INTERNATIONAL STANDARD

ISO/IEC 20009-3

First edition
2022-02

Information security — Anonymous entity authentication —

Part 3: Mechanisms based on blind signatures

Sécurité de l'information — Authentification d'entité anonyme —

Partie 3: Mécanismes fondés sur des signatures aveugles

Reference number
ISO/IEC 20009-3:2022(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

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.

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 https://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*.

A list of all parts in the ISO/IEC 20009 series can be found on the ISO and IEC websites.

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 and www.iec.ch/national-committees.

Introduction

In an anonymous entity authentication mechanism, the entity to be authenticated (the claimant) provides evidence to a verifier that it has knowledge of a secret without revealing its identifier to any unauthorized entity. That is, given complete knowledge of the messages exchanged between the parties, an unauthorized entity cannot discover the identifier of the entity being authenticated. Moreover, it is possible that even an authorized verifier is not authorized to learn the identifier of the entity being authenticated.

The anonymous entity authentication mechanisms specified in this document are based on blind signatures, specified in the ISO/IEC 18370 series.

Information security — Anonymous entity authentication —

Part 3: Mechanisms based on blind signatures

1 Scope

This document provides general descriptions and specifications of anonymous entity authentication mechanisms based on blind digital signatures.

2 Normative references

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 terminology 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
anonymous entity authentication
corroboration that an entity possesses certain *attributes* (3.2), without distinguishing this entity from other entities with the same attributes

[SOURCE: ISO/IEC 20009-1:2013, 2.2]

3.2
attribute
application-specific data element

[SOURCE: ISO/IEC 18370-1:2016, 3.1]

3.3
claimant
entity which is or represents a principal for the purposes of authentication

Note 1 to entry: A claimant includes the functions and the private data necessary for engaging in authentication exchanges on behalf of a principal.

[SOURCE: ISO/IEC 9798-1:2010, 3.6]

3.4
claimant information field
special *credential* (3.6) *attribute* (3.2) encoded within a credential that is not seen by the *issuer* (3.13) during credential issuance, and that is always disclosed to a *verifier* (3.15)

3.5
collision-resistant hash-function
hash-function (3.12) satisfying the following property: it is computationally infeasible to find any two distinct inputs which map to the same output

[SOURCE: ISO/IEC 10118-1:2016, 3.1, modified — Note 1 to entry has been deleted.]

3.6
credential
data held by a *claimant* (3.3) that provides evidence that the claimant is the rightful holder of encoded *attributes* (3.2) and/or a public key, corresponding to a private key

Note 1 to entry: In the context of this definition, attributes can include information regarding the qualification, competence or clearance of the claimant.

3.7
credential information field
special *attribute* (3.2) encoded within a *credential* (3.6) that contains metadata about the credential, such as its expiry date, that is always disclosed to *verifiers* (3.15)

3.8
credential private key
data item specific to a *claimant's* (3.3) *credential* (3.6) that should only be used by this claimant

3.9
credential public key
data item mathematically related to a *credential* (3.6) that is disclosed to the *verifier* (3.15) upon authentication

3.10
domain
set of entities operating under a single security policy

[SOURCE: ISO/IEC 18370-1:2016, 3.11]

3.11
domain parameter
data element which is common to and known by or accessible to all entities within the *domain* (3.10)

[SOURCE: ISO/IEC 14888-1:2008, 3.5]

3.12
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

[SOURCE: ISO/IEC 10118-1:2016, 3.4, modified — Note 1 to entry has been deleted.]

