INTERNATIONAL STANDARD

ISO/IEC 11770-3

Third edition
2015-08-01

Information technology — Security techniques — Key management —

Part 3: Mechanisms using asymmetric techniques

Technologies de l'information — Techniques de sécurité — Gestion de clés —

Partie 3: Mécanismes utilisant des techniques asymétriques

Reference number
ISO/IEC 11770-3:2015(E)

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 27, *Security techniques*.

This third edition cancels and replaces the second edition (ISO/IEC 11770-3:2008 with ISO/IEC 11770-3/Cor1:2009), which has been technically revised.

ISO/IEC 11770 consists of the following parts, under the general title *Information technology — Security techniques — Key management*:

— *Part 1: Framework*

— *Part 2: Mechanisms using symmetric techniques*

— *Part 3: Mechanisms using asymmetric techniques*

— *Part 4: Mechanisms based on weak secrets*

— *Part 5: Group key management*

— *Part 6: Key derivation*

Further parts may follow.

Introduction

This part of ISO/IEC 11770 describes schemes that can be used for key agreement and schemes that can be used for key transport.

Public key cryptosystems were first proposed in the seminal paper by Diffie and Hellman in 1976. The security of many such cryptosystems is based on the presumed intractability of solving the discrete logarithm problem over certain finite fields. Other public key cryptosystems such as RSA are based on the difficulty of the integer factorization problem.

A third class of public key cryptosystems is based on elliptic curves. The security of such a public key system depends on the difficulty of determining discrete logarithms in the group of points of an elliptic curve. When based on a carefully chosen elliptic curve, this problem is, with current knowledge, much harder than the factorization of integers or the computation of discrete logarithms in a finite field of comparable size. All known general purpose algorithms for determining elliptic curve discrete logarithms take exponential time. Thus, it is possible for elliptic curve based public key systems to use much shorter parameters than the RSA system or the classical discrete logarithm based systems that make use of the multiplicative group of some finite field. This yields significantly shorter digital signatures, as well as system parameters, and allows for computations using smaller integers.

This part of ISO/IEC 11770 includes mechanisms based on the following:

— finite fields;

— elliptic curves;

— bilinear pairings.

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.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have assured ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC. Information may be obtained from those in Annex H.

Attention is drawn to the possibility that some of the elements of this document 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.

ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain on-line databases of patents relevant to their standards. Users are encouraged to consult the databases for the most up to date information concerning patents.

Information technology — Security techniques — Key management —

Part 3:
Mechanisms using asymmetric techniques

1 Scope

This part of ISO/IEC 11770 defines key management mechanisms based on asymmetric cryptographic techniques. It specifically addresses the use of asymmetric techniques to achieve the following goals.

a) Establish a shared secret key for use in a symmetric cryptographic technique between two entities *A* and *B* by key agreement. In a secret key agreement mechanism, the secret key is computed as the result of a data exchange between the two entities *A* and *B*. Neither of them should be able to predetermine the value of the shared secret key.

b) Establish a shared secret key for use in a symmetric cryptographic technique between two entities *A* and *B* via key transport. In a secret key transport mechanism, the secret key is chosen by one entity *A* and is transferred to another entity *B*, suitably protected by asymmetric techniques.

c) Make an entity's public key available to other entities via key transport. In a public key transport mechanism, the public key of entity *A* shall be transferred to other entities in an authenticated way, but not requiring secrecy.

Some of the mechanisms of this part of ISO/IEC 11770 are based on the corresponding authentication mechanisms in ISO/IEC 9798-3.[6]

This part of ISO/IEC 11770 does not cover certain aspects of key management, such as

— key lifecycle management,

— mechanisms to generate or validate asymmetric key pairs, and

— mechanisms to store, archive, delete, destroy, etc. keys.

While this part of ISO/IEC 11770 does not explicitly cover the distribution of an entity's private key (of an asymmetric key pair) from a trusted third party to a requesting entity, the key transport mechanisms described can be used to achieve this. A private key can in all cases be distributed with these mechanisms where an existing, non-compromised key already exists. However, in practice the distribution of private keys is usually a manual process that relies on technological means such as smart cards, etc.

This part of ISO/IEC 11770 does not specify the transformations used in the key management mechanisms.

NOTE To provide origin authentication for key management messages, it is possible to make provisions for authenticity within the key establishment protocol or to use a public key signature system to sign the key exchange messages.

