INTERNATIONAL STANDARD

ISO/IEC 18033-2

First edition
2006-05-01

Information technology — Security techniques — Encryption algorithms —

Part 2:
Asymmetric ciphers

Technologies de l'information — Techniques de sécurité — Algorithmes de chiffrement —

Partie 2: Chiffres asymétriques

Reference number
ISO/IEC 18033-2:2006(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

ISO/IEC 18033-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

ISO/IEC 18033 consists of the following parts, under the general title *Information technology — Security techniques — Encryption algorithms*:

— *Part 1: General*

— *Part 2: Asymmetric ciphers*

— *Part 3: Block ciphers*

— *Part 4: Stream ciphers*

Introduction

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this International Standard may involve the use of patents.

The ISO and IEC take no position concerning the evidence, validity and scope of this patent right. The holder of this patent right has assured the ISO and IEC that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with the ISO and IEC. Information may be obtained from:

ISO/IEC JTC 1/SC 27 Standing Document 8 (SD8) "Patent Information"

Standing Document 8 (SD8) is publicly available at: http://www.ni.din.de/sc27

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — Security techniques — Encryption algorithms —

Part 2: Asymmetric ciphers

1 Scope

This part of ISO/IEC 18033 specifies several asymmetric ciphers. These specifications prescribe the functional interfaces and correct methods of use of such ciphers in general, as well as the precise functionality and cipher text format for several specific asymmetric ciphers (although conforming systems may choose to use alternative formats for storing and transmitting cipher-texts).

A normative annex (Annex A) gives ASN.1 syntax for object identifiers, public keys, and parameter structures to be associated with the algorithms specified in this part of ISO/IEC 18033.

However, these specifications do not prescribe protocols for reliably obtaining a public key, for proof of possession of a private key, or for validation of either public or private keys; see ISO/IEC 11770-3 for guidance on such key management issues.

The asymmetric ciphers that are specified in this part of ISO/IEC 18033 are indicated in Clause 7.6.

NOTE Briefly, the asymmetric ciphers are:

— ECIES-HC; PSEC-HC; ACE-HC: generic hybrid ciphers based on ElGamal encryption;

— RSA-HC: a generic hybrid cipher based on the RSA transform;

— RSAES: the OAEP padding scheme applied to the RSA transform;

— HIME(R): a scheme based on the hardness of factoring.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9797-1:1999, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher*

ISO/IEC 9797-2:2002, *Information technology — Security techniques— Message Authentication Codes (MACs) — Part 2: Mechanisms using a dedicated hash-function*

ISO/IEC 10118-2:2000, *Information technology — Security techniques — Hash-functions — Part 2: Hash-functions using an n-bit block cipher*

ISO/IEC 10118-3:2004, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions*

ISO/IEC 18033-3:2005, *Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers*

3 Definitions

For the purposes of this document, the following terms and definitions apply.

NOTE Where appropriate, forward references are given to clauses which contain more detailed definitions and/or further elaboration.

3.1
asymmetric cipher
system based on asymmetric cryptographic techniques whose public transformation is used for encryption and whose private transformation is used for decryption
[ISO/IEC 18033-1]
NOTE See Clause 7.

3.2
asymmetric cryptographic technique
cryptographic technique that uses two related transformations, a public transformation (defined by the public key) and a private transformation (defined by the private key). The two transformations have the property that, given the public transformation, it is computationally infeasible to derive the private transformation.
[ISO/IEC 11770-1:1996]

3.3
asymmetric key pair
pair of related keys, a public key and a private key, where the private key defines the private transformation and the public key defines the public transformation
[ISO/IEC 9798-1:1997]
NOTE See Clauses 7, 8.1.

3.4
bit
one of the two symbols `0' or `1'
NOTE See Clause 5.2.1.

3.5
bit string
sequence of bits
NOTE See Clause 5.2.1.

3.6
block
string of bits of a defined length
[ISO/IEC 18033-1]
NOTE In this part of ISO/IEC 18033, a block will be restricted to be an octet string (interpreted in a natural way as a bit string).

3.7
block cipher
symmetric cipher with the property that the encryption algorithm operates on a block of plain-text, i.e., a string of bits of a defined length, to yield a block of cipher text
[ISO/IEC 18033-1]
NOTE See Clause 6.4.
NOTE In this part of ISO/IEC 18033, plaintext/cipher text blocks will be restricted to be octet strings (interpreted in a natural way as bit strings).

3.8
cipher
cryptographic technique used to protect the confidentiality of data, and which consists of three component processes: an encryption algorithm, a decryption algorithm, and a method for generating keys
[ISO/IEC 18033-1]

3.9
cipher text
data which has been transformed to hide its information content
[ISO/IEC 10116:1997]

3.10
concrete group
explicit description of a finite abelian group, together with algorithms for performing the group operation and for encoding and decoding group elements as octet strings
NOTE See Clause 10.1.