3.13
issuer
entity responsible for provisioning of a *credential* (3.6) to a *claimant* (3.3)

3.14
unilateral anonymous authentication
anonymous entity authentication (3.1) that provides one entity with assurance of the legitimacy of the other entity, but not vice versa

[SOURCE: ISO/IEC 20009-1:2013, 2.20]

3.15
verifier
entity which requires assurance of the legitimacy of another entity (the *claimant* (3.3))

[SOURCE: ISO/IEC 20009-1:2013, 2.22]

4 Symbols and abbreviated terms

$\emptyset$	The null value, a zero-length octet string.
`0x`	Prefix of a hexadecimal value. For example, `0x37c5` represents the two octet values `37` and `c5` in sequence.
$a \in A$	Indicates that element is in set A.
$a\|\|b$	Concatenation of data items a and b in the order specified. In cases where the result of concatenating two or more data items is input to a cryptographic algorithm 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 data strings, i.e. so that there is no possibility of ambiguity in interpretation. This latter property can be achieved in a variety of different ways, depending on the application. For example, it can be guaranteed by: a) fixing the length of each of the substrings throughout the domain of use of the mechanism; or b) encoding the sequence of concatenated strings using a method that guarantees unique decoding, e.g. using the distinguished encoding rules defined in ISO/IEC 8825-1.
$A \subseteq B$	Indicates that set A is a subset of or equal to set B.
$\mathrm{A} \setminus \mathrm{B}$	When A and B are sets, represents the set of elements present in A but not in B.
CI	An extra claimant information field.
$cred$	The claimant's credential.
$\|D\|$	Bit length of D if D is a bit string, or bit size of D if D is a non-negative number (i.e. 0 if $D = 0$, or the unique integer i such that $2^{i-1} \leq D < 2^i$ if $D > 0$).
$desc(G_q)$	Specifies a group G_q of prime order q in which it is infeasible to compute discrete logarithms.
E	Elliptic curve over the finite field F_p for a prime $p > 3$.
$E(F_p)$	Set of all points (x, y), $x \in F_p$, $y \in F_p$, which satisfy the defining equation of the curve E, together with the point at infinity O_E.
$\#E(F_p)$	Order (or cardinality) of $E(F_p)$.
F_p	Finite field containing exactly p elements.
g, g_i	Generators of G_q.
$\gcd(N_1, N_2)$	Greatest common divisor of integers N_1 and N_2.

G_q	Cyclic group of prime order q. For uniformity, the multiplicative notation is used throughout. As such, when using the elliptic curve construction it should be understood that ab represents the group addition of points a and b, that a/b represents the group addition of the point a to the additive inverse of the point b, and that a^b represents the scalar multiplication of point a by the integer b. NOTE This document specifies two constructions for the group G_q in which it is infeasible to compute discrete logarithms. The first is based on a subgroup of a finite field, and the second is based on an elliptic curve over a finite field F_q, where q is a prime number. Each construction is specified by a description denoted by $desc(G_q)$. Details of these two constructions with their corresponding descriptions $desc(G_q)$ are provided in Annex C.
H	Cryptographic hash-function.
I	Finite set of positive integers.
k	Security parameter (a positive integer).
l_q	Security parameter (a positive integer).
n	Positive integer.
$[n]P$	Scalar multiplication operation that takes a positive integer n and a point P on the elliptic curve E as input and produces as output another point Q on the elliptic curve E, where $Q = [n]P = P + P + ... + P$ added $n - 1$ times. The operation satisfies $[0]P = O_E$ (the point at infinity), and $[-n]P = [n](-P)$.
O_E	Point at infinity on the elliptic curve E.
$P + Q$	Elliptic curve sum of points P and Q.
q	Prime number satisfying $\|q\| = l_q$.
TI	A credential information field.
UID_p	A unique identifier for the domain parameters.
Z_p	Set of integers in $[0, p - 1]$ with arithmetic defined modulo p.
Z_N^*	Set of integers U with $0 < U < N$ and $\gcd(U, N) = 1$, with arithmetic defined modulo N.
$\prod_{i \in I} a_i$	Product of the values a_i for which $i \in I$.
$[x, y]$	Set of integers from x to y inclusive, if x, y are integers satisfying $x \leq y$.
$\langle ... \rangle$	Ordered list of values to be hashed.

5 General model and requirements

This clause specifies the general model and requirements for the mechanisms specified in this document.

