INTERNATIONAL STANDARD

ISO/IEC 9796-3

First edition
2000-04-15

Information technology — Security techniques — Digital signature schemes giving message recovery —

Part 3: Discrete logarithm based mechanisms

Technologies de l'information — Techniques de sécurité — Schéma de signature numérique rétablissant le message —

Partie 3: Mécanismes basés sur les logarithmes discrets

Reference number
ISO/IEC 9796-3:2000(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed 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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. 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 part of ISO/IEC 9796 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 9796-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

This first edition cancels and replaces ISO/IEC 9796:1991, which has been technically revised.

ISO/IEC 9796 consists of the following parts, under the general title *Information technology — Security techniques — Digital signature schemes giving message recovery*:

— *Part 2: Mechanisms using a hash-function*

— *Part 3: Discrete logarithm based mechanisms*

Annex A forms a normative part of this part of ISO/IEC 9796. Annexes B to D are for information only.

Introduction

Digital signature mechanisms can be used to provide services such as entity authentication, data origin authentication, non-repudiation, and integrity of data.

A digital signature mechanism satisfies the following requirements:

— Given only the verification key and not the signature key it is computationally infeasible to produce any message and a valid signature for this message.

— The signatures produced by a signer can neither be used for producing any new message and a valid signature for this message nor for recovering the signature key.

— It is computationally infeasible, even for the signer, to find two different messages with the same signature.

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

Most digital signature mechanisms are based on asymmetric cryptographic techniques and involve three basic operations:

— A process of generating pairs of keys, where each pair consists of a private signature key and the corresponding public verification key.

— A process using the signature key; called the signature process.

— A process using the verification key; called the verification process.

There are two types of digital signature mechanisms:

— When, for each given signature key, the signatures produced for the same message are the same, the mechanism is said to be non-randomized (or deterministic, see ISO/IEC 14888-1).

— When, for a given message and a given signature key, each application of the signature process produces a different signature, the mechanism is said to be randomized.

Digital signature schemes can also be divided into the following two categories:

— When the whole message has to be stored and/or transmitted along with the signature, the mechanism is named a ”signature mechanism with appendix” (see ISO/IEC 14888).

— When the whole message or a part of it is recovered from the signature, the mechanism is named a ”signature mechanism giving message recovery” (see ISO/IEC 9796).

NOTE Any signature mechanism giving message recovery, for example, the mechanisms specified in ISO/IEC 9796, can be converted for provision of digital signatures with appendix. In this case, the signature is produced by application of the signature mechanism to a hash-token of the message.

The mechanisms specified in ISO/IEC 9796 give either total or partial recovery, aiming at reducing storage and transmission overhead.

The mechanisms specified in this part of ISO/IEC 9796 use a hash-function for hashing the entire message. ISO/IEC 10118 specifies hash-functions for digital signatures. If the message is short enough, then the entire message can be included in the signature, and recovered from the signature in the verification process. Otherwise, a part of the message can be included in the signature and the rest of it is stored and/or transmitted along with the signature.

Information technology — Security techniques — Digital signature schemes giving message recovery —

Part 3:
Discrete logarithm based mechanisms

1 Scope

This part of ISO/IEC 9796 specifies two randomized digital signature schemes giving message recovery. The security of both schemes is based on the difficulty of the discrete logarithm problem. The first scheme is defined on a prime field and the second one on an elliptic curve.

This part of ISO/IEC 9796 also defines a redundancy scheme using hash-codes and specifies how the basic signature schemes are to be combined with the redundancy scheme.

This part of ISO/IEC 9796 also defines an optional control field in the hash-token, which can provide added security to the signature.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9796. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 9796 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 10118 (all parts), *Information technology – Security techniques – Hash-functions*.

ISO/IEC 11770-3:1999, *Information technology – Security techniques – Key management – Part 3: Mechanisms using asymmetric techniques*.

ISO/IEC 14888-1:1998, *Information technology – Security techniques – Digital signatures with appendix – Part 1: General*.

ISO/IEC 15946 (parts 1 and 2, to be published), *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 1: General* and *Part 2: Digital signatures*.

3 Terms and definitions

For the purposes of this part of ISO/IEC 9796, the following definitions apply.

3.1 assignment

[ISO/IEC 14888-1] A data item which is a function of the witness and possibly of a part of the message, and forms part of the input to the signature function.

3.2 certification authority

[ISO/IEC 11770-3] A centre trusted to create and assign public key certificates. Optionally, the certification authority may create and assign keys to the entities.

3.3 collision-resistant hash-function
[ISO/IEC 10118-1] A 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.

3.4 data input
A data item which depends on the entire message and forms a part of the input to the signature function.

3.5 domain parameter
[ISO/IEC14888-1] A 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.