3.11
cryptographic hash function
function that maps octet strings of any length to octet strings of fixed length, such that it is computationally infeasible to find correlations between inputs and outputs, and such that given one part of the output, but not the input, it is computationally infeasible to predict any bit of the remaining output. The precise security requirements depend on the application.
NOTE See Clause 6.1.

3.12
data encapsulation mechanism
cryptographic mechanism, based on symmetric cryptographic techniques, which protects both the confidentiality and the integrity of data
NOTE See Clause 8.2.

3.13
decryption
reversal of the corresponding encryption
[ISO/IEC 11770-1:1996]

3.14
decryption algorithm
process which transforms ciphertext into plaintext
[ISO/IEC 18033-1]

3.15
encryption
(reversible) transformation of data by a cryptographic algorithm to produce cipher text, i.e., to hide the information content of the data
[ISO/IEC 9797-1]

3.16
explicitly given finite field
finite field that is represented explicitly in terms of its characteristic and a multiplication table for a basis of the field over the underlying prime field
NOTE See Clause 5.3.

3.17
encryption algorithm
process which transforms plaintext into cipher text
[ISO/IEC 18033-1]

3.18
encryption option
option that may be passed to the encryption algorithm of an asymmetric cipher, or of a key encapsulation mechanism, to control the formatting of the output cipher text
NOTE See Clauses 7, 8.1.

3.19
field
mathematical notion of a field, i.e., a set of elements, together with binary operations for addition and multiplication on this set, such that the usual field axioms apply

3.20
finite abelian group
group such that the underlying set of elements is finite, and such that the underlying binary operation is commutative

3.21
finite field
field such that the underlying set of elements is finite

3.22
group
mathematical notion of a group, i.e., a set of elements, together with a binary operation on this set, such that the usual group axioms apply

3.23
hybrid cipher
asymmetric cipher that combines both asymmetric and symmetric cryptographic techniques

3.24
key
sequence of symbols that controls the operation of a cryptographic transformation (e.g., encryption, decryption)
[ISO/IEC 11770-1:1996]

3.25
key derivation function
a function that maps octet strings of any length to octet strings of an arbitrary, specified length, such that it is computationally infeasible to find correlations between inputs and outputs, and such that given one part of the output, but not the input, it is computationally infeasible to predict any bit of the remaining output. The precise security requirements depend on the application.
NOTE See Clause 6.2.

3.26
key encapsulation mechanism
similar to an asymmetric cipher, but the encryption algorithm takes as input a public key and generates a secret key and an encryption of this secret key
NOTE See Clause 8.1.

3.27
key generation algorithm
method for generating asymmetric key pairs
NOTE See Clauses 7, 8.1.

3.28
label
octet string that is input to both the encryption and decryption algorithms of an asymmetric cipher, and of a data encapsulation mechanism. A label is public information that is bound to the cipher text in a non-malleable way
NOTE See Clauses 7, 8.2.

3.29
length
length of a string of digits or the representation of an integer
Specifically:
(1) length of a bit string is the number of bits in the string
NOTE See Clause 5.2.1.
(2) length of an octet string is the number of octets in the string
NOTE See Clause 5.2.2.
(3) bit length of a non-negative integer n is the number of bits in its binary representation, i.e., dlog$_2$(n + 1)
NOTE See Clause 5.2.4.
(4) octet length of a non-negative integer n is the number of digits in its representation base 256, i.e., dlog$_{256}$(n + 1)
NOTE See Clause 5.2.4.

3.30
message authentication code (MAC)
string of bits which is the output of a MAC algorithm
[ISO/IEC 9797-1]
NOTE See Clause 6.3.
NOTE In this part of ISO/IEC 18033, a MAC will be restricted to be an octet string (interpreted in a natural way as a bit string).

3.31
MAC algorithm
algorithm for computing a function which maps strings of bits and a secret key to fixed-length strings of bits, satisfying the following two properties:
- for any key and any input string, the function can be computed efficiently;
- for any fixed key, and given no prior knowledge of the key, it is computationally infeasible to compute the function value on any new input string, even given knowledge of the set of input strings and corresponding function values, where the value of the ith input string may have been chosen after observing the value of the first *i* - 1 function values.
[ISO/IEC 9797-1]
NOTE See Clause 6.3.
NOTE In this part of ISO/IEC 18033, the input and output strings of a MAC algorithm will be restricted to be octet strings (interpreted in a natural way as bit strings).