NOTE 1 Blind signatures, as specified in the ISO/IEC 18370 series, allow a user to obtain a digital signature as specified in the ISO/IEC 9796 series on a message of the user's choice, without giving the signer any information about the actual message or the resulting signature.

An anonymous entity authentication mechanism based on blind signatures involves an issuer, a set of claimants and a set of verifiers. Such an anonymous entity authentication mechanism is defined by the specification of the following processes:

- parameter generation process;
- key generation process;
- credential issuance process;
- authentication process.

Entities of different types can be involved in the mechanism specified in this document, as follows.

- Claimant: an entity to be authenticated in such a way that the claimant's identity is not revealed. In this document, a claimant plays the role of requestor in a blind digital signature scheme, as specified in ISO/IEC 18370-2:2016.
- Verifier: an entity that verifies the validity of a claimant's credential and which does not learn the claimant's identity.
- Issuer: an entity issuing a credential to a claimant. In this document, an issuer plays the role of signer in a blind digital signature scheme as specified in ISO/IEC 18370-2:2016.

  NOTE 2 In the context of this document, the issuer serves as an offline trusted third party (TTP) in the sense of ISO/IEC 20009-1. It gains knowledge of all a claimant's attributes but does not learn which subset is later selected to present the signature.

Annex A lists the object identifiers which shall be used to identify the mechanism defined in this document.

6 Unilateral anonymous authentication

6.1 General

Unilateral anonymous authentication means that only one of the two entities, the claimant, is authenticated by use of the mechanism and that the identity of the authenticated entity is anonymous to the other entity, the verifier.

6.2 Mechanism 1 — Two-pass unilateral anonymous authentication

6.2.1 General

Two-pass means that the authentication phase consists of two messages being exchanged between the claimant and the verifier.

This mechanism is based on mechanism 4 in ISO/IEC 18370-2:2016. In addition to verifying that a claimant possesses a valid credential issued by the issuer, this mechanism also enables a verifier to request the presentation of claimant attributes encoded in the credential. That is, at the end of the authentication process, the verifier is guaranteed that the claimant holds a credential received from the issuer that certifies the attributes disclosed during the authentication process.

The mechanism only guarantees anonymity to the claimant if a credential received from the issuer is used in only one session of the authentication process. If a credential is used in multiple sessions, these sessions can still not be linked to the corresponding session of the credential issuance process. However, they can be linked with each other by the verifiers, even if different sets of attributes are disclosed. In particular, a returning claimant can be recognized by a verifier.

Security considerations and guidance for concrete parameter selections are given in Annex E.

6.2.2 Requirements

In order to use this two-pass unilateral anonymous authentication mechanism, the following requirements apply.

- Each entity involved in this mechanism shall be aware of the public domain parameters.
- The parties shall agree on the security parameters in use.

  NOTE 1 Guidance for parameter choice is given in Clause E.2.

- Each entity shall have access to an authentic copy of the necessary public keys, such as the issuer's verification key.
- The entities involved in this mechanism shall agree in advance of use of the mechanism on a positive integer n, representing the maximum number of attributes that can be encoded in a credential.
- Both issuer and claimant shall have the means to generate integers uniformly at random from a given range. Techniques for generation of sequences of random bits are specified in ISO/IEC 18031. A method for converting a string of bits to an integer in a given range is specified in Annex B.
- A collision-resistant hash-function shall be used. Possible admissible schemes are specified in the ISO/IEC 10118 series.

  NOTE 2 Guidance for hash formatting rules is given in Annex D.
- Prime numbers shall be generated in a secure way. Secure mechanisms can, for instance, be found in ISO/IEC 18032.

Before issuing a credential, the issuer can wish to authenticate the claimant. This document does not specify mechanisms for conventional entity authentication. For this purpose, one of the mechanisms specified in the ISO/IEC 9798 series should be used.