2 Normative references

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

ISO/IEC 10118 (all parts), *Information technology — Security techniques — Hash-functions*

ISO/IEC 11770-1, *Information technology — Security techniques — Key management — Part 1: Framework*

ISO/IEC 15946-1, *Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 1: General*

ISO/IEC 18031, *Information technology — Security techniques — Random bit generation*

3 Terms and definitions

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

3.1
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), and has the property that given the public transformation, then it is computationally infeasible to derive the private transformation

Note 1 to entry: A system based on asymmetric cryptographic techniques can either be an encryption system, a signature system, a combined encryption and signature system, or a key agreement scheme. With asymmetric cryptographic techniques there are four elementary transformations: *signature* and *verification* for signature systems, *encryption* and *decryption* for encryption systems. The signature and the decryption transformations are kept private by the owning entity, whereas the corresponding verification and encryption transformations are published. There exist asymmetric cryptosystems (e.g. RSA) where the four elementary functions can be achieved by only two transformations: one private transformation suffices for both signing and decrypting messages, and one public transformation suffices for both verifying and encrypting messages. However, since this does not conform to the principle of key separation, throughout this part of ISO/IEC 11770 the four elementary transformations and the corresponding keys are kept separate.

3.2
asymmetric encryption system
system based on asymmetric cryptographic techniques whose public transformation is used for encryption and whose private transformation is used for decryption

3.3
asymmetric key pair
pair of related keys where the private key defines the private transformation and the public key defines the public transformation

3.4
certification authority
CA
centre trusted to create and assign public key certificates

3.5
collision-resistant hash-function
hash-function satisfying the following property: it is computationally infeasible to find any two distinct inputs which map to the same output

[SOURCE: ISO/IEC 10118-1:2000, 3.2]

3.6
decryption
reversal of a corresponding encryption

[SOURCE: ISO/IEC 11770-1:2010, 2.6]

3.7
digital signature
data unit appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to verify the origin and integrity of the data unit and protect the sender and the recipient of the data unit against forgery by third parties, and the sender against forgery by the recipient

3.8
distinguishing identifier
information which unambiguously distinguishes an entity

[SOURCE: ISO/IEC 11770-1:2010, 2.9]

3.9
encryption
(reversible) transformation of data by a cryptographic algorithm to produce ciphertext, i.e. to hide the information content of the data

[SOURCE: ISO/IEC 11770-1:2010, 2.10]

3.10
entity authentication
corroboration that an entity is the one claimed

[SOURCE: ISO/IEC 9798-1:2010, 3.14]

3.11
entity authentication of entity *A* to entity *B*
assurance of the identity of entity *A* for entity *B*

3.12
explicit key authentication from entity *A* to entity *B*
assurance for entity *B* that entity *A* is the only other entity that is in possession of the correct key

Note 1 to entry: Implicit key authentication from entity *A* to entity *B* and key confirmation from entity *A* to entity *B* together imply explicit key authentication from entity *A* to entity *B*.

3.13
forward secrecy with respect to entity *A*
property that knowledge of entity *A*'s long-term private key subsequent to a key agreement operation does not enable an opponent to recompute previously derived keys

3.14
forward secrecy with respect to both entity *A* and entity *B* individually
property that knowledge of entity *A*'s long-term private key or knowledge of entity *B*'s long-term private key subsequent to a key agreement operation does not enable an opponent to recompute previously derived keys

Note 1 to entry: This differs from mutual forward secrecy in which knowledge of both entity *A*'s and entity *B*'s long-term private keys do not enable recomputation of previously derived keys.

3.15
hash-function
function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties:

— it is computationally infeasible to find for a given output, an input which maps to this output;

— it is computationally infeasible to find for a given input, a second input which maps to the same output

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment.

Note 2 to entry: For the purposes of this standard all hash-functions are assumed to be collision-resistant (see 3.5).

[SOURCE: ISO/IEC 10118-1:2000, 3.5]

3.16
implicit key authentication from entity *A* to entity *B*
assurance for entity *B* that entity *A* is the only other entity that can possibly be in possession of the correct key

3.17
key
sequence of symbols that controls the operation of a cryptographic transformation (e.g. encryption, decryption, cryptographic check function computation, signature calculation, or signature verification)

[SOURCE: ISO/IEC 11770-1:2010, 2.12]

3.18
key agreement
process of establishing a shared secret key between entities in such a way that neither of them can predetermine the value of that key

Note 1 to entry: By predetermine it is meant that neither entity *A* nor entity *B* can, in a computationally efficient way, choose a smaller key space and force the computed key in the protocol to fall into that key space.

3.19
key commitment

process of committing to use specific keys in the operation of a key agreement scheme before revealing the specified keys

3.20

key confirmation from entity *A* to entity *B*
assurance for entity *B* that entity *A* is in possession of the correct key

3.21
key control
ability to choose the key or the parameters used in the key computation

3.22
key derivation function
function that outputs one or more shared secrets, for use as keys, given shared secrets and other mutually known parameters as input

3.23
key establishment
process of making available a shared secret key to one or more entities, where the process includes key agreement and key transport

3.24
key token
key management message sent from one entity to another entity during the execution of a key management mechanism

3.25
key transport
process of transferring a key from one entity to another entity, suitably protected

3.26
message authentication code
MAC

string of bits which is the output of a MAC algorithm

Note 1 to entry: A MAC is sometimes called a cryptographic check value (see for example ISO 7498-2[1]).

