INTERNATIONAL STANDARD

ISO/IEC 14888-1

First edition
1998-12-15

Corrected and reprinted
1999-12-15

Information technology — Security techniques — Digital signatures with appendix —

Part 1: General

Technologies de l'information — Techniques de sécurité — Signatures digitales avec appendice —

Partie 1: Généralités

Reference number
ISO/IEC 14888-1:1998(E)

Contents

Page

Foreword

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

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.

International Standard ISO/IEC 14888-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

ISO/IEC 14888 consists of the following parts, under the general title *Information technology — Security techniques — Digital signatures with appendix*:

— *Part 1: General*

— *Part 2: Identity-based mechanisms*

— *Part 3: Certificate-based mechanisms*

Introduction

Digital signature mechanisms are asymmetric cryptographic techniques which can be used to provide entity authentication, data origin authentication, data integrity and non-repudiation services. There are two types of digital signature mechanisms:

— When the verification process needs the message as part of the input, the mechanism is named a "signature mechanism with appendix". A hash-function is involved in the calculation of the appendix. ISO/IEC 10118 specifies hash-functions for use in digital signatures with appendix.

— When the verification process reveals the message together with its specific redundancy (sometimes called the shadow of the message), the mechanism is named a "signature mechanism giving message recovery". Redundancy schemes designed for use as part of such a signature scheme are specified in the multipart standard ISO/IEC 9796.

These two types are not mutually exclusive. Specifically, any digital signature mechanism giving message recovery, for example, the mechanism specified in ISO/IEC 9796, can be used for provision of digital signatures with appendix. In this case, the signature is generated by application of the signature process to a hash-token of the message.

Information technology — Security techniques — Digital signatures with appendix —

Part 1:
General

1 Scope

ISO/IEC 14888 specifies several digital signature mechanisms with appendix for messages of arbitrary length. This part of ISO/IEC 14888 contains general principles and requirements for digital signatures with appendix. It also contains definitions and symbols common to all parts of ISO/IEC 14888.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 14888. 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 14888 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 9796:1991, *Information technology — Security techniques — Digital signature scheme giving message recovery*.

ISO/IEC 9796-2:1997, *Information technology — Security techniques — Digital signature schemes giving message recovery — Part 2: Mechanisms using a hash-function.*

ISO/IEC 10118-1:1994, *Information technology — Security techniques — Hash functions — Part 1: General.*

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

3 General

The mechanisms specified in ISO/IEC 14888 are based upon asymmetric cryptographic techniques. Every asymmetric digital signature mechanism involves three basic operations.

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

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

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

The verification of a digital signature requires the signing entity's verification key. It is thus essential for a verifier to be able to associate the correct verification key with the signing entity, or more precisely, with (parts of) the signing entity's identification data. If this association is somehow inherent in the verification key itself, the scheme is said to be "identity-based". If not, the association between the correct verification key with the signing entity's identification data shall be provided by another means. Whatever the nature of such means, the scheme is then said to be "certificate-based".

The procedures of validation and management of verification keys in a certificate-based scheme is outside the scope of ISO/IEC 14888. Mechanisms for distribution of public verification keys are provided in ISO/IEC 11770-3.

4 Terms and definitions

For the purposes of ISO/IEC 14888, the following terms and definitions apply.

4.1
appendix
a string of bits formed by the signature and an optional text field

4.2
assignment
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

4.3
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.

4.4
deterministic
independent of a randomizer, not randomized

4.5
digital signature
see signature

4.6
domain parameter
a data item which is common to and known by or accessible to all entities within the domain

4.7
hash-code
the string of bits which is the output of a hash-function

4.8
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

— 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.

4.9
hash-token
a concatenation of a hash-code and an optional control field, called hash-function identifier, which can be used to identify the hash-function and the padding method

4.10
identification data
a sequence of data items, including the distinguishing identifier for an entity, assigned to an entity and used to identify it

NOTE The identification data may additionally contain data items such as identifier of the signature process, identifier of the signature key, validity period of the signature key, restrictions on key usage, associated security policy parameters, key serial number, or domain parameters.