6.2.3 Domain parameters generation process

The set of domain parameters for this two-pass unilateral anonymous authentication mechanism are generated as given in ISO/IEC 18370-2:2016, 8.2.2, and includes the following parameters:

- q: a prime number where $|q| = l_q$;
- G_q: a cyclic group of prime order q;
- $desc(G_q)$: the description of group G_q;
- g: a randomly chosen generator of G_q;
- n: an integer indicating the maximum number of attributes to be certified by the issuer;
- $g_1,..., g_n, g_{n+1}$: n+1 randomly chosen generators of G_q, all distinct from each other and g;

  NOTE 1 An example of recommended parameters for typical security levels is provided in Clause E.2.

  NOTE 2 A method for generating random generators is given in ISO/IEC 14888-3:2018, D.2.2.
- H: a hash-function that outputs a k-bit message digest;
- H_1: $\{0,1\}^* \rightarrow [0, q - 1]$ a hash-function;

  NOTE 3 An example of how to construct H_1 is provided in Clause D.3.
- UID_p: a unique identifier for the domain parameters.

  NOTE 4 UID_p is an octet string specifying an application-specific unique identifier for the domain parameters. For example, UID_p can be computed as the digest of the other domain parameters.

6.2.4 Key generation process

The public and private keys of the issuer are computed as follows:

a) the issuer picks an integer y_0 uniformly at random from the range $[1, q - 1]$;

b) the issuer computes $g_0 = g^{y_0}$.

The signature key is the element y_0 and the verification key is g_0.

6.2.5 Credential issuance process

The claimant and issuer engage in an interactive protocol to issue one credential to the claimant.

In this protocol, the issuer takes as its inputs the domain parameters, the verification key, the signature key, an array of *n* claimant's attributes, consisting of bit strings, to be certified, as well as a credential information field attribute *TI* in form of a bit string.

The claimant takes as its inputs the domain parameters, the public verification key, the array of *n* claimants' attributes to be certified, the credential information field attribute *TI* and an extra claimant information field, the attribute bit string *CI*.

The output of this process for the claimant is a pair of keys (private and public) along with a credential that certifies both the public key and the array of attributes.

The credential consists of a (blind) signature, as in ISO/IEC 18370-2:2016, 8.2.3, on the credential public key and the array of attributes.

NOTE The attributes, which consist of bit strings, can be supplied by either the claimant or the issuer or can be jointly determined by the claimant and the issuer. The negotiation of these attributes, which is performed prior to the execution of the credential issuance process, is outside the scope of this document.

The credential issuance process involves the following steps. The attributes are first converted to integer in $[0, q - 1]$. The resulting vector of *n* claimant attribute to be certified is denoted by $(x_1, ..., x_\mathrm{n})$ where $x_i \in [0, q - 1]$ for all i $(1 \le i \le n)$.

a) The issuer and the claimant both compute a special metadata attribute $y = \mathrm{H}_1\,(\texttt{0x01}||P||TI)$, where $P = \mathrm{H}(UID_p||desc(G_q)||\langle g_0, ..., g_n, g_{n+1}\rangle|| \emptyset || \emptyset)$ is the hash digest of the domain parameters, and $\gamma = g_0 g_1^{x_1} \ldots g_n^{x_n} g_{n+1}^{y} \in G_q$

b) The issuer computes $\sigma_z = \gamma^{y_0}$.

c) The issuer picks an integer, w, uniformly at random from the range $[0, q - 1]$.

d) The issuer computes $\sigma_a = g^w$.

e) The issuer computes $\sigma_b = \gamma^w$.

f) The claimant picks an integer, α, uniformly at random from the range $[1, q - 1]$.

g) The claimant picks an integer, β_1, uniformly at random from the range $[0, q - 1]$.

h) The claimant picks an integer, β_2, uniformly at random from the range $[0, q - 1]$.

i) The claimant computes $h = \gamma^\alpha$.

j) The claimant computes $t_1 = {g_0}^{\beta_1} g^{\beta_2}$.

k) The claimant computes $t_2 = h^{\beta_2}$.

l) The claimant computes $\alpha^{-1} \bmod q$.

m) The issuer sends $(\sigma_z, \sigma_a, \sigma_b)$ to the claimant.

