FINAL DRAFT

INTERNATIONAL STANDARD

ISO/IEC FDIS 11770-3

ISO/IEC JTC **1**/SC **27**

Secretariat: DIN

Voting begins on:
2021-06-14

Voting terminates on:
2021-08-09

Information security — Key management —

Part 3: **Mechanisms using asymmetric techniques**

Sécurité de l'information — Gestion de clés —

Partie 3: Mécanismes utilisant des techniques asymétriques

Reference number
ISO/IEC FDIS 11770-3:2021(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.

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 or www.iec.ch/members_experts/refdocs).

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) or the IEC list of patent declarations received (see patents.iec.ch).

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *Information security, cybersecurity and privacy protection*.

This fourth edition cancels and replaces the third edition (ISO/IEC 11770-3:2015), which has been technically revised. It aso incorporates Technical Corrigenda ISO/IEC 11770-3:2015/Cor1:2016 and ISO/IEC 11770-3:2015/Amd.1:2017).

The main changes compared to the previous edition are as follows:

— the blinded Diffie-Hellman key agreements are added as key agreement mechanism 13 and 14 and examples of the mechanisms are included in Annex E;

— key agreement mechanism 15 is added and the SM9 key agreement protocol as an example of the mechanism is included in Annex F.

A list of all parts in the ISO/IEC 11770 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document 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 document 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 document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she 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 ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents and http://patents.iec.ch.

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

Information security — Key management —

Part 3:
Mechanisms using asymmetric techniques

1 Scope

This document 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 is 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* is transferred to other entities in an authenticated way, but not requiring secrecy.

Some of the mechanisms of this document are based on the corresponding authentication mechanisms in ISO/IEC 9798-3.

This document 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 document 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 document 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 documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 10118-1, *Information technology — Security techniques — Hash-functions — Part 1: General*

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

ISO/IEC 11770-6, *Information technology — Security techniques — Key management — Part 6: Key derivation*

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*

ISO/IEC 19772, *Information security — Authenticated encryption*

3 Terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
asymmetric cryptographic technique
cryptographic technique that uses two related transformations, a public transformation [defined by the *public key* (3.33)] and a private transformation [defined by the *private key* (3.32)], 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 document the four elementary transformations and the corresponding keys are kept separate.

3.2
asymmetric encryption system
system based on *asymmetric cryptographic techniques* (3.1) whose public transformation is used for *encryption* (3.9) and whose private transformation is used for *decryption* (3.6)

3.3
asymmetric key pair
pair of related *keys* (3.17) where the *private key* (3.32) defines the private transformation and the *public key* (3.33) defines the public transformation

3.4
certification authority
CA
centre trusted to create and assign *public key* (3.33) certificates

3.5
collision-resistant hash-function
hash-function (3.15) satisfying the following property: it is computationally infeasible to find any two distinct inputs which map to the same output

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

[SOURCE: ISO/IEC 10118-1:2016, 3.1]

3.6
decryption
reversal of a corresponding *encryption* (3.9)

[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* (3.17)

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 both entity *A* and entity *B* individually
property that knowledge of entity *A*'s long-term *private key* (3.32) or knowledge of entity *B*'s long-term *private key* (3.32) subsequent to a *key agreement* (3.18) operation does not enable an opponent to recompute previously derived *keys* (3.17)

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.

3.14
forward secrecy with respect to entity *A*
property that knowledge of entity *A*'s long-term *private key* (3.32) subsequent to a *key agreement* (3.18) operation does not enable an opponent to recompute previously derived *keys* (3.17)

3.15
hash-function
function which maps strings of bits of variable (but usually upper bounded) length to fixed-length strings of bits, satisfying the following two properties:

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

— for a given input, it is computationally infeasible to find 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 document all hash-functions are assumed to be collision-resistant hash-functions.