4.11
message
a string of bits of any length

4.12
pre-signature
a value computed in the signature process which is a function of the randomizer but which is independent of the message

4.13
randomized
dependent on a randomizer

4.14
randomizer
a secret data item produced by the signing entity in the pre-signature production process, and not predictable by other entities

4.15
signature
[ISO/IEC 9796] the string of bits resulting from the signature process

NOTE This string of bits may have internal structure specific to the signature mechanism.

4.16
signature equation
an equation defining the signature function

4.17
signature function
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

4.18
signature key
a secret data item specific to an entity and usable only by this entity in the signature process

4.19
signature process
a process which takes as inputs the message, the signature key and the domain parameters, and which gives as output the signature

4.20
signed message
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

NOTE In the context of this part of ISO/IEC 14888 the entire message is included in the signed message and no part of the message is recovered from the signature.

4.21
verification function
a function in the verification process which is determined by the verification key and which gives a recomputed value of the witness as output

4.22
verification key
a data item which is mathematically related to an entity's signature key and which is used by the verifier in the verification process

4.23
verification process
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

4.24
witness
a data item which provides evidence to the verifier

5 Symbols, conventions, and legend for figures

5.1 Symbols

Throughout all parts of ISO/IEC 14888 the following symbols are used.

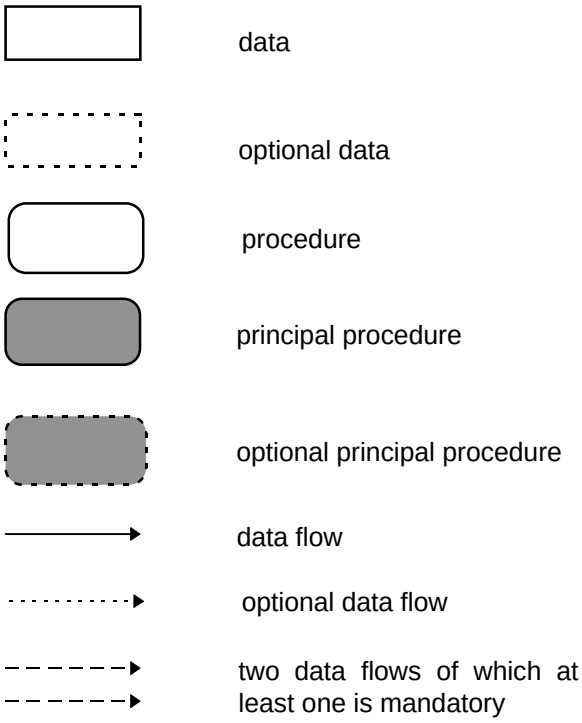
H	Hash-token
$\overline{H}$	Recomputed hash-token
K	Randomizer
M	Message
M_1, M_2	Parts of the prepared message
R	First part of a signature
$\overline{R}$	Recomputed first part of a signature
S	Second part of a signature
T	Assignment
X	Signature key
Y	Verification key
Z	Set of one or more domain parameters
Π	Pre-signature
$\overline{\Pi}$	Recomputed pre-signature
Σ	Signature
$A \bmod N$	The remainder obtained when integer A is divided by integer N
$A \equiv B \pmod N$	Integer A is congruent to integer B modulo N, i.e., $(A - B) \bmod N = 0$.

5.2 Coding convention

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

5.3 Legend for figures

The legend for the figures of all parts of ISO/IEC 14888 is the following.



6 General model

A digital signature mechanism with appendix is defined by the specification of the following processes:

— key generation process;

— signature process;

— verification process.

In the signature process the signing entity computes its digital signature for a given message. The signature, together with an optional text field, form the appendix, which is appended to the message to form the signed message.