n) The claimant receives $(\sigma_z, \sigma_a, \sigma_b)$ from the issuer.

o) The claimant computes $\sigma'_z = \sigma_z^\alpha$.

p) The claimant computes $\sigma'_a = t_1 \sigma_a$.

q) The claimant computes $\sigma'_b = \sigma_z'^{\beta_1} t_2 \sigma_b^{\alpha}$.

r) The claimant computes $\sigma'_c = \mathrm{H}_1\,(h||CI||\sigma'_z\,||\,\sigma'_a\,||\,\sigma'_b)$.

s) The claimant computes $\sigma_c = \sigma'_c + \beta_1$ mod q.

t) The claimant sends σ_c to the issuer.

u) The issuer receives σ_c from the claimant.

v) The issuer computes $\sigma_r = \sigma_c y_0 + w$ mod q.

w) The issuer sends σ_r to the claimant.

x) The claimant receives σ_r from the issuer.

y) The claimant computes $\sigma'_r = \sigma_r + \beta_2$ mod q.

z) The claimant verifies that $\sigma'_a\sigma'_b = (gh)^{\sigma'_r}\,(g_0\sigma'_z)^{-\sigma'_c}$. If this verification fails, the claimant outputs reject and stops.

aa) The claimant outputs the credential *cred*, consisting of the credential public key h and the (blind) signatures $(\sigma'_z, \sigma'_c, \sigma'_r)$, and the corresponding credential private key α^{-1}.

bb) The claimant stores the attributes $(x_1, ..., x_n)$, the *TI* and *CI* values, the issued credential *cred*, and its associated credential private key α^{-1}.

6.2.6 Authentication process

This authentication process allows the claimant to authenticate to a verifier by presenting a credential along with a selected subset of the encoded attributes.

The claimant takes as its inputs the domain parameters, the issuer public key, the certified attributes, the credential *cred* and the credential private key α^{-1}.

The verifier takes as its inputs the domain parameters and the issuer public key. It gives as output the result of the authentication: valid or invalid.

The protocol is as follows.

a) The verifier sends to the claimant the list $D \subseteq [1, n]$ of attributes indices to disclose and a message bit string pair (m, m_d) to be signed by the credential private key. The resulting signature is called a “presentation proof” in ISO/IEC 18370-2:2016. The set of undisclosed attributes’ indices is denoted by $U = \{1, ..., n\}\backslash D$.

NOTE 1 Only the attributes listed in D are disclosed to the verifier. All others are undisclosed by virtue of the selective disclosure mechanism described in ISO/IEC 18370-2:2016.

NOTE 2 The message to be signed is separated into two parts m and m_d to allow extension mechanisms, not defined herein, to use a second-factor to sign part of the protocol message m_d without seeing all of the protocol details. Both m and m_d can contain application-specific data to be signed by the claimant, the precise content of which is outside the scope of this document. Possible values include timestamps, random challenges or protocol details.

b) The claimant picks an integer, w_0, uniformly at random from the range $[0, q - 1]$.

c) For each $i \in U$, the claimant picks an integer, w_i, uniformly at random from the range $[0, q - 1]$.

d) The claimant computes $a = \mathrm{H}[h^{w_0} \prod_{(i \in U)} g_i^{w_i}]$.

e) The claimant computes $UID_t = \mathrm{H}(h \| \sigma'_z \| \sigma'_c \| \sigma'_r)$.

f) The claimant computes $c_p = \mathrm{H}(UID_t \| a \| \langle D \rangle \| \langle \{x_i\} \ i \in D \rangle \|$ 0x00000000$\|$0x00000000$\|$0x00000000$\|$0x00000000$\|$0x00000000$\|$0x00000000$\| m)$.

g) The claimant computes $c = \mathrm{H}_1 (\langle c_p , m_d \rangle)$.

h) The claimant computes $r_0 = c\alpha^{-1} + w_0 \bmod q$.

