INTERNATIONAL STANDARD

ISO/IEC 11770-4

Second edition
2017-11

Information technology — Security techniques — Key management —

Part 4: Mechanisms based on weak secrets

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

Partie 4: Mécanismes basés sur des secrets faibles

Reference number
ISO/IEC 11770-4:2017(E)

Contents

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 ISO documents 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 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 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 the following URL: www.iso.org/iso/foreword.html.

This document was prepared by ISO/IEC JTC 1, *Information technology*, SC 27, *IT Security techniques*.

This second edition cancels and replaces the first edition (ISO/IEC 11770-4:2006), which has been technically revised. It also incorporates the Technical Corrigendum ISO/IEC 11770-4:2006/Cor1:2009.

This edition includes the following significant changes with respect to the previous edition:

— revision of the Balanced Key Agreement Mechanism 1 (BKAM1) to address the attacks reported in Reference [6];

— addition of a new Balanced Key Agreement Mechanism 2 (BKAM2) based on the J-PAKE scheme of Reference [5];

— addition of a new Augmented Key Agreement Mechanism 3 (AKAM3) based on the AugPAKE scheme of Reference [23].

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

Introduction

The mechanisms specified in this document are designed to achieve one of the following three goals.

a) **Balanced password-authenticated key agreement:** Establish one or more shared secret keys between two entities that share a common weak secret. In a balanced password-authenticated key agreement mechanism, the shared secret keys are the result of a data exchange between the two entities; the shared secret keys are established if, and only if, the two entities have used the same weak secret; and neither of the two entities can predetermine the values of the shared secret keys.

b) **Augmented password-authenticated key agreement:** Establish one or more shared secret keys between two entities *A* and *B*, where *A* has a weak secret and *B* has verification data derived from a one-way function of *A*'s weak secret. In an augmented password-authenticated key agreement mechanism, the shared secret keys are the result of a data exchange between the two entities; the shared secret keys are established if, and only if, the two entities have used the weak secret and the corresponding verification data; and neither of the two entities can predetermine the values of the shared secret keys.

   NOTE 1 This type of key agreement mechanism is unable to protect *A*'s weak secret being discovered by *B*, but only increases the cost for an adversary to get *A*'s weak secret from *B*. A typical application scenario would involve use between a client (*A*) and a server (*B*).

c) **Password-authenticated key retrieval:** Establish one or more secret keys for an entity, *A*, associated with another entity, *B*, where *A* has a weak secret and *B* has a strong secret associated with *A*'s weak secret. In an authenticated key retrieval mechanism, the secret keys, retrievable by *A* (not necessarily derivable by *B*), are the result of a data exchange between the two entities, and the secret keys are established if, and only if, the two entities have used the weak secret and the associated strong secret. However, although *B*'s strong secret is associated with *A*'s weak secret, the strong secret does not (in itself) contain sufficient information to permit either the weak secret or the secret keys established in the mechanism to be determined.

   NOTE 2 This type of key retrieval mechanism is used in those applications where *A* does not have secure storage for a strong secret, and requires *B*'s assistance to retrieve the strong secret. Such a mechanism is appropriate for use between a client (*A*) and a server (*B*).

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:

National Institute of Advanced Industrial Science and Technology

1–1–1 Umezono

Tsukuba, Ibaraki

305–8560 Japan

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 online databases of patents relevant to their documents. Users are encouraged to consult the databases for the most up to date information concerning patents.

Information technology — Security techniques — Key management —

Part 4: Mechanisms based on weak secrets

1 Scope

This document defines key establishment mechanisms based on weak secrets, i.e. secrets that can be readily memorized by a human, and hence, secrets that will be chosen from a relatively small set of possibilities. It specifies cryptographic techniques specifically designed to establish one or more secret keys based on a weak secret derived from a memorized password, while preventing offline brute-force attacks associated with the weak secret. This document is not applicable to the following aspects of key management:

— life-cycle management of weak secrets, strong secrets, and established secret keys;

— mechanisms to store, archive, delete, destroy, etc. weak secrets, strong secrets, and established secret keys.

2 Normative reference

There are no normative references in this document.

