INTERNATIONAL STANDARD

ISO/IEC 11770-3

Second edition
2008-07-15

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 11770-3:2008(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 ... iv
Introduction ... v
1 Scope ... 1
2 Normative references ... 2
3 Terms and definitions ... 2
4 Symbols and abbreviations ... 6
5 Requirements ... 8
6 Key derivation functions ... 8
7 Cofactor multiplication ... 9
8 Key commitment ... 10
9 Key confirmation ... 10
10 Secret key agreement ... 12
10.1 Key agreement mechanism 1 ... 12
10.2 Key agreement mechanism 2 ... 13
10.3 Key agreement mechanism 3 ... 14
10.4 Key agreement mechanism 4 ... 16
10.5 Key agreement mechanism 5 ... 17
10.6 Key agreement mechanism 6 ... 18
10.7 Key agreement mechanism 7 ... 20
10.8 Key agreement mechanism 8 ... 22
10.9 Key agreement mechanism 9 ... 23
10.10 Key agreement mechanism 10 ... 24
10.11 Key agreement mechanism 11 ... 25
11 Secret key transport ... 27
11.1 Key transport mechanism 1 ... 27
11.2 Key transport mechanism 2 ... 28
11.3 Key transport mechanism 3 ... 30
11.4 Key transport mechanism 4 ... 31
11.5 Key transport mechanism 5 ... 33
11.6 Key transport mechanism 6 ... 35
12 Public key transport ... 37
12.1 Public key distribution without a trusted third party ... 37
12.2 Public key distribution using a trusted third party ... 39
Annex A (informative) Properties of key establishment mechanisms ... 41
Annex B (informative) Examples of key derivation functions ... 43
Annex C (informative) Examples of key establishment mechanisms ... 51
Annex D (informative) Examples of elliptic curve based key establishment mechanisms ... 57
Annex E (informative) Key transport ... 69
Annex F (normative) ASN.1 module ... 74
Bibliography ... 82

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 11770-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

This second edition cancels and replaces the first edition (ISO/IEC 11770-3:1999), and ISO/IEC 15946-3:2002, which have been merged and updated to present a uniform standard on key management.

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*

Introduction

This part of ISO/IEC 11770 defines 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 cryptosystems is based on the presumed intractability of solving the discrete logarithm problem over a finite field. Other cryptosystems such as RSA are based on the difficulty of the integer factorization problem.

The second form of cryptography discussed in this part of ISO/IEC 11770 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. This problem is, with current knowledge, much harder than the factorization of integers or the computation of discrete logarithms in a finite field. Indeed, since V. Miller and N. Koblitz in 1985 independently suggested the use of elliptic curves for public key cryptographic systems, no substantial progress in tackling the elliptic curve discrete logarithm problem has been reported. In general, only algorithms that take exponential time are known to determine elliptic curve discrete logarithms. 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 and system parameters and allows for computations using smaller integers.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document 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:

ISO/IEC JTC 1/SC 27 Standing Document 8 (SD 8).

SD 8 is publicly available at http://www.jtc1sc27.din.de/sce/sd8.

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.

INTERNATIONAL STANDARD **ISO/IEC 11770-3:2008(E)**

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.

1) Establish a shared secret key for a symmetric cryptographic technique between two entities *A* and *B* by key agreement. In a secret key agreement mechanism, the secret key is the result of a data exchange between the two entities *A* and *B*. Neither of them can predetermine the value of the shared secret key.

2) Establish a shared secret key for a symmetric cryptographic technique between two entities *A* and *B* by 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.

3) Make an entity's public key available to other entities by key transport. In a public key transport mechanism, the public key of entity *A* must 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.

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

— key lifecycle management,

— mechanisms to generate or validate asymmetric key pairs,

— 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 like smart cards, etc.

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

NOTE To achieve authenticity of 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 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 10118 (all parts), *Information technology — Security techniques — Hash-functions*

ISO/IEC 14888 (all parts), *Information technology — Security techniques — Digital signatures with appendix*

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, it is computationally infeasible to derive the private transformation

NOTE A system based on asymmetric cryptographic techniques can either be an encipherment system, a signature system, a combined encipherment and signature system, or a key agreement system. With asymmetric cryptographic techniques there are four elementary transformations: sign and verify for signature systems, encipher and decipher for encipherment systems. The signature and the decipherment transformation are kept private by the owning entity, whereas the corresponding verification and encipherment 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 encipherment system
system based on asymmetric cryptographic techniques whose public transformation is used for encipherment and whose private transformation is used for decipherment

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
decipherment
reversal of a corresponding encipherment

[ISO/IEC 11770-1:1996]

3.6
digital signature
data unit appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove 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

3.7
distinguishing identifier
information which unambiguously distinguishes an entity

[ISO/IEC 11770-1:1996]

3.8
encipherment
(reversible) transformation of data by a cryptographic algorithm to produce ciphertext, i.e. to hide the information content of the data

[ISO/IEC 9797-1:1999, ISO/IEC 11770-1:1996, ISO/IEC 18033-1:2005]

3.9
entity authentication
corroboration that an entity is the one claimed

[ISO/IEC 9798-1:1997]

3.10
entity authentication of entity *A* to entity *B*
assurance of the identity of entity *A* for entity *B*

3.11
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 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*.