[SOURCE: ISO/IEC 9797-1:2011, 3.9]

3.27
Message Authentication Code algorithm
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 a set of input strings and corresponding function values, where the value of the ith input string might have been chosen after observing the value of the first $i - 1$ function values (for integers $i > 1$)

Note 1 to entry: A MAC algorithm is sometimes called a cryptographic check function (see for example ISO 7498-2[1]).

Note 2 to entry: Computational feasibility depends on the user's specific security requirements and environment.

[SOURCE: ISO/IEC 9797-1:2011, 3.10]

3.28
mutual entity authentication
entity authentication which provides both entities with assurance of each other's identity

3.29
mutual forward secrecy
property that knowledge of both entity A's and entity B's long-term private keys subsequent to a key agreement operation does not enable an opponent to recompute previously derived keys

3.30
one-way function
function with the property that it is easy to compute the output for a given input but it is computationally infeasible to find an input which maps to a given output

3.31
prefix free representation
representation of a data element for which concatenation with any other data does not produce a valid representation

3.32
private key
key of an entity's asymmetric key pair that is kept private

Note 1 to entry: The security of an asymmetric system depends on the privacy of this key.

[SOURCE: ISO/IEC 11770-1:2010, 2.35]

3.33
public key
key of an entity's asymmetric key pair which can usually be made public without compromising security

Note 1 to entry: In the case of an asymmetric signature system, the public key defines the verification transformation. In the case of an asymmetric encryption system, the public key defines the encryption transformation, conditional on the inclusion of randomisation elements. A key that is "publicly known" is not necessarily globally available. The key can only be available to all members of a pre-specified group.

[SOURCE: ISO/IEC 11770-1:2010, 2.36]

3.34
public key certificate
public key information of an entity signed by the certification authority and thereby rendered unforgeable

3.35
public key information
information containing at least the entity's distinguishing identifier and public key, but can include other static information regarding the certification authority, the entity, restrictions on key usage, the validity period, or the involved algorithms

3.36
secret key
key used with symmetric cryptographic techniques by a specified set of entities

3.37
sequence number
time variant parameter whose value is taken from a specified sequence which is non-repeating within a certain time period

[SOURCE: ISO/IEC 11770-1:2010, 2.44]

3.38
signature system
system based on asymmetric cryptographic techniques whose private transformation is used for signing and whose public transformation is used for verification

3.39
third party forward secrecy
property that knowledge of a third party's private key subsequent to a key agreement operation does not enable an opponent to recompute previously derived keys

Note 1 to entry: Instead of third party forward secrecy, master key forward secrecy is also used in Reference [19].

3.40
time stamp
data item which denotes a point in time with respect to a common time reference

3.41
time-stamping authority
trusted third party trusted to provide a time-stamping service

[SOURCE: ISO/IEC 13888-1:2009, 3.58]

3.42
time variant parameter
data item used to verify that a message is not a replay, such as a random number, a time stamp or a sequence number

Note 1 to entry: If a random number is used. then this is as a challenge in a challenge-response protocol. See also ISO/IEC 9798-1:2010, Annex B.

[SOURCE: ISO/IEC 9798-1:2010, 3.36]

3.43
trusted third party
security authority or its agent, trusted by other entities with respect to security related activities

[SOURCE: ISO/IEC 9798-1:2010, 3.38]

4 Symbols and abbreviations

The following symbols and abbreviations are used in this part of ISO/IEC 11770.

A, B, C	distinguishing identifiers of entities
BE	encrypted data block
BS	signed data block
CA	certification authority
$Cert_A$	entity A's public key certificate
D_A	entity A's private decryption transformation function
d_A	entity A's private decryption key
E	elliptic curve, either given by an equation of the form $Y^2 = X^3 + aX + b$ over the field GF(p^m) for p>3 and a positive integer m, by an equation of the form $Y^2 + XY = X^3 + aX^2 + b$ over the field GF(2^m), or by an equation of the form $Y^2 = X^3 + aX^2 + b$ over the field GF(3^m), together with an extra point O_E referred to as the point at infinity, which is denoted by E/GF(p^m), E/GF(2^m), or E/GF(3^m), respectively
E_A	entity A's public encryption transformation function
e_A	entity A's public encryption key
F	key agreement function
F(h,g)	key agreement function using as input a factor h and a common element g
FP	key agreement function based on pairing
G	point on E with order n
g	common element shared publicly by all the entities that use the key agreement function F
gcd(a,b)	greatest common divisor of two integers a and b
GF(p^m), GF(2^m), GF(3^m)	finite field with p^m , 2^m, 3^m elements for a prime p>3 and a positive integer m
h_A	entity A's private key agreement key
hash	hash-function
j	cofactor used in performing cofactor multiplication
K	secret key for a symmetric cryptosystem
K_{AB}	secret key shared between entities A and B