3.6 hash-code
[ISO/IEC 10118-1] The string of bits which is the output of a hash-function.

3.7 hash-function
[ISO/IEC 10118-1] A function which maps strings of bits 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; and

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

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

3.8 hash-token
[ISO/IEC 14888-1] A concatenation of a hash-code and an optional control field, which can be used to identify the hash-function and the padding method.

NOTE The control field with hash-function identifier is mandatory unless the hash-function is uniquely determined by the signature mechanism or by the domain parameters.

3.9 message
A string of bits of any length.

3.10 pre-signature
[ISO/IEC 14888-1] A value computed in the signature process which is a function of the randomizer but is independent of the message.

3.11 public key certificate
[ISO/IEC 11770-3] The public key information of an entity signed by the certification authority and thereby rendered unforgeable.

NOTE In the context of this part of ISO/IEC 9796 the public key information contains the information about the verification key and the domain parameters.

3.12 randomized
[ISO/IEC 14888-1] Dependent on a randomizer.

3.13 randomizer
[ISO/IEC 14888-1] A secret data item produced by the signing entity in the pre-signature production process, and not predictable by other entities.

3.14 signature
The string of bits resulting from the signature process.

NOTE This string of bits may have internal structure specific to the signature mechanism. The signatures produced by the mechanisms specified in this part of ISO/IEC 9796 have two parts, of which only the second one depends on the signature key.

3.15 signature function
[ISO/IEC 14888-1] A function in the signature process which is determined by the signature key and the domain parameters. A signature function takes the assignment and possibly the randomizer as inputs and gives the second part of the signature as output.

NOTE In the context of this part of ISO/IEC 9796, the assignment is the data input.

3.16 signature key
[ISO/IEC 14888-1] A secret data item specific to an entity and usable only by this entity in the signature process.

3.17 signature process
[ISO/IEC 14888-1] A process which takes as inputs the message, the signature key and the domain parameters, and which gives as output the signature.

3.18 signed message
[ISO/IEC 14888-1] A set of data items consisting of the signature, the part of the message which cannot be recovered from the signature, and an optional text field.

3.19 verification function
[ISO/IEC 14888-1] A function in the verification process which is determined by the verification key and which gives a recomputed value of the witness as output.

3.20 verification key
[ISO/IEC 14888-1] A data item which is mathematically related to an entity's signature key and which is used by the verifier in the verification process.

3.21 verification process
[ISO/IEC 14888-1] A process which takes as input the signed message, the verification key and the domain parameters, and which gives as output the result of the signature verification: valid or invalid.

3.22 witness
[ISO/IEC 14888-1] A data item which provides evidence to the verifier.

NOTE In the context of this part of ISO/IEC 9796 the witness is based on a hash-token.

4 Symbols, conventions, and legend for figures

4.1 Symbols and notation

The following symbols and notation are used in this part of ISO/IEC 9796.

D, D'	data input, recovered data input, respectively
F	finite field
G	element of a prime field or point on an elliptic curve
H, H', H''	hash-token, recovered (truncated) hash-token, recomputed (truncated) hash-token, respectively
$\mathfrak{I}$	elliptic curve

K	randomizer
k, m, n	positive integers
L	length in bytes of (truncated) hash-token
L_F	size of the field F
L_{rec}, L_{clr}, L_P, L_Q	length in bytes of M_{rec}, M_{clr}, P, and Q, respectively
L_1, L_2	length in bytes of short and long redundancy, respectively
len_Q	length in bits of Q
M, M_{clr}, M_{rec}	message, nonrecoverable part of message, and recoverable part of message, respectively
M', M'_{rec}	recovered message, recovered part of message, respectively
NA	Nth multiple of a point A on an elliptic curve
O	point at infinity on an elliptic curve
P	prime number
Π, Π'	pre-signature, recovered pre-signature, respectively
Q	prime number, order of G
R	first part of the signature
S	second part of the signature
x	x-coordinate of a point on an elliptic curve
X	signature key
y	y-coordinate of a point on an elliptic curve
Y	verification key
$A * B$	sum of points A and B on an elliptic curve
U mod V	the remainder, when integer U is divided by integer V
$U \equiv V \pmod{W}$	integer U is congruent to integer V modulo integer W

4.2 Coding convention, length and size of the field

All integers are written with the most significant digit (or bit, or byte) in the leftmost position.