3.12
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

3.13
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 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.

3.14
hash-function
function which maps strings of bits to fixed-length strings of bits, satisfying two properties:

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

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

NOTE 1 The literature on this subject contains a variety of terms which have the same or similar meaning as hash-function. Compressed encoding and condensing function are some examples.

NOTE 2 Computational feasibility depends on the user's specific security requirements and environment.

3.15
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

3.16
key
sequence of symbols that controls the operation of a cryptographic transformation (e.g. encipherment, decipherment, *MAC* function computation, signature calculation, or signature verification)

[ISO/IEC 11770-1:1996]

3.17
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 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.

3.18
key commitment
process of committing to use specific keys in the operation of a key agreement scheme before revealing the specified keys

3.19
key confirmation from entity *A* to entity *B*
assurance for entity *B* that entity *A* is in possession of the correct key

3.20
key control
ability to choose the key or the parameters used in the key computation

3.21
key derivation function
function that outputs one or more shared secrets, used as keys, given shared secrets and other mutually known parameters as input

3.22
key establishment
process of making available a shared secret key to one or more entities, where the process includes key agreement and key transport

3.23
key token
key management message sent from one entity to another entity during the execution of a key management mechanism

3.24
key transport
process of transferring a key from one entity to another entity, suitably protected

3.25
message authentication code
MAC
string of bits which is the output of a *MAC* algorithm

NOTE A *MAC* is sometimes called a cryptographic check value (see, for example, ISO 7498-2).

3.26
message authentication code (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:

1) for any key and any input string the function can be computed efficiently;

2) 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 *i* th input string may have been chosen after observing the value of the first *i* – 1 function values

3.27
mutual entity authentication
entity authentication which provides both entities with assurance of each other's identity

3.28
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

3.29
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 for a given output an input which maps to this output

3.30
prefix free representation
representation of a data element for which concatenation with any other data does not produce a valid representation

3.31
private key
key of an entity's asymmetric key pair which can only be used by that entity

NOTE In the case of an asymmetric signature system, the private key defines the signature transformation. In the case of an asymmetric encipherment system, the private key defines the decipherment transformation.

3.32
public key
key of an entity's asymmetric key pair which can be made public

NOTE In the case of an asymmetric signature system, the public key defines the verification transformation. In the case of an asymmetric encipherment system, the public key defines the encipherment transformation. A key that is "publicly known" is not necessarily globally available. The key can be available only to all members of a pre-specified group.

3.33
public key certificate
public key information of an entity signed by the certification authority and thereby rendered unforgeable

3.34
public key information
information containing at least the entity's distinguishing identifier and public key, but which can include other static information regarding the certification authority, the entity, restrictions on key usage, the validity period, or the involved algorithms

3.35
secret key
key used with symmetric cryptographic techniques by a specified set of entities

3.36
sequence number
time variant parameter whose value is taken from a specified sequence which is non-repeating within a certain time period

[ISO/IEC 11770-1:1996]

3.37
signature system
system based on asymmetric cryptographic techniques whose private transformation is used for signing and whose public transformation is used for verification

3.38
time stamp
data item which denotes a point in time with respect to a common time reference

3.39
time stamping authority
trusted third party trusted to provide evidence which includes the time when the secure time stamp is generated

[ISO/IEC 13888-1:2004]

3.40
time variant parameter
TVP
data item used to verify that a message is not a replay, such as a random number, a sequence number, or a time stamp

NOTE If time stamps are used, secure and synchronized time clocks are required. If sequence numbers are used, the ability to maintain and verify bilateral counters is required.

3.41
trusted third party
security authority, or its agent, trusted by other entities with respect to security related activities

[ISO/IEC 10181-1:1996]

4 Symbols and abbreviations

A, *B*	distinguishing identifiers of entities
BE	enciphered data block
BS	signed data block
CA	certification authority
$Cert_A$	entity *A*'s public key certificate
D_A	entity *A*'s private decipherment transformation
d_A	entity *A*'s private decipherment key
E	An elliptic curve, either given by an equation of the form $Y^2 = X^3 + aX + b$ over the field $F(p^m)$ for $p>3$, by an equation of the form $Y^2 + XY = X^3 + aX^2 + b$ over the field $F(2^m)$, or by an equation of the form $Y^2 = X^3 + aX^2 + b$ over the field $F(3^m)$, together with an extra point O_E referred to as the point at infinity. The elliptic curve is denoted by $E/F(p^m)$, $E/F(2^m)$, or $E/F(3^m)$, respectively.
E_A	entity *A*'s public encipherment transformation
e_A	entity *A*'s public encipherment key
F	the key agreement function