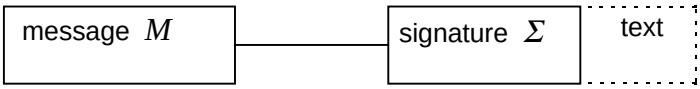


Figure 1 — Signed message

Depending on the application, there are different ways of forming the appendix and associating it to the message. The general requirement is that the verifier is able to relate the correct signature to the message.

For successful verification it is also essential that, prior to the verification process, the verifier is able to associate the correct verification key with the signature. The optional text field can be used for transmitting the signer's identification data or an authenticated copy of the signer's verification key to the verifier. In some cases the signer's identification data may need to be part of the message *M*, so that it gets protected by the signature.

A digital signature mechanism shall satisfy 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.

7 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. In this clause four options are presented in order of increasing risk.

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.

7.1 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.

7.2 Allow a set of hash-functions and explicitly indicate in every signed message 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 signed message. ISO/IEC 9796-2 gives an example of this option.

7.3 Allow a set of hash-functions and explicitly indicate the hash-function in use in the certificate domain parameters. Inside the certificate domain, the verification process shall exclusively use the hash-function indicated in the certificate. Outside the certificate domain, there is a risk due to unrigorous 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.

7.4 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 other method. However, there is a risk that an adversary may forge a signature using another hash-function.

8 Key generation process

The key generation process of a digital signature mechanism consists of the following two procedures:

— generating domain parameters;

— generating signature key and verification key.

The first procedure is executed once when the domain is set up. The resulting set Z of domain parameters is needed in subsequent processes and functions. The second procedure is executed for each signer entity within the domain and the outputs are the signature key X and the verification key Y. For a specific set of domain parameters, a value of X, which is different with overwhelming probability from values used previously, shall be used.

NOTE Validation of domain parameters and keys may be required. However, it is outside the scope of this part of ISO/IEC 14888.

9 Signature process

The following data items are required for the signature process:

— domain parameters Z

— signature key X

— message M

— hash-function identifier (optional)

— other text (optional).

The hash-function identifier can be used for binding the signature mechanism and the hash-function, see clause 7.

The signature process of a digital signature mechanism with appendix consists of the following procedures:

— producing pre-signature

— preparing message for signing

— computing witness

— computing signature.

The first procedure is optional. A signature mechanism without pre-signature is said to be deterministic. A signature mechanism with pre-signature is said to be randomized.

A witness for a digital signature is a data item, the value of which is determined in the signature process. The correctness of the value of the witness is verified in the verification process. A witness is computed as a function of the message or the pre-signature or both.

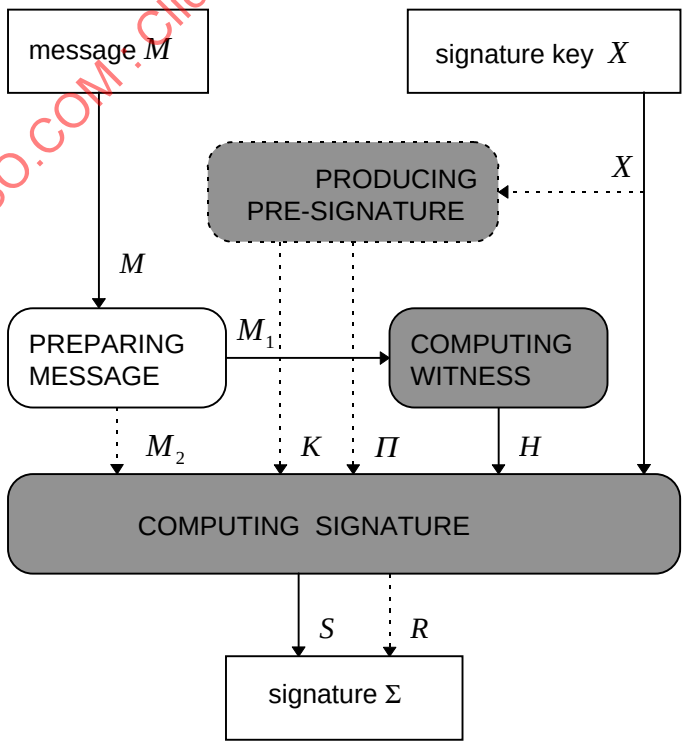


Figure 2 — Signature process with a deterministic witness

If a witness is independent of the pre-signature or pre-signatures do not exist, it is said to be deterministic. A deterministic witness need not be given to the verifier, who can compute it in the same way as the signer computes it as a function of the message only. The signature process with deterministic witness is depicted in Figure 2.