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:

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

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

3.1
augmented password-authenticated key agreement
password-authenticated key agreement where entity *A* uses a password-based weak secret and entity *B* uses verification data derived from a one-way function of *A*'s weak secret to negotiate and authenticate one or more shared secret keys

3.2
balanced password-authenticated key agreement
password-authenticated key agreement where two entities *A* and *B* use a shared common password-based weak secret to negotiate and authenticate one or more shared secret keys

3.3
brute-force attack
attack on a cryptosystem that employs an exhaustive search of a set of keys, passwords or other data

3.4
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

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment. Refer to ISO/IEC 10118-1:2016, Annex C.

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

3.5
dictionary attack
(on a password-based system) attack on a cryptosystem that employs a search of a given list of passwords

Note 1 to entry: A dictionary attack on a password-based system can use a stored list of specific password values or a stored list of words from a natural language dictionary.

3.6
domain parameter
data item which is common to and known by or accessible to all entities within the domain

Note 1 to entry: The set of domain parameters may contain data items such as hash-function identifier, length of the hash-token, length of the recoverable part of the message, finite field parameters, elliptic curve parameters, or other parameters specifying the security policy in the domain.

[SOURCE: ISO/IEC 9796-3:2006, 3.2]

3.7
elliptic curve
cubic curve E without a singular point

Note 1 to entry: The set of points E together with an appropriately defined operation for a field that includes all coefficients of the equation describing E is called the definition field of E. In ISO/IEC 15946-1, only finite fields F are dealt with as the definition field. When it is necessary to describe the definition field F of E explicitly, the curve is denoted as E/F.

Note 2 to entry: The form of a cubic curve equation used to define an elliptic curve varies depending on the field. The general form of an appropriate cubic equation for all possible finite fields is defined in ISO/IEC 15946-1:2016, 6.1.

[SOURCE: ISO/IEC 15946-1:2016, 3.3, modified]

3.8
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.

[SOURCE: ISO/IEC 11770-3:2015, 3.12, modified]

3.9
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. Refer to ISO/IEC 10118-1:2016, Annex C.

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

3.10
hashed password
result of applying a hash-function to a password

3.11
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

[SOURCE: ISO/IEC 11770-3:2015, 3.16, modified]

3.12
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 11770-3:2015, 3.17]

3.13
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.

[SOURCE: ISO/IEC 11770-3:2015, 3.18]

3.14
key confirmation
<from entity *A* to entity *B*> assurance for entity *B* that entity *A* is in possession of the correct key

[SOURCE: ISO/IEC 11770-3:2015, 3.20, modified]

3.15
key control
ability to choose the key or the parameters used in the key computation

[SOURCE: ISO/IEC 11770-3:2015, 3.21]

3.16
key derivation function
function which takes as input a number of parameters, at least one of which shall be secret, and which gives as output keys appropriate for the intended algorithm(s) and applications

3.17
key establishment
process of making available a shared secret key to one or more entities

Note 1 to entry: Key establishment includes key agreement, key transport and key retrieval.

3.18
key management
administration and use of generation, registration, certification, deregistration, distribution, installation, storage, archiving, revocation, derivation and destruction of keying material in accordance with a security policy

[SOURCE: ISO/IEC 11770-1:2010, 2.28]

3.19
key retrieval
process of establishing a key for one or more entities known as the retrieving entities with the involvement of one or more other entities who are not necessarily able to access the key after the process, and which normally requires authentication of the retrieving entity/entities by the other entity/entities

3.20
key token
key establishment message sent from one entity to another entity during the execution of a key establishment mechanism

3.21
key token check function
function that utilizes a key token and other publicly known parameters as input and outputs a Boolean value during the execution of a key establishment mechanism

3.22
key token factor
value that is kept secret and that is used, possibly in conjunction with a weak secret, to create a key token

3.23
key token generation function
function that utilizes a key token factor and other parameters as input and outputs a key token during the execution of a key establishment mechanism

3.24
message authentication code 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 *i*th input string may have been chosen after observing the value of the first *i*-1 function values (for integers *i* > 1)

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

3.25
mutual key authentication
assurance for two entities that only the other entity is in possession of the correct key

3.26
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

[SOURCE: ISO/IEC 11770-3:2015, 3.30]

3.27
password
secret word, phrase, number, or character sequence used for entity authentication, which is a memorized weak secret

