INTERNATIONAL STANDARD

ISO/IEC 11770-4

First edition
2006-05-01

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 11770-4:2006(E)

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

Published in Switzerland

Contents

Page

Foreword

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

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

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

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.

ISO/IEC 11770-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, Subcommittee SC 27, *IT Security techniques*.

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*

Further parts may follow.

Information technology — Security techniques — Key management —

Part 4: Mechanisms based on weak secrets

1 Scope

This part of ISO/IEC 11770 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 off-line brute-force attacks associated with the weak secret. More specifically, these mechanisms are designed to achieve one of the following three goals.

1) **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.

2) **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 – 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*. Therefore it is normally used between a client (*A*) and a server (*B*).

3) **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 – 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 for her. It is normally used between a client (*A*) and a server (*B*).

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

— lifecycle 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.

NOTE – The keys generated or retrieved through the use of weak secrets cannot be more secure against exhaustion than the sum of the weak secrets themselves. With this proviso, the mechanisms specified in this part of ISO/IEC 11770 are recommended for practical use in low-security environments.

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

ISO/IEC 11770-1:1996, *Information technology — Security techniques — Key management — Part 1: Framework*

3 Terms and definitions

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

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 – Computational feasibility depends on the specific security requirements and environment.

[ISO/IEC 10118-1:2000]

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

NOTE – 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 – 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.

[ISO/IEC 9796-3:2000]

3.7
explicit key authentication from *A* to *B*
assurance for entity B that A is the only other entity that is in possession of the correct key

NOTE - Implicit key authentication from *A* to *B* and key confirmation from *A* to *B* together imply explicit key authentication from *A* to *B*.

[ISO/IEC 11770-3:1999]

3.8
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 – Computational feasibility depends on the specific security requirements and environment.

[ISO/IEC 10118-1:2000]

3.9
hashed password
result of applying a hash-function to a password

3.10
implicit key authentication from *A* to *B*
assurance for entity *B* that *A* is the only other entity that can possibly be in possession of the correct key

[ISO/IEC 11770-3:1999]

3.11
key
sequence of symbols that controls the operation of a cryptographic transformation (e.g. encipherment, decipherment, cryptographic check function computation, signature calculation, or signature verification)

[ISO/IEC 11770-3:1999]

3.12
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

[ISO/IEC 11770-1:1996]

3.13
key confirmation from *A* to *B*
assurance for entity *B* that entity *A* is in possession of the correct key

[ISO/IEC 11770-3:1999]

3.14
key control
ability to choose the key, or the parameters used in the key computation

[ISO/IEC 11770-1:1996]

3.15
key derivation function
function that utilizes shared secrets and other mutually known parameters as inputs, and outputs one or more shared secrets, which can be used as keys

3.16
key establishment
process of making available a shared secret key to one or more entities; key establishment includes key agreement, key transport and key retrieval

3.17
key management
administration and use of the generation, registration, certification, deregistration, distribution, installation, storage, archiving, revocation, derivation and destruction of keying material in accordance with a security policy

[ISO/IEC 11770-1:1996]

3.18
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

3.19
key token
key establishment message sent from one entity to another entity during the execution of a key establishment mechanism

3.20
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

3.21
key token factor
value that is kept secret and that is used, possibly in conjunction with a weak secret, to create a key token

3.22
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

3.23
mutual key authentication
assurance for two entities that only the other entity can possibly be in possession of the correct key

3.24
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

[ISO/IEC 11770-3:1999]

3.25
password
secret word, phrase, number or character sequence used for entity authentication, which is a memorized weak secret

3.26
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

3.27
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*

3.28
password-entangled key token
key token which is derived from both a weak secret and a key token factor

3.29
password verification data
data that is used to verify an entity's knowledge of a specific password

3.30
random element derivation function
function that utilizes a password and other parameters as input, and outputs a random element

3.31
salt
random variable incorporated as secondary input to a one-way or encryption function that is used to derive password verification data

3.32
secret
value known only to authorized entities

3.33
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

3.34
secret key
key used with symmetric cryptographic techniques by a specified set of entities

[ISO/IEC 18033-1:2005]

3.35
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 – This might, for example, be achieved by randomly choosing the secret from a sufficiently large set of possible values with an even probability distribution.