i) For each $i \in U$, the claimant computes $r_i = -c\, x_i + w_i \bmod q$.

j) The claimant sends the presentation proof $\{x_i\}_{i \in D}$, y, a, r_0, $\{r_i\}_{i \in U}$, the credential *cred*, consisting of the credential public key h and the (blind) signatures $(\sigma'_z , \sigma'_c , \sigma'_r)$, and the claimant information field attribute *CI* to the verifier.

k) The verifier verifies that $h \neq 1$. If this verification fails, the verifier outputs invalid and stops.

l) The verifier verifies that $\sigma'_c = \mathrm{H}_1(h \| CI \| \sigma'_{\mathrm{z}} \| g^{\sigma'_r} g_0^{-\sigma'_c} \| h^{\sigma'_r} \sigma'_{\mathrm{z}}{}^{-\sigma'_c})$. If this verification fails, the verifier outputs invalid and stops.

m) The verifier computes $UID_t = \mathrm{H}(h \| \sigma'_z \| \sigma'_c \| \sigma'_r)$.

n) The verifier computes $c_p = \mathrm{H}(UID_t \| a \| \| \| \langle D \rangle \| \langle \{x_i\} \ {}_{i \ \in \ D} \rangle \|$ 0x00000000$\|$0x00000000$\|$0x00000000$\|$0x00000000$\|$0x00000000$\|$0x00000000$\| m)$.

o) The verifier computes $c = \mathrm{H}_1 (\langle c_p , m_d \rangle)$.

p) The verifier verifies that $a = \mathrm{H}\left(\left[g_0 g_{n+1}^{y} \prod_{(i \in D)} g_i^{x_i} \right]^{-c} h^{r_0} \left[\prod_{(i \in U)} g_i^{r_i} \right] \right)$. If this verification fails, the verifier outputs invalid, and valid otherwise.

Annex A
(normative)

Object identifiers

This annex gives the object identifiers assigned to the mechanism defined in this document.

```
AnonymousEntityAuthentication-3 {
iso(1) standard(0) anonymous-entity-authentication(20009) part3(3)
asn1-module(0) object-algorithm-identifiers(0)
}
DEFINITIONS EXPLICIT TAGS::= BEGIN
-- EXPORTS All; --
-- IMPORTS None; --
OID::= OBJECT IDENTIFIER -- alias
-- Synonyms --
is20009-3 OID::= { iso(1) standard(0) anonymous-entity-authentication(20009) part3(3) }
mechanism OID::= { is20009-3 mechanisms(1) }
 -- Assignments --
area-U-Prove OID::= { mechanism 1 }
END -- AnonymousEntityAuthentication-3 –
```

Annex B
(informative)

Conversion functions

Primitives BS2IP and I2BSP convert between bit strings and integers, and are defined as follows.

- The function BS2IP(x) maps a bit string x to an integer value m as follows. If $x = (x_{l-1}, ..., x_0)$ where $x_0, ..., x_{l-1}$ are bits, then the value m is defined as $m = 2^{l-1}\, x_{l-1} + 2^{l-2}\, x_{l-2} + ... + 2x_1 + x_0$.
- The function I2BSP(m, l) takes as input two non-negative integers m and l, and outputs the unique bit string x of length l such that BS2IP(x) = m, if such an x exists. Otherwise, the function outputs an error message.

Annex C
(informative)

Group description

The ISO/IEC 18370 series defines two constructions for the group G_q. Either construction may be used for the mechanisms described in this document. Each construction is specified by a description $desc(G_q)$.

- **Subgroup construction**: The description $desc(G_q) = (p, q, g)$ specifies a subgroup G_q of prime order q of a finite field of order p. Both p and q are prime numbers, q divides $p - 1$, and g is a generator of G_q. It is recommended to use the method defined in ISO/IEC 14888-3:2018, Annex D, to generate the group description (p, q, g).
- **Elliptic curve construction**: The description $desc(G_q) = (p, a, b, g, q, 1)$ specifies an elliptic curve over a finite field F_p, where p is a prime number, a and b are two field elements defining the elliptic curve, g is a base point of prime-order q on the curve (and the generator of G_q), q is the order of the group, and 1 is the cofactor of the curve (which implies that $\#E(F_p) = q$). Methods of generating pseudo-random elliptic curves and points of prime order q (the order of the elliptic curve E) are given in ISO/IEC 15946-5 and examples of pseudo-random elliptic curves are given in ISO/IEC 15946-5:2017, Clause C.1.