3.28
password-authenticated key agreement
process of establishing one or more shared secret keys between two entities using prior shared password-based information (which means that either both of them have the same shared password or one has the password and the other has password verification data) and neither of them can predetermine the values of the shared secret keys

3.29
password-authenticated key retrieval
key retrieval process where one entity *A* has a weak secret derived from a password and the other entity *B* has a strong secret associated with *A*'s weak secret; these two entities, using their own secrets, negotiate a secret key which is retrievable by *A*, but not (necessarily) derivable by *B*

3.30
password-entangled key token
key token which is derived from both a weak secret and a key token factor

3.31
password verification data
data that is used to verify an entity's knowledge of a specific password

3.32
random element derivation function
function that utilizes a password and other parameters as input and outputs a random element

3.33
salt
random variable incorporated as secondary input to a one-way or encryption function that is used to derive password verification data

3.34
secret
value known only to authorized entities

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

[SOURCE: ISO/IEC 11770-3:2015, 3.36]

3.36
secret value derivation function
function that utilizes a key token factor, a key token, and other parameters as input and outputs a secret value which is used to compute one or more secret keys

3.37
strong secret
secret with a sufficient degree of entropy that conducting an exhaustive search for the secret is infeasible, even given knowledge that would enable a correct guess for the secret to be distinguished from an incorrect guess

Note 1 to entry: This may, for example, be achieved by randomly choosing the secret from a sufficiently large set of possible values under uniform distribution.

3.38
weak secret
secret that can be conveniently memorized by a human being

Note 1 to entry: Typically, this means that the entropy of the secret is limited, so that an exhaustive search for the secret (or, a dictionary attack) may be feasible, given knowledge that would enable a correct guess for the secret to be distinguished from an incorrect guess.

4 Symbols and abbreviated terms

A, B	distinguishing identities of entities represented as octet strings
a_1, a_2	elliptic curve coefficients
BS2I	a function that converts a bit string into an integer (described in Annex A)
b, b_i	bits (i.e. either 0 or 1)
C, C_{DL}, C_{EC}	functions for generating a key token based on a password and a key token factor
c	an integer satisfying $1 \le c \le q - 1$
D, D_{DL}, D_{EC}	functions for generating a key token based on only a key token factor
E	an elliptic curve defined by two elliptic curve coefficients, a_1 and a_2. For the purpose of this document, an elliptic curve is not only the set of points on the curve, but also a group operation defined on these points as specified in ISO/IEC 15946-1[13].
FE2I	a function that converts a field element into an integer (described in Annex A)
FE2OS	a function that converts a field element into an octet string (described in Annex A)
$F(q)$	the finite field with q elements
G,G_a,G_b	points of order r on E over $F(q)$
GE2OS$_X$	a function that converts a group element into an octet string; when the group element is a point on E, this function converts the x-coordinate of the point into an octet string and ignores the y-coordinate (described in Annex A)
g, g_1, g_a, g_b	elements of multiplicative order r in $F(q)$
$g_{q\text{-}1}$	an element of multiplicative order q - 1 in $F(q)$
H	a collision-resistant hash-function taking an octet string as input and giving a bit string as output, e.g. based on one of the dedicated hash-functions specified in ISO/IEC 10118-3[11]
$h(\mathrm{x}, L_K)$	a collision-resistant hash-function taking an octet string x and an integer L_K as input and giving a bit string of length L_K (in bits) as output, e.g. based on one of the dedicated hash-functions specified in ISO/IEC 10118-3[11]
I2FE	a function that converts an integer into a field element (described in Annex A)
I2OS	a function that converts an integer into an octet string (described in Annex A)
I2P	a function that converts an integer into a point on the curve E (described in Annex A)
J, J_{DL}, J_{EC}	functions for generating a password verification element from a password
K	a function for deriving a key from a secret value and a key derivation parameter
KC_1_U	an octet string of the constant value "KC_1_U" that literally means unilateral key confirmation
K_1, K_2, ...	secret keys established using a key establishment mechanism
k	the cofactor that is either the value $(q - 1)/r$ in DL domain parameters or the value of $\#E/r$ in EC domain parameters