If a witness depends on the pre-signature it is said to be randomized. The value of a randomized witness is computed by the signer and it forms the first part of the signature. The signature process with a randomized witness is depicted in Figure 3.

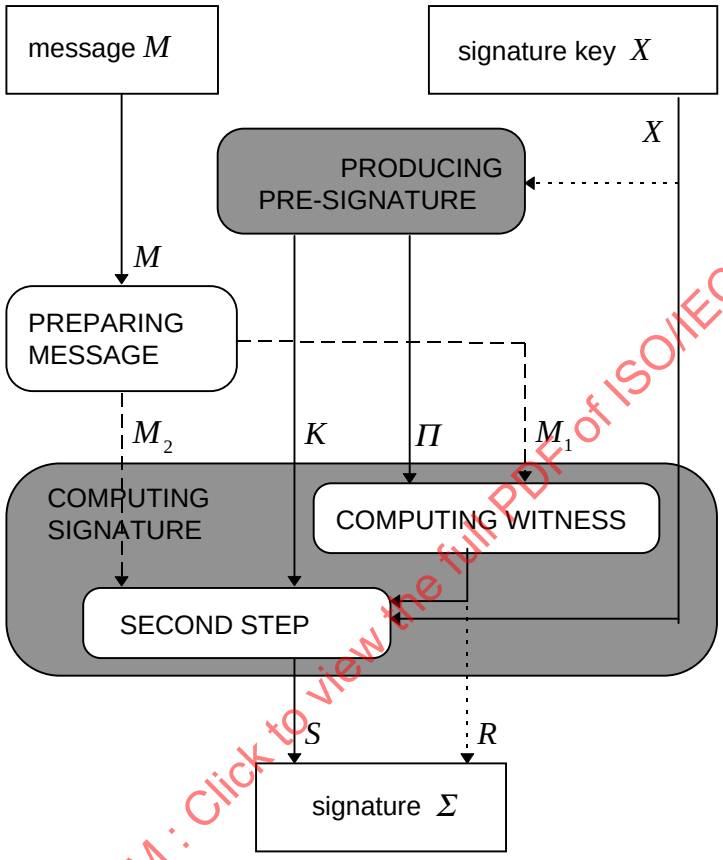


Figure 3 — Signature process with a randomized witness

9.1 Producing pre-signature

A pre-signature procedure is needed in a randomized signature mechanism and consists of the following two steps:

— producing the randomizer K

— computing the pre-signature Π.

The output from the first step is the randomizer *K*, which is a secret value usable only by the signature process. For each message, a value of *K* which is different with overwhelming probability from the values used previously (within the lifetime of the signature key) shall be used to protect the secrecy of the signature key. In the next step the pre-signature Π is computed from *K* by using a function which is determined by the domain parameters *Z* and possibly by the signature key *X*. The outputs are the randomizer *K* and the pre-signature Π.

Randomizers can be produced and corresponding pre-signatures can be computed off-line and stored securely for future use by the signature process. If this off-line feature is required then the randomizer cannot be computed as a (pseudo-random) function of the message.

9.2 Preparing message

The signature process can take (parts of) the message as input either into the computation of the witness or into the computation of (the second part of) the signature, or both. For this purpose, two data fields M_1 and M_2 are derived from the message M. The process of preparing the message shall satisfy one of the following two conditions:

— The entire message M shall be reconstructible given M_1 and M_2.

— It shall be computationally infeasible to find two messages M and M' such that the derived two pairs (M_1, M_2) and (M_1', M_2') are equal.