All entities involved in the mechanisms described in this document should check that all externally received mathematical elements belong to their corresponding algebraic structures prior to relying on or computing with them; failure to do so can result in critical security or privacy problems as pointed out, for example, by Lim and Lee[13] or Pavloski and Boyd[17]. For an element $x \in Z_q$, this means verifying that $0 \leq x < q$. For an element $x \in G_q$, it is sufficient to make sure the curve equation holds when using the elliptic curve construction and to verify that $0 < x < p$ and that $x^q = 1$ when using the subgroup construction.

NOTE 1 For the elliptic curve construction, since the cofactor is 1 for curves of prime order, all curve points are part of the group, and therefore checking that the curve equation holds is enough to verify that a point is part of the group.

NOTE 2 In the subgroup construction, selecting p and q as prime numbers such that $(p - 1)/(2q)$ has no prime factor less than q mitigates attacks of the type described by Lim and Lee[13] or Pavloski and Boyd[17]. Ideally, p and q are chosen such that $(p - 1)/(2q)$ is prime.

Annex D
(informative)

Special hash-functions

D.1 Hash-function inputs

To prevent ambiguous interpretations of the inputs to a hash algorithm, the input data should be encoded as follows, depending on its type (see Section 2.2. of Reference [16]).

- A byte: the value is encoded directly.
- The length of an octet string, the length of a list and the index of an attribute: the binary value is conditionally zero-extended to a length of 32 bits. The four bytes forming the extended value are then encoded, leading with the most-significant byte (e.g. the value 11 588 062 is encoded as `0x00b0d1de`). Such values are therefore in the range $\{0, \ldots, 2^{32} - 1\}$; larger values should be rejected.
- An octet string: the length of the string is encoded followed by the contents of the string (e.g. the string `0x01fe` is encoded as `0x0000000201fe`).
- An element of Z_q, an element of G_q, the values p and q in $desc(G_q)$ for a subgroup construction, and the values p, a, b and q in $desc(G_q)$ for an elliptic curve construction: the binary value is conditionally zero-extended to make its length a multiple of 8 bits (the value 0 is zero-extended to a full 8 bits). The bytes forming the extended value are then encoded as an octet string, leading with the most significant byte (e.g. the number 254 666 256 150 is encoded as `0x000000053b4b4aaf16`).
- A list (delimited with ⟨… ⟩): the length of the list is encoded followed by the recursive encoding of the list elements, in order.
- The null value ($\emptyset$): a zero-length octet string is encoded, yielding the sequence `0x00000000`.
- A point $e = (e_x, e_y)$ on an elliptic curve (all elements of G_q when using the elliptic curve construction): the point is converted to an octet string following the procedure described in Section 2.3.3 of Reference [12], without using point compression.

D.2 Hash-function with larger output length: HL

HL is a cryptographic function that hashes a string m into $\{0, 1\}^k$ based on a hash-function H: $\{0, 1\}^* \rightarrow \{0, 1\}^h$ in the ISO/IEC 10118 series, where $k > h$. HL is constructed using MGF1 in PKCS#1. It involves the following steps.

a) If $k > 2^{32}h$, output “Fail” and stop.

b) Let T be an empty binary string.

c) For i from 0 to $[k/h] - 1$, set $T = T \,||\, \mathrm{H}(m||\ \mathrm{I2BSP}(i, 32))$.

d) Return the leading k bits of T.

D.3 Hashing to an element of a prime field: HBS2PF

HBS2PF is a cryptographic function that hashes a string m into an element in Z_p.