NOTE 1 In practical implementations the shared secret key should be subject to further processing before it can be used for a symmetric cryptosystem.

kdf	key derivation function
KT	key token
KT_A	entity A's key token

KT_{Ai}	key token sent by entity A after processing phase i
l	supplementary value used in performing cofactor multiplication
M	data message
MAC	Message Authentication Code
$\mathrm{MAC}_K(Z)$	output of a MAC algorithm when using as input the secret key K and an arbitrary data string Z
MQV	Menezes-Qu-Vanstone
n	prime divisor of the order (or cardinality) of an elliptic curve E over a finite field
O_E	elliptic curve point at infinity
P	point on an elliptic curve E
p_A	entity A's public key-agreement key
pairing	pairing defined over an elliptic curve and used in FP
parameters	parameters used in the key derivation function
PKI_A	entity A's public key information
P_X	public key-agreement key in an elliptic curve of entity X
q	prime power p^m for some prime $p \neq 3$ and some integer $m \geq 1$
r	random number generated in the course of a mechanism
r_A	random number issued by entity A in a key agreement mechanism
S_1, S_2, S_3	sets of elements
S_A	entity A's private signature transformation function
s_A	entity A's private signature key
T	trusted third party
Text$i$	i th optional text, data or other information that may be included in a data block, if desired
TVP	time-variant parameter such as a random number, a time stamp, or a sequence number
V_A	entity A's public verification transformation function
v_A	entity A's public verification key
w	one-way function
$\mathrm{X}(P)$	x-coordinate of a point P
$\sqrt{q}$	square root of a positive number q
$\#E$	order (or cardinality) of an elliptic curve E

$\|\|$	concatenation of two data elements
$\lceil x \rceil$	smallest integer greater than or equal to the real number x
$\sum$	digital signature
$\pi(P)$	$(\mathrm{X}(P) \bmod 2^{\lceil \rho/2 \rceil}) + 2^{\lceil \rho/2 \rceil}$ where $\rho = \lceil \log_2 n \rceil$ and X(P) is the x-coordinate of the point P

NOTE 2 No assumption is made on the nature of the signature transformation. In the case of a signature system with message recovery, $S_A(M)$ denotes the signature $\sum$ itself. In the case of a signature system with appendix, $S_A(M)$ denotes the message M together with the signature $\sum$.

NOTE 3 The keys of an asymmetric cryptosystem are denoted by lower case letters (indicating its function) indexed with the identifier of its owner, e.g., the public verification key of entity A is denoted by v_A. The corresponding transformations are denoted by upper case letters indexed with the identifier of their owner, e.g., the public verification transformation of entity A is denoted by V_A.

5 Requirements

It is assumed that the entities involved in a mechanism are aware of each other's claimed identities. This may be achieved by the inclusion of identifiers in information exchanged between the two entities, or it may be apparent from the context of use of the mechanism. Verifying the identity means checking that a received identifier field agrees with some known (trusted) or expected value.

If a public key is registered with an entity, then that entity shall make sure that the entity who registers the key is in possession of the corresponding private key (see ISO/IEC 11770-1 for further guidance on key registration).

6 Key derivation functions

The use of a shared secret as derived in Clause 10 as a key for a symmetric cryptosystem without further processing is not recommended. It will often be the case that the form of a shared secret established as a result of using a mechanism specified in this part of ISO/IEC 11770 will not conform to the form needed for a specific cryptographic algorithm, so some processing will be needed. Moreover, the shared secret (often) has arithmetic properties and relationships that might result in a shared symmetric key not being chosen from the full key space. It is therefore advisable to pass the shared secret through a key derivation function, e.g. involving the use of a hash function. The use of an inadequate key derivation function could compromise the security of the key agreement scheme with which it is used. It is recommended to use a one-way function as a key derivation function.

A key derivation function produces keys that are computationally indistinguishable from randomly generated keys. The key derivation function takes as input a shared secret and a set of key derivation parameters and produces an output of the desired length.

In order for the two parties in a key establishment mechanism to agree on a common secret key, the key derivation function shall be agreed upon (see ISO/IEC 11770-6 for further guidance on key derivation functions).

Annex C provides examples of key derivation functions.

7 Cofactor multiplication

This clause applies only to mechanisms using elliptic curve cryptography. The key agreement mechanisms in Clause 11 and the key transport mechanisms in Clauses 12 and 13 require that the user's private key or key token be combined with another entity's public key or key token. If the other entity's