L_K	the length (in bits) of an established secret key
MAX	a function that takes two integers as input and outputs the integer with a larger value; if the two integers are identical, it outputs an error message
MIN	a function that takes two integers as input and outputs the integer with a smaller value; if the two integers are identical, it outputs an error message
M_i	an octet that is represented by values from 00 hex to FF hex
m	an integer
$mac(k, m)$	a message authentication code (MAC) function taking a key k and a variable-length message m as input and giving a fixed-length output, e.g., by using one of the MAC algorithms specified in ISO/IEC 9797-2[10]
mod	binary operation, where $y = a \bmod b$ is defined to be the unique integer y satisfying $0 \leq y < b$ and $(a - y)$ is an integer multiple of b
n	an integer
null	an empty octet string with the byte length zero
OS2I	a function that converts an octet string into an integer (described in Annex A)
o_A, o_A', o_B, o_B'	bit strings, which are used to specify a key confirmation process
$P_1, P_2, \ldots$	key derivation parameter octet strings
p, p_i	odd prime integers
q	the number of elements in the finite field $F(q)$. In the EC setting, q is either p or 2^m for some integer $m \geq 1$. In the DL setting, q is p. NOTE 1 This document treats only a prime field or a binary field in the EC setting and only a prime field in the DL setting, because these cases are widely used and their security properties have been well-explored.
$R, R_{1DL}, R_{1EC}, R_{2DL}, R_{2EC}$	functions for deriving a random element from a password
r	the order of the desired group, which is a prime dividing either $q - 1$ in the DL setting or $\#E$ in the EC setting
s_A, s_B	key token factors of entities A and B, respectively, corresponding to key tokens w_A and w_B. The key token factors should be generated at random from a selected range since this maximizes the difficulty of recovering the key token factor by collision-search methods. Methods of random number generation are specified in ISO/IEC 18031[14].
T	a function for checking validity of a key token
$V, V_A, V_B, V_{ADL}, V_{AEC}, V_{BDL}, V_{BEC}$	functions for generating secret values
w_A, w_B	key tokens or password-entangled key tokens of entities A and B respectively, corresponding to key token factors s_A and s_B; they are integers in the DL setting and points in the EC setting
z	a secret value used to derive the keys; it is an integer in the DL setting and a point in the EC setting

π	a password-based octet string which is generally derived from the following data items: a) a password or a hashed password, b) identities for one or more entities, c) an identity of a communication session if more than one session may execute concurrently, and d) a salt value and/or other data. All items except the first are optional. NOTE 2 As addressed in Reference [26], including entity and session identities in the computation of π can prevent an unknown key-share attack. As further addressed in Reference [6], putting these identities into the session key computation function rather than in the computation of π can also avoid a number of attacks, including the unknown key-share attack.
$\#E$	the number of points on the elliptic curve, E
$[x] \times Y$	multiplication operation in the EC setting that takes an integer x and a point Y on the curve E as input and produces a point Z on the curve E, where $Z = [x] \times Y = Y + Y + \ldots + Y$ adding $x - 1$ times if x is positive. The operation satisfies $[0] \times Y = 0_E$ (the point at infinity), and $[-x] \times Y = [x] \times (-Y)$.
$\{\beta_{m-1}, \beta_{m-2}, \ldots, \beta_0\}$	an element of $F(s^m)$ where s is either p or 2, and β_i is an integer satisfying $0 \leq \beta_i \leq s - 1$
$\|\|$	$X \|\| Y$ denotes the result of the concatenation of octet strings X and Y in the order specified. In cases where the result of concatenating two or more octet strings is input to a cryptographic function as part of one of the mechanisms specified in this document, this result shall be composed so that it can be uniquely resolved into its constituent octet strings, i.e. so that there is no possibility of ambiguity in interpretation. This latter property could be achieved in a variety of different ways, depending on the application. For example, it could be guaranteed by a) fixing the length of each of the octet strings throughout the domain of use of the mechanism, or b) encoding the sequence of concatenated octet strings using a method that guarantees unique decoding, e.g. using the distinguished encoding rules defined in ISO/IEC 8825-1[9].
0_E	the point at infinity on the elliptic curve, E

5 Requirements

It is assumed that the entities are aware of each other's claimed identities. This may be achieved by the inclusion of identities in information exchanged between the two entities, or it may be apparent from the context of use of the mechanism.