3.32
octet
a bit string of length 8
NOTE See Clause 5.2.2.

3.33
octet string
a sequence of octets
NOTE See Clause 5.2.2.
NOTE When appropriate, an octet string may be interpreted as a bit string, simply by concatenating all of the component octets.

3.34
plaintext
unencrypted information
[ISO/IEC 10116:1997]

3.35
prefix free set
a set S of bit/octet strings such that there do not exist strings $x \neq y \in S$ such that x is a prefix of y

3.36
primitive
a function used to convert between data types

3.37
private key
the key of an entity's asymmetric key pair which should only be used by that entity
[ISO/IEC 11770-1:1996]
NOTE See Clauses 7, 8.1.

3.38
public key
the key of an entity's asymmetric key pair which can be made public
[ISO/IEC 11770-1:1996]
NOTE See Clauses 7, 8.1.

3.39
secret key
key used with symmetric cryptographic techniques by a specified set of entities
[ISO/IEC 11770-3:1999]

3.40
symmetric cipher
cipher based on symmetric cryptographic techniques that uses the same secret key for both the encryption and decryption algorithms
[ISO/IEC 18033-1]

3.41
system parameters
choice of parameters that selects a particular cryptographic scheme or function from a family of cryptographic schemes or functions

4 Symbols and notation

For the purposes of this document, the following symbols and notation apply.

NOTE Where appropriate, forward references are given to clauses which contain more detailed definitions and/or further elaboration.

$\lfloor x \rfloor$ the largest integer less than or equal to the real number x. For example,
$\lfloor 5 \rfloor = 5, \lfloor 5.3 \rfloor = 5$, and $\lfloor -5.3 \rfloor = -6$

$\lceil x \rceil$ the smallest integer greater than or equal to the real number *x*. For example,
$\lceil 5 \rceil = 5, \lceil 5.3 \rceil = 6$, and $\lceil -5.3 \rceil = -6$

$[a..b]$ the interval of integers from *a* to *b*, including both *a* and *b*

$[a..b)$ the interval of integers from *a* to *b*, including *a* but not *b*

$|X|$ if X is a finite set, then the cardinality of X; if X is a finite abelian group or a finite field, then the cardinality of the underlying set of elements; if X is a real number, then the absolute value of X; if X is a bit/octet string, then the length in bits/octets of the string
NOTE See Clauses 5.2.1, 5.2.2.

$x \oplus y$ if *x* and *y* are bit/octet strings of the same length, the bit-wise exclusive-or (XOR) of the two strings.
NOTE See Clauses 5.2.1, 5.2.2.

$\langle x_1, . . ., x_l \rangle$ if $x_1, . . ., x_l$ are bits/octets, the bit/octet string of length l consisting of the bits/octets x1; : : : ; xl, in the given order
NOTE See Clauses 5.2.1, 5.2.2.

$x \| y$ if *x* and *y* are bit/octet strings, the concatenation of the two strings *x* and *y*, resulting in the string consisting of x followed by *y*
NOTE See Clauses 5.2.1, 5.2.2.

gcd(a, b) for integers *a* and *b*, the greatest common divisor of *a* and *b*, i.e., the largest positive integer that divides both *a* and *b* (or 0 if $a = b = 0$)

$a \mid b$ relation between integers *a* and *b* that holds if and only if *a* divides *b*, i.e., there exists an integer *c* such that $b = ac$

$a \equiv b \pmod{n}$ for a non-zero integer *n*, a relation between integers *a* and *b* that holds if and only if *a* and *b* are congruent modulo *n*, i.e., $n \mid (a - b)$

$a \bmod n$ for integer *a* and positive integer *n*, the unique integer $r \in [0 . . n)$ such that $r \equiv a \pmod{n}$

$a^{-1} \bmod n$ for integer a and positive integer n, such that gcd$(a; n) = 1$, the unique integer $b \in [0..n)$ such that $ab \equiv 1 \pmod{n}$

F^* for a field *F*, the multiplicative group of units of *F*

0_F for a field *F*, the additive identity (zero element) of *F*

1_F for a field *F*, the multiplicative identity of *F*

BS2IP bit string to integer conversion primitive
NOTE See Clause 5.2.5.