3.36
weak secret
secret that can be conveniently memorized by a human being; typically this means that the entropy of the secret is limited, so that an exhaustive search for the secret may be feasible, given knowledge that would enable a correct guess for the secret to be distinguished from an incorrect guess

4 Symbols and notation

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

a_1, a_2	elliptic curve coefficients
A, B	distinguishing identifiers of entities
b, b_i	bits (i.e. either 0 or 1)
BS2I	a function that converts a bit string into an integer
c	an integer satisfying $1 \leq c \leq q - 1$
C,C_{DL},C_{EC}	functions for generating a key token based on a password and a key token factor
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
$F(q)$	the finite field of cardinality q
FE2I	a function that converts a field element into an integer
FE2OS	a function that converts a field element into an octet string
g,g_1,g_a,g_b	elements of multiplicative order r in $F(q)$
G,G_a,G_b	points of order r on E over $F(q)$
g_{q-1}	an element of multiplicative order q -1 in $F(q)$
$GE2OS_X$	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
H	a hash-function taking an octet string as input and giving a bit string as output, e.g. one of the dedicated hash-functions specified in ISO/IEC 10118-3
$h(x, L_K)$	a hash-function taking an octet string x and an integer L_K, which indicates the length (in bits) of output, as input and giving a bit string of length L_K as output, e.g. one of the dedicated hash-functions specified in ISO/IEC 10118-3
I2FE	a function that converts an integer into a field element
I2OS	a function that converts an integer into an octet string
I2P	a function that converts an integer into a point on the curve E
J,J_{DL},J_{EC}	functions for generating a password verification element from a password
k	the cofactor that is either the value $(q\text{-}1)/r$ in DL domain parameters or the value of $\#E/r$ in EC domain parameters
K	a function for deriving a key from a secret value and a key derivation parameter
K_1,K_2,...	secret keys established using a key establishment mechanism
L_K	the length (in bits) of an established secret key
m	an integer
M_i	an octet that is represented by values from 00 hex to FF hex
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

o_A, o_A', o_B, o_B' bit strings, which are used to specify a key confirmation process

OS2I a function that converts an octet string into an integer

p, p_i odd prime integers

P_1,P_2,... key derivation parameter octet strings

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 – this part of ISO/IEC 11770 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 the order of the desired group, which is a prime dividing either $q - 1$ in the DL setting or $\#E$ in the EC setting

R,R_{1DL},R_{1EC}, R_{2DL},R_{2EC} functions for deriving a random element from a password

s_A, s_B Key token factors of entities A and B respectively, corresponding to key tokens w_A and w_B

NOTE – 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.

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

$[x] \times Y$ 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 $[-x] \times Y = [x] \times (-Y)$.

z a secret value used to derive the keys; it is an integer in the DL setting and a point in the EC setting

$\{\beta_{m-1}, \beta_{m-2}, \ldots, \beta_0\}$ 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

π a password-based octet string which is generally derived from a password or a hashed password, identifiers for one or more entities, an identifier of a communication session if more than one session might execute concurrently, and optionally includes a salt value and/or other data

NOTE – It is required to include one or more the entity identifiers and a unique session identifier into the value of π, in order to avoid that a key establishment mechanism might be vulnerable to an unknown key-share attack addressed in [TC05].

$\#E$ the number of points on the elliptic curve E

|| concatenation operator, defined on octet strings

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 identifiers in information exchanged between the two 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.

NOTE – 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 might allow an unauthorised 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 on 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-1} – an element of multiplicative order q -1 in $F(q)$.

NOTE – a method of generating g_{q-1} can be found in Chapter 4 of [MvV96] and [Ka86].

k – the value $(q-1)/r$, also called the cofactor, satisfying $k = 2p_1p_2\ldots p_t$, for primes $p_i > r$, $i = 1, 2, \ldots, t$. Optionally, $t = 0$.

A set of EC domain parameters consists of:

$F(q)$ – a specific representation of the finite field on 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. It is defined by one of the following two equations

- $Y^2 = X^3 + a_1X + a_2$ 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.