It is assumed that if the entities are engaged in several sessions in parallel, the entities are aware of a unique session identity for each session. This may be achieved by the inclusion of session identities in information exchanged between the entities, or it may be apparent from the context of use of the mechanism.

It is assumed that the entities are aware of a common set of domain parameters, which are used to compute a variety of functions in the key establishment mechanism. Each mechanism can be used with one of two different sets of domain parameters, depending on whether the mechanism operates over the multiplicative group of values in $F(q)$ or over the additive group of elements in an elliptic curve defined over $F(q)$. In the first case, the mechanism is said to operate in the DL (for "discrete logarithm") setting, and in the second case, the mechanism is said to operate in the EC (for "elliptic curve") setting. The selection of parameters for both cases is discussed in Annex C.

NOTE 1 It is fundamentally important to the correct operation of the mechanisms that any domain parameters are held correctly by each participant. Use by any party of accidentally or deliberately corrupted domain parameters can result in compromise of the mechanisms, which could allow an unauthorized third party to discover an established secret key.

The two sets of domain parameters are as follows.

A set of DL domain parameters consists of:

— $F(q)$ a specific representation of the finite field with q elements;

— q the number of elements in $F(q)$, which is an odd prime integer;

— r the order of the desired group of elements from the finite field, which is a prime divisor of $q - 1$;

— g an element of multiplicative order r in $F(q)$ [g is called the generator of a subgroup of r elements in $F(q)$];

— $g_{q\text{-}1}$ an element of multiplicative order $q - 1$ in $F(q)$;

— k the value $(q - 1)/r$, also called the cofactor, satisfying $k = 2$ or $k = 2p_1p_2...p_t$, for primes $p_i > r$, $i = 1, 2, ..., t$.

A method of generating $g_{q\text{-}1}$ can be found in Chapter 4 of References [18] and [22].

A set of EC domain parameters consists of:

— $F(q)$ a specific representation of the finite field with q elements;

— q the number of elements in $F(q)$, which is

  — p, an odd prime integer, or

  — 2^m for some positive integer $m \geq 1$;

— a_1, a_2 two elliptic curve coefficients, elements of $F(q)$, that define an elliptic curve E;

— E an elliptic curve defined by two elliptic curve coefficients, a_1 and a_2. For the purpose of this document, an elliptic curve is not only the set of points on the curve, but also a group operation defined on these points as specified in ISO/IEC 15946-1[13]. It is defined by one of the following two formulae:

  — $Y^2 = X^3 + a_1X + a_2$ over the field $F(p)$,

  — $Y^2 + XY = X^3 + a_1X^2 + a_2$ over the field $F(2^m)$, together with an extra point 0_E referred to as the point of infinity;

— $\#E$ the number of points on E;

— r the order of the desired group, which is a prime integer dividing $\#E$;

— G a curve point of order r (G is called the generator of a subgroup of r points on E);

— k the value $\#E/r$, also called the cofactor, satisfying $k = 2^n$ or $k = 2^np_1p_2...p_t$, for $n = \{0, 1, 2\}$ and primes $p_i > r$, $i = 1, 2, ..., t$.

NOTE 2 If the cofactor k is $2^np_1p_2...p_t$, for $n = \{0, 1, 2\}$ and primes $p_i < r$, $i = 1, 2, ..., t$, and the key token check function T_{EC} in 6.2.3.3 is used, small subgroup attacks (e.g. see Reference [21]) are possible.

When entities make use of a specified mechanism in the EC setting, it is assumed that the entities are aware of the form of the point representation, i.e. a point is represented in either compressed, uncompressed or hybrid form. For further information on point representations, see ISO/IEC 15946-1[13].

In many of the mechanisms specified in this document, one of the participants is required to select a value at random from a given set of values, e.g. an integer from a specified range. This shall be implemented using a random number generation method based on a random bit generation (e.g. one of the methods in ISO/IEC 18031[14]) in such a way that the selected value is chosen uniformly (or near uniformly) at random from the complete set of possible values. Furthermore, when prime numbers are generated, it is assumed that the generation follows a secure method, e.g., using one of the methods in ISO/IEC 18032[15].