Typically, in the first case, $M_1 = M$ and M_2 is empty, or $M_2 = M$ and M_1 is empty, or $M_1 = M_2 = M$. In the second case, either M_1 or M_2, or both, are hash-tokens of M.

9.3 Computing witness

A deterministic witness is computed as a hash-token H of M_1 using a collision-resistant hash-function (see Figure 2). Unless the hash-function is uniquely determined by the signature mechanism or by the domain parameters, the hash-function identifier shall be included in the hash-token and in the signed message (see Clause 7).

A randomized witness depends on the pre-signature Π and, optionally, on M_1. The randomized witness is computed as a part of the signature computation and is described in 9.4.

9.4 Computing signature

In a deterministic mechanism, the inputs to this procedure are the witness H, the signature key X and, optionally, the part M_2 of the message. In this case the output S of this step is the signature Σ, see Figure 2.

In a randomized mechanism with a deterministic witness, the inputs to this procedure are the randomizer K, the signature key X, the deterministic witness H and the pre-signature Π. The output of the procedure is the entire signature Σ, which has one part S, or two parts R and S, see Figure 2.

In a randomized mechanism with a randomized witness this procedure consists of two steps. First, the witness R is computed. The witness R depends on the pre-signature Π and, optionally, on M_1. If in the computation of witness a hash-function is used, it shall be specified (see Clause 7). At the second step, the inputs are the randomizer K, the signature key X, the randomized witness R, and, optionally, the part M_2 of the prepared message. The output of the second step is S. The signature Σ has one part S, or two parts R and S, see Figure 3.

10 Verification process

The following data items are required for the verification process:

— domain parameters Z

— verification key Y

— message M

— signature Σ

— identifiers of the hash-functions in use, if not uniquely determined by other means (see Clause 7)

— other text (optional).

The verification process of a digital signature mechanism with appendix consists of the following procedures:

— preparing message for verification

— retrieving witness

— computing verification function

— verifying witness.

Preparing the message for verification may involve extracting the signing entity's identification data from the message *M* in order to identify the right verification key.

If the witness is deterministic, the verifier retrieves the value of the witness as a function of the message. This verification process is depicted in Figure 4. In the other case, depicted in Figure 5, the verifier retrieves the value of the witness from the signature. The first verification process has the advantage that the witness can be retrieved and recomputed in parallel.

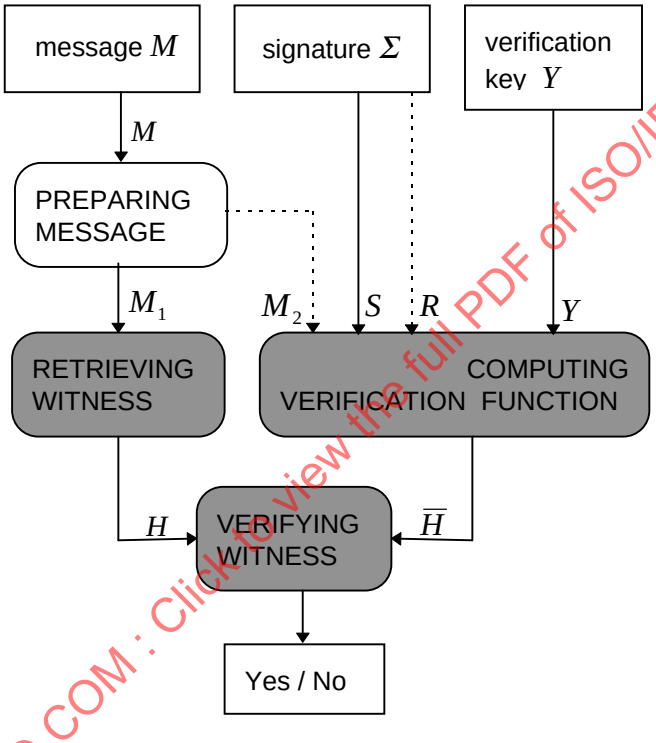


Figure 4 — Verification process with a deterministic witness

10.1 Preparing message

This procedure shall be identical to 9.2. The input is the message *M*, the outputs are the two parts M_1 and M_2 of the message. If a hash-function is used and if it is not uniquely determined by the mechanism or by the domain parameters, the verifier obtains the hash-function identifier from the signed message.