EC2OSP	elliptic curve to octet string conversion primitive. *(See Clause 5.4.3.)*
FE2OSP	field element to octet string conversion primitive. *(See Clause 5.3.1.)*
FE2IP	field element to integer conversion primitive. *(See Clause 5.3.1.)*
I2BSP	integer to bit string conversion primitive. *(See Clause 5.2.5.)*
I2OSP	integer to octet string conversion primitive. *(See Clause 5.2.5.)*
OS2ECP	octet string to elliptic curve conversion primitive. *(See Clause 5.4.3.)*
OS2FEP	octet string to field element conversion primitive. *(See Clause 5.3.1.)*
OS2IP	octet string to integer conversion primitive. *(See Clause 5.2.5.)*
$Oct(m)$	the octet whose integer value is m. *(See Clause 5.2.4.)*
$\mathcal{L}(n)$	the length in octets of an integer n. *(See Clause 5.2.5.)*

5 Mathematical conventions

This clause describes certain mathematical conventions used in this part of ISO/IEC 18033, including the representation of mathematical objects, and primitives for data type conversion.

5.1 Functions and algorithms

For ease of presentation, functions and probabilistic functions (i.e., functions whose value depends not only on the input value but also on a randomly chosen auxiliary value) are often specified in algorithmic form. Except where explicitly noted, an implementor may choose to employ any equivalent algorithm (i.e., one which yields the same function or probabilistic function). Moreover, in the case of probabilistic functions, when the algorithm describing the function indicates that a random value should be generated, an implementor shall use an appropriate random generator to generate this value (see ISO/IEC 18031 for more guidance on this issue).

In describing a function in algorithmic terms, the following convention is adopted. An algorithm either computes a value, or alternatively, it may **fail**. By convention, if an algorithm **fails**, then unless otherwise specified, another algorithm that invokes this algorithm as a sub-routine also **fails**.

NOTE Thus, **failing** is analogous to the notion of "throwing an exception" in many programming languages; however, it can also be viewed as returning a special value that is by definition distinct from all values returned by the algorithm when it does not **fail**. With this latter interpretation of **failing**, an algorithm still properly describes a function. The details of how an implementation achieves the effect of **failing** are not specified here. However, in a typical implementation, an algorithm may return an "error code" of some sort to its environment that indicates the reason for the failure. It should be noted that in some cases, for reasons of security, the implementation should take care *not* to reveal the precise cause of certain types of errors.

5.2 Bit strings and octet strings

5.2.1 Bits and bit strings

A *bit* is one of the two symbols '0' or '1'.

A *bit string* is a sequence of bits. For bits $x_1, \ldots, x_l$, $\langle x_1, \ldots, x_l \rangle$ denotes the bit string of length l consisting of the bits $x_1, \ldots, x_l$, in the given order.

For a bit string $x = \langle x_1, \ldots, x_l \rangle$, the length l of x is denoted by $|x|$, and if $l > 0$, x_1 is called the *first* bit of x, and x_l the *last* bit of x.

For bit strings x and y, $x \,\|\, y$ denotes the concatenation of x and y; that is, if $x = \langle x_1, \ldots, x_l \rangle$ and $y = \langle y_1, \ldots, y_m \rangle$, then $x \,\|\, y = \langle x_1, \ldots, x_l, y_1, \ldots, y_m \rangle$.

For bit strings x and y of equal length, $x \oplus y$ denotes the bit-wise exclusive-or (XOR) of x and y.

The bit string of length zero is called the *null* bit string.

NOTE No special subscripting operator is defined for bit strings. Thus, if x is a bit string, x_i does not necessarily denote any particular bit of x.

5.2.2 Octets and octet strings

An *octet* is a bit string of length 8.

An *octet string* is a sequence of octets.

For octets $x_1, \ldots, x_l$, $\langle x_1, \ldots, x_l \rangle$ denotes the octet string of length l consisting of the octets $x_1, \ldots, x_l$, in the given order.

For an octet string $x = \langle x_1, \ldots, x_l \rangle$, the length l of x is denoted by $|x|$, and if $l > 0$, x_1 is called the *first* octet of x, and x_l the *last* octet of x.

For octet strings x and y, $x \,\|\, y$ denotes the concatenation of x and y; that is, if $x = \langle x_1, \ldots, x_l \rangle$ and $y = \langle y_1, \ldots, y_m \rangle$, then $x \,\|\, y = \langle x_1, \ldots, x_l, y_1, \ldots, y_m \rangle$.

For octet strings x and y of equal length, $x \oplus y$ denotes the bit-wise exclusive-or (XOR) of x and y.

The octet string of length zero is called the *null* octet string.

NOTE 1 No special subscripting operator is defined for octet strings. Thus, if x is an octet string, x_i does not necessarily denote any particular octet of x.

NOTE 2 Note that since an octet is a bit string of length 8, if x and y are octets, then $x \,\|\, y$ is a *bit* string of length 16, $\langle x \rangle$ and $\langle y \rangle$ are each *octet* strings of length 1, and $\langle x \rangle \,\|\, \langle y \rangle = \langle x, y \rangle$ is an *octet* string of length 2.