Given a non-negative integer n and an integer U in the range $0 \le U < 2^n$, the integer U is converted to a string of bits of length n using its binary representation as shown below:

$$U = u_1 \cdot 2^{n-1} + u_2 \cdot 2^{n-2} + \ldots + u_{n-1} \cdot 2 + u_n \rightarrow (u_1, u_2, \ldots, u_{n-1}, u_n).$$

Conversely, a string of bits of length n is converted to an integer by the rule

$$(u_1, u_2, \ldots, u_{n-1}, u_n) \rightarrow u_1 \cdot 2^{n-1} + u_2 \cdot 2^{n-2} + \ldots + u_{n-1} \cdot 2 + u_n.$$

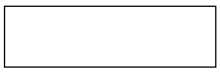
If an integer U is in the range $2^{k-1} \leq U < 2^k$, it is said that the length of U in bits is equal to k, and the notation $k = len_U$ is used.

If an integer U is in the range $256^{m-1} \leq U < 256^m$, it is said that the length of U in bytes equals m, and the notation $m = L_U$ is used. Hence, L_U is the least integer with the property $8 \cdot L_U \geq len_U$.

If F is a field over a prime P, the field size L_F is defined as $L_F = L_P$. If F is a field over a power of two 2^n, the field size L_F is defined as the least integer with the property $8 \cdot L_F \geq n$.

4.3 Legend for figures

The legend for figures in this part of ISO/IEC 9796 is as follows:



step of the process

mandatory data

optional data

5 Requirements

Users who wish to employ a digital signature mechanism from this part of ISO/IEC 9796 shall select:

— a signature scheme from between the two specified in Clause 9 and Clause 10;

— size L_F of the underlying field F;

— parameter Q for the selected signature scheme;

— the byte length L_1 of the short redundancy and the byte length L_2 of the long redundancy, where $8L_1$ and $8L_2$ are strictly less than the length len_Q of the binary representation of Q, and

— a collision-resistant hash-function which produces hash-codes such that the length of the resulting hash-tokens in bytes is L_2 .

Agreement on these choices amongst the users is essential for the purpose of the operation of the digital signature mechanism giving message recovery.

Short redundancy is used if the entire message is recoverable from the signature. Then the bit-length of the message is at most $len_Q - 8L_1 - 1$ bits.

Long redundancy is used if a part of the message is not recoverable from the signature. Then the recoverable part of the message is at most $len_Q - 8L_2 - 1$ bits.

NOTE The sizes of the parameters Q, L_1 and L_2 also affect the security level of the signatures giving message recovery. Typical values of L_1 are 8 or 10. Typical values of L_2 vary from 17 to 21. It is also possible to set $L_1 = L_2$.

5.1 Options for binding signature mechanism and hash-function

When a digital signature mechanism uses a hash-function, there shall be a binding between the signature mechanism and the hash-function in use. Without such a binding, an adversary may claim the use of a weak hash-function (and not the actual one) and thereby forge the signature. There are various ways to accomplish the required binding. The following options are presented in order of increasing risk.

a) Require a particular hash-function when using a particular signature mechanism. The verification process shall exclusively use that particular hash-function. ISO/IEC 14888-3 gives an example of this option where the DSA mechanism requires the use of SHA-1.

b) Allow a set of hash-functions and explicitly indicate in every signature the hash-function in use by a hash-function identifier included as part of the signature calculation. The hash-function identifier is an extension of the hash-code: it indicates how to derive the hash-code. The verification process shall exclusively use the hash-function indicated by the identifier in the signature. This part of ISO/IEC 9796 and ISO/IEC 9796-2 give examples of this option.

c) Allow a set of hash-functions and explicitly indicate the hash-function in use in the certificate domain parameters. Within the domain, the verification process shall exclusively use the hash-function indicated in the certificate. Outside the domain, there is a risk due to less rigorous certification authorities. If other certificates may be created, then other signatures may be created. Then the attacked user would be in a dispute situation with the certification authority that produced the other certificate.

d) Allow a set of hash-functions and indicate the hash-function in use by some other method, e.g., an indication in the message or a bilateral agreement. The verification process shall exclusively use the hash-function indicated by the method. However, there is a risk that an adversary may forge a signature using another hash-function.

The user of a digital signature mechanism should conduct a risk assessment considering the costs and benefits of the various alternatives. This assessment includes the cost associated with the possibility of a bogus signature being produced.

6 Signature process

Figure 1 shows the signature process, which consists of the following steps:

— producing a randomizer and the pre-signature

— splitting the message

— hashing the message

— formatting the data input

— computing the signature

— formatting the signed message

6.1 Producing the pre-signature

Pre-signature is an intermediate data item produced in any randomized signature mechanism. First a randomizer K, which is an integer, is produced. Then the pre-signature Π is computed as an element in the finite field specified by the signature scheme, see 9.3 or 10.3, respectively. The pre-signature is a public data item, while the value of the randomizer shall be usable only by the signature process.

Randomizers can be produced and corresponding pre-signatures computed off-line and stored securely for future use by the signature process.

6.2 Producing the hash-token

The input to the hash-function is formed by concatenating the following strings of bits from the left to the right

— the 64-bit string representing the integer L_{rec};