[SOURCE: ISO/IEC 10118-1:2016, 3.4]

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)

3.17
key
sequence of symbols that controls the operation of a cryptographic transformation (e.g. *encryption* (3.9), *decryption* (3.6), 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* (3.38) between entities in such a way that neither of them can predetermine the value of that *key* (3.17)

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* (3.17) in the operation of a *key agreement* (3.18) scheme before revealing the specified *keys* (3.17)

3.20
key confirmation from entity *A* to entity *B*
assurance for entity *B* that entity *A* is in possession of the correct *key* (3.17)

3.21
key control
ability to choose the *key* (3.17) or the parameters used in the *key* (3.17) computation

3.22
key derivation function
function that outputs one or more shared secrets, for use as *keys* (3.17), given shared secrets and other mutually known parameters as input

3.23
key establishment
process of making available a shared *secret key* (3.38) to one or more entities, where the process includes *key agreement* (3.18) and *key transport* (3.25)

3.24
key token
key (3.17) management message sent from one entity to another entity during the execution of a *key* (3.17) management mechanism

3.25
key transport
process of transferring a *key* (3.17) 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* (3.27)

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

[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* (3.38) to fixed-length strings of bits, satisfying the following two properties:

— for any *key* (3.17) and any input string, the function can be computed efficiently;

— for any fixed *key* (3.17), and given no prior knowledge of the *key* (3.17), 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 can 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).

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 (3.10) 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* (3.32) subsequent to a *key agreement* (3.18) operation does not enable an opponent to recompute previously derived *keys* (3.17)

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 (3.17) of an entity's *asymmetric key pair* (3.3) 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 (3.17) of an entity's *asymmetric key pair* (3.3) 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 (3.35) of an entity signed by the *certification authority* (3.4) and thereby rendered unforgeable

3.35
public key information
information containing at least the entity's *distinguishing identifier* (3.8) and *public key* (3.33), but can include other static information regarding the *certification authority* (3.4), the entity, restrictions on *key* (3.17) usage, the validity period, or the involved algorithms

3.36
resilience to key compromise impersonation attack on *A*
resilience to attacks in which an adversary exploits knowledge of the long-term *private key* (3.32) of *A* to impersonate any entity in subsequent communication with *A*

3.37
resilience to unknown key share attack for *A* and *B*
resilience to attacks in which only *A* and *B* know the session *key* (3.17) *K*; however, *A* and *B* disagree on who they share *K* with

Note 1 to entry: Resilience to unknown key share attack can be achieved by choosing a key derivation function that includes the identifiers of the involved entities.

3.38
secret key
key (3.17) used with symmetric cryptographic techniques by a specified set of entities

3.39
sequence number
time variant parameter (3.44) 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]

3.40
signature system
system based on *asymmetric cryptographic techniques* (3.1) whose private transformation is used for signing and whose public transformation is used for verification

3.41
third party forward secrecy
property that knowledge of a third party's *private key* (3.32) subsequent to a *key agreement* (3.18) operation does not enable an opponent to recompute previously derived *keys* (3.17)

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

3.42
time stamp
data item which denotes a point in time with respect to a common time reference

3.43
time-stamping authority
trusted third party (3.45) trusted to provide a time-stamping service

[SOURCE: ISO/IEC 13888-1:2020, 3.54]

3.44
time variant parameter
data item used to verify that a message is not a replay, such as a random number, a *time stamp* (3.42) or a *sequence number* (3.39)

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]

3.45
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 document.

A, B, C	distinguishing identifiers of entities
BE	encrypted data block
BS	signed data block
BS2IP	bit string to integer conversion primitive
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
IHF2(A, n)	BS2IP(kdf(A, $8\lceil(5\ b_n)/32\rceil$)) mod (n-1) +1 where b_n is the bit-length of n
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 is 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
$\text{MAC}_K(Z)$	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
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
π(*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, $\mathrm{S}_A(M)$ denotes the signature $\sum$ itself. In the case of a signature system with appendix, $\mathrm{S}_A(M)$ 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 can be achieved by the inclusion of identifiers in information exchanged between the two entities, or it can 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).

Annex A lists the object identifiers which shall be used to identify the key management mechanisms specified in this document.