$F(h,g)$	the key agreement function using as input a cofactor h and a common element g
G	a point on E with order n
g	the common element shared publicly by all the entities that use the key agreement function F
H	set of elements
hash	hash-function
h_A	entity A's private key agreement key
j	the cofactor used in performing cofactor multiplication
K	a secret key for a symmetric cryptosystem
KT	key token
KT_A	entity A's key token
KT_{Ai}	the key token sent by entity A after processing phase i
K_{AB}	a secret key shared between entities A and B
	NOTE In practical implementations, the shared secret key should be subject to further processing before it can be used for a symmetric cryptosystem.
kdf	a key derivation function
l	a supplementary value used in performing cofactor multiplication
M	a data message
MAC	Message Authentication Code
$MAC_K(Z)$	The output of a *MAC* algorithm when using as input the secret key K and an arbitrary data string Z
MQV	Menezes-Qu-Vanstone
n	a prime divisor of the order (or cardinality) of an elliptic curve E over a finite field
O_E	the elliptic curve point at infinity
P	a point on an elliptic curve E
P_X	the x-coordinate of a point P
PKI_A	entity A's public key information
parameters	parameters used in the key derivation function
p_A	entity A's public key agreement key
q	a prime power p^m for some prime $p \neq 3$ and some integer $m \geq 1$
r	a random number generated in the course of a mechanism
r_A	a random number issued by entity A in a key agreement mechanism

S	set of elements
S_A	entity *A*'s private signature transformation
s_A	entity *A*'s private signature key
TVP	time-variant parameter such as a random number, a time stamp, or a sequence number
Texti	*i* th optional text, data or other information that may be included in a data block, if desired
V_A	entity *A*'s public verification transformation
v_A	entity *A*'s public verification key
w	one-way function
$\#E$	The order (or cardinality) of an elliptic curve *E*
$\|\|$	concatenation of two data elements
$\lceil x \rceil$	the smallest integer greater than or equal to the real number *x*
Σ	the digital signature
$\pi(P)$	$(P_X \bmod 2^{\lceil \rho/2 \rceil}) + 2^{\lceil \rho/2 \rceil}$ where $\rho = \lceil \log_2 n \rceil$ and P_X is the x-coordinate of the point *P*

NOTE 1 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 Σ itself. In the case of a signature system with appendix, $S_A(m)$ denotes the message *m* together with the signature Σ.

NOTE 2 The keys of an asymmetric cryptosystem are denoted by a lower case letter (indicating the function of that key) 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 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 the use of the mechanism. Verifying the identity means to check that a received identifier field agrees with some known (trusted) value or prior expectation.

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 most often will be the case that the form of a shared secret will not conform to the form needed for a shared symmetric key, so some processing will be needed. 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 advisable to pass the shared secret through a key derivation function, which includes the use of a hash function. The use of an inadequate key derivation function compromises the security of the key agreement scheme in which it is used.

A key derivation function produces keys that are computationally indistinguishable from randomly generated keys. The key derivation function takes as input the shared secret and a set of key derivation parameters and produce 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 must be agreed upon. The method of coming to such an agreement is outside the scope of this part of ISO/IEC 11770.

See Annex B for examples of key derivation functions.

7 Cofactor multiplication

This clause applies only to elliptic curve cryptography. The key agreement mechanisms in Clause 10 and the key transport mechanisms in Clauses 11 and 12 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 public key or key token is not valid (i.e. it is not a point on the elliptic curve, or is not in the subgroup of order *n*), then performing this operation may result in some bits of the private key being leaked to an attacker. An example of this attack is the 'small subgroup attack'.

In order to prevent the 'small subgroup attack' and similar attacks, one option is to validate public keys and key tokens received from the other party using public key validation. See ISO/IEC 11770-1 for a description of public key validation.

As an alternative to verifying the order of the public key or key token, a technique called cofactor multiplication can be used. The values j and l, defined below, are used for cofactor multiplication in Clause 10.

If cofactor multiplication is desired, there are two options:

1) If cofactor multiplication is used, and incompatibility with those not using cofactor multiplication is desired, then let $j = \#E / n$ and $l = 1$. If this option is chosen, both parties involved must agree to use this option, otherwise the mechanism will not work.

2) If cofactor multiplication is used, and compatibility with those not using cofactor multiplication is desired, then let $j = \#E / n$ and $l = j^{-1}$ mod n.

NOTE The value j^{-1} mod n will always exist since n is required to be greater than $4\sqrt{q}$ and therefore $gcd(n,j) = 1$.

If cofactor multiplication is not desired, there is one option:

1) If cofactor multiplication is not used, then let $j = 1$ and $l = 1$.

Regardless of whether or not cofactor multiplication is used and the type of compatibility that is chosen, if the shared key (or a component of the shared key) evaluates to the point at infinity (O_E), then the user shall assume that the key agreement has failed.

It should be noted that it is most appropriate to perform these operations (public key validation or cofactor multiplication) if the other entity's public key or key token is not authenticated or the user's public key is long term. Performing public key validation for long-term keys and cofactor multiplication for ephemeral (short term) keys may also have performance advantages.

It should also be noted that if the other entity's public key is authenticated and the cofactor is small, then the amount of information that can be leaked is limited. Thus, it may not always be desired that these tests be performed.