10.2 Retrieving witness

A deterministic witness *H* is retrieved by computing it as a hash-token of M_1 using the same collision-resistant hash-function as the signer in 9.3. Unless uniquely determined by the mechanism or by the domain parameters, the verifier obtains the hash-function identifier from the signed message.

A randomized witness *R* is retrieved either as the first part of the signature Σ, or as an output from the verification function (see Figure 5).

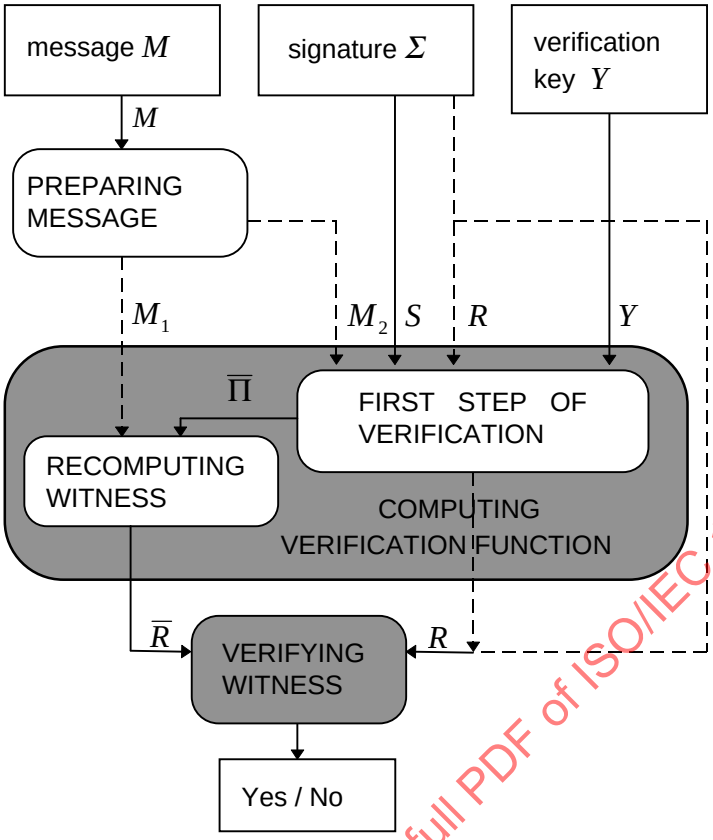


Figure 5 — Verification process with a randomized witness

10.3 Computing verification function

The verification function is determined by the signer's public key *Y*. The signature Σ is a mandatory input to the verification function. If the witness is randomized, the parts M_1 and M_2 of the message which are not empty are mandatory inputs to this step (see Figure 5). Otherwise (see Figure 4), only the second part M_2, if it is not empty, is input to the verification function.

The output from this procedure is a recomputed value of the witness, either $\overline{H}$ or $\overline{R}$. If the randomized witness *R* is not part of the signature, then it is recovered by the verifier as a second output from this procedure, see Figure 5.

10.4 Verifying witness

At this step the two values of the witness are compared, the one retrieved as explained in 10.2, and the one recomputed in 10.3. If these two values are equal, the verifier obtains evidence that the signature Σ was obtained for the message M with the signature key *X* corresponding to the verification key *Y* used in the verification process.

11 Randomized mechanisms with two-part signatures

In this clause refinements of the model described in Clauses 9 and 10 are presented. These refinements are applicable to specification of randomized digital signature mechanisms where the signature is computed in two parts.

11.1 Computing signature

The following data items are required in the computation of the signature in a randomized signature mechanism:

— domain parameters *Z*