { 1 0 29150 1 ifsc(3) },
   parameters  SCparameters : {
      kdf {
         algorithm { 1 0 18033 2 5 kdf(1) },
         parameters HashFunction : {
            algorithm { 2 16 840 1 101 3 4 2 sha384(2) }
         }
      },
      hash {
         algorithm { 2 16 840 1 101 3 4 2 sha384(2) }
      }
   }
}
```

In either value notation form, this example value can be represented using DER, the ASN.1 Distinguished Encoding Rules [4] in 49 bytes, shown here using hexadecimal notation where two characters represent one byte:

```
302F0606 2881E35E 01033025 30160607 28818C71 02050130 0B060960
86480165 03040202 300B0609 60864801 65030402 02
```

The same example value can be represented using a canonical variant of XER, the XML Encoding Rules [5] of ASN.1 in 363 bytes of XML markup [6], shown here formatted for reading ease as an XML Document:

```
<?xml version="1.0" encoding="UTF-8"?>
<SigncryptionAlgorithmIdentifier>
   <algorithm>1.0.29150.1.3</algorithm>
   <parameters>
      <SCparameters>
         <kdf>
            <algorithm>1.0.18033.2.5.1</algorithm>
            <parameters>
               <HashFunction>
                  <algorithm>2.16.840.1.101.3.4.2.2</algorithm>
               </HashFunction>
            </parameters>
         </kdf>
         <hash>
            <algorithm>2.16.840.1.101.3.4.2.2</algorithm>
         </hash>
      </SCparameters>
   </parameters>
</SigncryptionAlgorithmIdentifier>
```

References

[1] ISO/IEC 29150:2011 Information technology – Security techniques - Signcryption.

[2] ITU-T Recommendation X.680-series | ISO/IEC 8824 (All parts), *Information Technology - Abstract Syntax Notation One (ASN.1).* Retrieved June 20, 2012, from http://www.itu.int/rec/T-REC-X/en

[3] ISO/IEC 18033-2:2006 Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers.

[4] ITU-T Recommendation X.690 | ISO/IEC 8825-1, *Information Technology - ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules CER), Distinguished Encoding Rules (DER).* Retrieved June 20, 2012, from http://www.itu.int/rec/T-REC-X.690-200811-I/en

[5] ITU-T Recommendation X.693 | ISO/IEC 8825-4, *Information Technology - ASN.1 Encoding Rules: Specification of XML Encoding Rules (XER).* Retrieved June 20, 2012, from http://www.itu.int/rec/T-REC-X.693-200811-I/en

[6] W3C Recommendation (2000). *Extensible Markup Languqge (XML) 1.0 (Second Edition)*. Retrieved June 20, 2012, from http://www.w3.org/TR/2000/REC-xml-20001006